Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.1141

(06/2006)

SERIE X: REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

Seguridad de las telecomunicaciones

Lenguaje de marcaje de aserción de seguridad (SAML 2.0)

Recomendación UIT-T X.1141

UIT-T

Unión Internacional de Telecomunicaciones

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.379
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.889
Aplicaciones genéricas de la notación de sintaxis abstracta uno	X.890–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LAS TELECOMUNICACIONES	**X.1000–**

Para más información, véase la Lista de Recomendaciones del UIT-T.

Recomendación UIT-T X.1141

Lenguaje de marcaje de aserción de seguridad (SAML 2.0)

Resumen

El lenguaje SAML representa una estructura basada en XML que es útil para intercambiar información relativa a la seguridad. La información de seguridad se expresa mediante aserciones acerca de sujetos, siendo un sujeto una entidad (persona física u ordenador) que posee una identidad en algún dominio de seguridad. Una sola aserción puede contener varios enunciados internos diferentes de autenticación, autorización y atributos. En la presente Recomendación se define un protocolo que permite que los clientes soliciten aserciones de autoridades del SAML y obtengan las respuestas correspondientes. Este protocolo, que consiste en formatos de mensajes de petición y respuesta basados en XML, puede vincularse con muchos protocolos diferentes de comunicación y transporte subyacentes; hoy en día, el SAML define una vinculación al protocolo simple de acceso a objetos (SOAP) por el protocolo de transferencia de hipertexto (HTTP). Las autoridades del SAML pueden aprovechar varias fuentes de información para crear sus respuestas, como políticas almacenadas externamente y aserciones que se han recibido como parte de las peticiones. En esta Recomendación se definen los elementos de las aserciones, los sujetos, las condiciones, las reglas de procesamiento y los enunciados del SAML. Asimismo, se concibe un perfil de metadatos SAML exhaustivo que incluye el espacio de nombre asociado, los tipos de datos comunes, las reglas de procesamiento y el procesamiento de firmas. Además, se han desarrollado varias vinculaciones con protocolos tales como SOAP, PAOS (SOAP inverso), HTTP redirect, HTTP POST, entre otros. La Recomendación ofrece una lista completa de perfiles del SAML como es el caso de un perfil SSO de explorador web y un perfil de fin de sesión (logout) único que facilita la adopción amplia del SAML 2.0 en la industria. Se incluyen también las directrices necesarias para el contexto y la conformidad de la autenticación.

Esta Recomendación equivale técnicamente a la norma OASIS SAML 2.0 y es compatible con la misma.

Orígenes

La Recomendación UIT-T X.1141 fue aprobada el 13 de junio de 2006 por la Comisión de Estudio 17 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB en la dirección http://www.itu.int/ITU-T/ipr/.

ÍNDICE

Página

1 Alcance 1

2 Referencias 1

3 Definiciones 4
- 3.1 Definiciones de otras Recomendaciones 4
- 3.2 Definiciones adicionales 4

4 Abreviaturas, siglas o acrónimos 8

5 Convenios 9

6 Perspectiva general 10

7 Tipos de datos comunes 11
- 7.1 Valores de cadena 11
- 7.2 Valores de URI 11
- 7.3 Valores de tiempo 11
- 7.4 Identificador (ID) y valores de referencia de ID 11

8 Aserciones y protocolos del SAML 12
- 8.1 Aserciones del SAML 12
- 8.2 Protocolos del SAML 33
- 8.3 Versiones del SAML 60
- 8.4 Sintaxis y procesamiento de firmas XML y SAML 63
- 8.5 Sintaxis y procesamiento de la criptación de XML y SAML 67
- 8.6 Capacidad de extensión del SAML 68
- 8.7 Identificadores definidos en el SAML 70

9 Metadatos del SAML 74
- 9.1 Metadatos 74
- 9.2 Procesamiento de firmas 94
- 9.3 Edición y resolución de los metadatos 95

10 Vinculaciones del SAML 100
- 10.1 Directrices para especificar vinculaciones de protocolo adicionales 100
- 10.2 Vinculaciones de protocolo 101

11 Perfiles para el SAML 128
- 11.1 Conceptos relativos al perfil 128
- 11.2 Especificación de perfiles adicionales 128
- 11.3 Identificadores de método de confirmación 129
- 11.4 Perfiles SSO del SAML 130

12 Contexto de autenticación del SAML 165
- 12.1 Conceptos del contexto de autenticación 165
- 12.2 Declaración del contexto de autenticación 165
- 12.3 Clases del contexto de autenticación 166

13 Requisitos de conformidad para el SAML 210
- 13.1 Perfiles del SAML y posibles implementaciones 211
- 13.2 Conformidad 212
- 13.3 Firma digital XML y criptación XML 215
- 13.4 Utilización de TLS 1.0 215

Anexo A – Esquemas del SAML 216
- A.1 Esquema de la aserción del SAML 216
- A.2 Esquema del contexto de autenticación del SAML 220
- A.3 Esquema del contexto de autenticación del SAML AuthenticatedTelephony (Telefonía autenticada) 221
- A.4 Esquema del contexto de autenticación del SAML específico del protocolo Internet (IP) 222
- A.5 Esquema del contexto de autenticación del SAML relativo a la contraseña del protocolo Internet (IPPWord, *Internet protocol password*) 223
- A.6 Esquema del contexto de autenticación del SAML relativo a Kerberos 224

Página

A.7 Esquema del contexto de autenticación del SAML relativo al servicio móvil con un factor registrado (MobileOneFactor-reg) 225
A.8 Esquema del contexto de autenticación del SAML relativo al servicio móvil con un factor no registrado (MobileOneFactor-unreg) 228
A.9 Esquema del contexto de autenticación del SAML relativo al servicio móvil con dos factores registrados (MobileTwoFactor-reg) 231
A.10 Esquema del contexto de autenticación del SAML relativo al servicio móvil con dos factores no registrados (MobileTwoFactor-unreg) 234
A.11 Esquema del contexto de autenticación del SAML relativo a NomadTelephony (Telefonía nómada) 237
A.12 Esquema del contexto de autenticación del SAML relativo a PersonalizedTelephony (Telefonía personal) 238
A.13 Esquema del contexto de autenticación del SAML relativo a la Privacidad bastante aceptable (PGP) 240
A.14 Esquema del contexto de autenticación del SAML relativo al transporte protegido mediante contraseña (PPT) 241
A.15 Esquema del contexto de autenticación del SAML relativo a la contraseña 242
A.16 Esquema del contexto de autenticación del SAML relativo a PreviousSession (Sesión anterior) 243
A.17 Esquema del contexto de autenticación del SAML relativo a la Smartcard (Tarjeta inteligente) 244
A.18 Esquema del contexto de autenticación del SAML relativo a la SmartcardPKI (Infraestructura de clave pública para la tarjeta inteligente) 245
A.19 Esquema del contexto de autenticación del SAML relativo a SoftwarePKI (Infraestructura de clave pública para el software) 247
A.20 Esquema del contexto de autenticación del SAML relativo a la infraestructura de clave pública única (SPKI) 249
A.21 Esquema del contexto de autenticación del SAML relativo a SRP 250
A.22 Esquema del contexto de autenticación del SAML relativo a telefonía 252
A.23 Esquema del contexto de autenticación del SAML relativo a la sincronización del tiempo (TimeSync) 253
A.24 Esquema del contexto de autenticación del SAML relativo a los tipos 255
A.25 Esquema del contexto de autenticación del SAML relativo a X509 267
A.26 Esquema del contexto de autenticación del SAML relativo a la firma digital XML (XMLDSig) 268
A.27 Esquema del SAML relativo al cliente/mandatario mejorado (ECP) 269
A.28 Esquema del SAML relativo a los metadatos 270
A.29 Esquema del SAML relativo al protocolo 276
A.30 Esquema del SAML relativo a X500 280
A.31 Esquema del SAML relativo a XACML 281

Apéndice I – Consideraciones relativas a la seguridad y la privacidad 282
I.1 Privacidad 282
I.2 Confidencialidad 282
I.3 Seudoanonimidad y anonimidad 283
I.4 Seguridad 283
I.5 Técnicas de seguridad 285
I.6 Consideraciones generales de seguridad en el SAML 287
I.7 Consideraciones de seguridad relativas a las vinculaciones SAML 288

Apéndice II – Registro de aplicación del tipo de medios MIME application/samlassertion+xml 295

Apéndice III – Registro de tipos de medios MIME application/samlmetadata+xml 297

Apéndice IV – Utilización de SSL 299

Apéndice V – Contexto de autenticación de esquema SAML 299

Apéndice VI – Esquema XML de tipos de contexto de autenticación 301

Página

Apéndice VII – Perfil de atributo PAC DCE SAML 314
VII.1 Perfil de atributo PAC DCE 314
VII.2 DCE de esquema SAML 316
VII.3 Ejemplo 317
Apéndice VIII – Aclaraciones del OASIS relativas al SAML 318
VIII.1 Posible error: PE14 318
VIII.2 Posible error: PE26 318
BIBLIOGRAFÍA 321

Recomendación UIT-T X.1141

Lenguaje de marcaje de aserción de seguridad (SAML 2.0)

1 Alcance

En la presente Recomendación se define el lenguaje de marcaje de aserción de seguridad (SAML 2.0). El SAML define la sintaxis y el procesamiento de la semántica de las aserciones expedidas por una entidad del sistema con respecto a un sujeto. Las entidades del sistema SAML, durante el proceso de expedición o examen de las aserciones, pueden aprovechar otros protocolos para la comunicación teniendo en cuenta la propia aserción o el sujeto de la misma. En esta Recomendación se define la estructura de las aserciones del SAML, un conjunto de protocolos asociado, además de las reglas de procesamiento que intervienen en la gestión de un sistema SAML.

Las aserciones del SAML y los mensajes de protocolo se codifican en XML y emplean espacios de nombre XML. Éstos, por lo general, se incorporan en otras estructuras para efectos de transporte, como en las peticiones POST HTTP o los mensajes SOAP codificados en XML. En esta Recomendación se especifican además las vinculaciones del SAML que proporcionan los marcos necesarios para la incorporación y el transporte de los mensajes de protocolo SAML. Asimismo, se ofrece un conjunto de perfiles básico para el empleo de las aserciones y los protocolos del SAML a fin de lograr casos de utilización específicos o el interfuncionamiento cuando se usan las características del SAML.

En la presente Recomendación se define lo siguiente:

1) Requisitos de conformidad del SAML.
2) Aserciones y protocolos del SAML:
   - esquema de aserciones del SAML;
   - esquema de protocolos del SAML;
3) Vinculaciones del SAML.
4) Perfiles del SAML:
   - esquema del perfil ECP SAML;
   - esquema del perfil del atributo SAML X.500/LDAP;
   - esquema del perfil del atributo PAC DCE SAML;
   - esquema del perfil del atributo XACML SAML;
5) Metadatos del SAML.
6) Esquema de metadatos del SAML.
7) Contexto de autenticación del SAML.

2 Referencias

Las siguientes Recomendaciones y demás referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de esta Recomendación. A la fecha de esta publicación, las ediciones citadas están en vigor. Todas las Recomendaciones y demás referencias son objeto de revisión, por lo que se alienta a los usuarios de esta Recomendación a que utilicen la edición más reciente de las Recomendaciones y demás referencias que se indican a continuación. La Oficina de Normalización de las Telecomunicaciones de la UIT publica periódicamente una lista de las Recomendaciones UIT-T vigentes. El IETF publica una lista de las RFC, incluyendo aquellas que han sido reemplazadas por RFC recientes. El W3C, Unicode Consortium and Liberty Alliance, publica una lista de las últimas Recomendaciones y demás publicaciones.

– Recomendación UIT-T X.660 (2004) | ISO/CEI 9834-1:2005, *Tecnología de la información – Interconexión de sistemas abiertos – Procedimientos para la operación de autoridades de registro para interconexión de sistemas abiertos: Procedimientos generales y arcos superiores del árbol de identificadores de objeto de ASN.1*.

- Recomendación UIT-T X.667 (2004) | ISO/CEI 9834-8:2005, *Tecnología de la información – Interconexión de sistemas abiertos – Procedimientos para el funcionamiento de autoridades de registro OSI: Generación y registro de identificadores únicos universales y su utilización como componentes de identificador de objetos ASN.1.*
- Recomendación UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.800 (1991), *Arquitectura de seguridad de la interconexión de sistemas abiertos para aplicaciones del CCITT.*
- Recomendación UIT-T X.811 (1995) | ISO/ CEI 10181-2:1996, *Marcos de seguridad para sistemas abiertos: Marco de autenticación.*
- Recomendación UIT-T X.812 (1995) | ISO/CEI 10181-3:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de autenticación.*
- Recomendación UIT-T X.1142 (2006), *Lenguaje de etiquetas de control de acceso extensible.*
- IETF RFC 1034 (1987), *Domain Names – Concepts and Facilities*.
- IETF RFC 1510 (1993), *The Kerberos Network Authentication Service (V5)*.
- IETF RFC 1750 (1994), *Randomness Recommendations for Security*.
- IETF RFC 1951 (1996), *DEFLATE Compressed Data Format Specification Version 1.3.*
- IETF RFC 1991 (1996), *PGP Message Exchange Formats.*
- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.*
- IETF RFC 2119 (1997), *Keywords for use in RFCs to Indicate Requirement Levels*.
- IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.
- IETF RFC 2253 (1997), *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names.*
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax.*
- IETF RFC 2535 (1999), *Domain Name System Security Extensions.*
- IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1.*
- IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*.
- IETF RFC 2798 (2000), *Definition of the inetOrgPerson LDAP Object Class*.
- IETF RFC 2828 (2000), *Internet Security Glossary.*
- IETF RFC 2914 (2000), *Congestion Control Principles*.
- IETF RFC 2915 (2000), *The Naming Authority Pointer (NAPTR) DNS Resource Record*.
- IETF RFC 2945 (2000), *The SRP Authentication and Key Exchange System*.
- IETF RFC 2965 (2000), *HTTP State Management Mechanism*.
- IETF RFC 3023 (2001), *XML Media Types*.
- IETF RFC 3061 (2001), *A URN Namespace of Object Identifiers.*
- IETF RFC 3075 (2001), *XML-Signature Syntax and Processing*.
- IETF RFC 3377 (2002), *Lightweight Directory Access Protocol (v3): Technical Specification*.
- IETF RFC 3403 (2002), *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database.*
- IETF RFC 3513 (2003), *Internet Protocol Version 6 (IPv6) Addressing Architecture*.
- IETF RFC 3546 (2003), *Transport Layer Security (TLS) Extensions.*
- IETF RFC 3923 (2004), *End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP).*
- IETF RFC 4122 (2005), *A Universally Unique IDentifier (UUID) URN Namespace.*
- Liberty Alliance POAS:2003, R. Aarts, *Liberty Reverse HTTP Binding for SOAP Specification Version 1.0, Liberty Alliance Project.*
- OASIS WSS:2006, WS-Security Core Specification 1.1.

– UNICODE-C, M. Davis; M. J. Dürst: *Unicode Normalization Forms.* UNICODE Consortium, marzo de 2001.

– W3C Canonicalization:2002, *Exclusive XML Canonicalization Version 1.0,* W3C Recommendation, Copyright © [18 de julio de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xml-exc-c14n/.

– W3C Character Model:2005, *Character Model for the World Wide Web 1.0: Fundamentals*, W3C Recommendation, Copyright © [15 de febrero de 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2005/REC-charmod-20050215/.

– W3C Datatypes:2001, *XML Schema Part 2: Datatypes*, W3C Recommendation, Copyright © [2 de mayo de 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

– W3C Encryption:2002, *XML Encryption Syntax and Processing*, W3C Recommendation, Copyright © [10 de diciembre de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/.

– W3C Web Services Glossary:2004, *Web Services Glossary*, W3C Note, Copyright © [11 de febrero de 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/ws-gloss/.

– W3C HTML:1999, *HTML 4.01 Specification*, W3C Recommendation, Copyright © [24 de diciembre de 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/REC-html40/.

– W3C Namespaces:1999, *Namespaces in XML*, W3C Recommendation, Copyright © [14 de enero de 1999] World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/REC-xml-names/.

– W3C Primer:2005, *SOAP Version 1.2 Part 0: Primer*, W3C Recommendation, Copyright © [24 de junio de 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2003/REC-soap12-part0-20030624/.

– W3C Signature:2002, *XML Signature Syntax and Processing*, W3C Recommendation, Copyright © [12 de febrero de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xmldsigcore/.

– W3C Signature Schema:2001, *XML Signature Schema*, W3C Recommendation, Copyright © [1 de marzo de 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd.

– W3C String:1998, *Requirements for String Identity Matching and String Indexing*, W3C Note, Copyright © [10 de julio de 1998] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University, http://www.w3.org/TR/WD-charreq.

– W3C SOAP:2000, *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, Copyright © [8 de mayo de 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

– W3C XHTML:2002, *The Extensible HyperText Markup Language (Second Edition)*, W3C Recommendation, Copyright © [1 de agosto de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xhtml1/.

– W3C XML 1.0:2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, Copyright © [4 de febrero de 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/REC-xml/.

– W3C XML Schema Part 1:2001, *XML Schema Part 1: Structures*, W3C Recommendation, Copyright © [2 de mayo de 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.

NOTA – La referencia a un documento en el marco de esta Recomendación no le confiere al mismo carácter de Recomendación.

3 Definiciones

A los efectos de esta Recomendación, se aplican las siguientes definiciones.

3.1 Definiciones de otras Recomendaciones

3.1.1 En esta Recomendación se utiliza el siguiente término definido en la Rec. UIT-T X.667:

a) UUID.

3.1.2 En esta Recomendación se utilizan los siguientes términos definidos en la Rec. UIT-T X.680:

a) Identificador de objeto.

b) Notación de tipo abierto.

3.1.3 En esta Recomendación se utiliza el siguiente término definido en la Rec. UIT-T X.811:

a) Principio.

3.1.4 En esta Recomendación se utilizan los siguientes términos definidos en la Rec. UIT T X.812:

a) Información de control de acceso.

b) Usuario.

3.1.5 En esta Recomendación se utilizan los siguientes términos definidos en el glosario de servicios web del consorcio W3C:

a) Emisor SOAP inicial.

b) Espacio de nombre.

c) Receptor SOAP final.

d) Esquema XML.

3.1.6 En esta Recomendación se utilizan los siguientes términos definidos en RFC 2828 del IETF:

a) Acceso.

b) Control de acceso.

c) Mandatario.

d) Servidor mandatario.

e) Extraer.

f) Introducir.

g) Arquitectura de seguridad.

h) Política de seguridad.

i) Servicio de seguridad.

3.1.7 En esta Recomendación se utilizan los siguientes términos definidos en RFC 2396 del IETF:

a) Identificador uniforme de recursos (*URI, uniform resource identifier*).

b) Referencia de URI (*URI reference*).

3.2 Definiciones adicionales

3.2.1 **derechos de acceso**: Descripción del tipo de interacciones autorizadas que puede realizar un sujeto con un recurso. Ejemplos: leer, escribir, ejecutar, añadir, modificar y suprimir.

3.2.2 **cuenta**: Acuerdo comercial formal para ofrecer transacciones y servicios regulares entre un principal y un proveedor de servicios comerciales.

3.2.3 vinculación de cuentas: Método para relacionar cuentas de dos proveedores diferentes que representan el mismo principal de manera que los proveedores pueden comunicarse acerca del principal. La vinculación puede establecerse gracias al uso compartido de atributos o a la federación de identidades.

3.2.4 cometido activo: Cometido que asume una entidad del sistema cuando ejecuta alguna operación, por ejemplo, acceder a un recurso.

3.2.5 dominio administrativo: Entorno o contexto definido por alguna combinación de una o varias políticas administrativas, registros de nombre de dominio Internet, entidades jurídicas civiles (por ejemplo, individuos, empresas u otras entidades organizadas formalmente), más un grupo de anfitriones, dispositivos de red y las redes de interconexión (y posiblemente otros dispositivos), más (a menudo varios) servicios de red y las aplicaciones que funcionan en ellos. Este dominio puede incluir o definir uno o varios dominios de seguridad. Un dominio administrativo puede abarcar uno o múltiples emplazamientos. Las características que definen un dominio administrativo pueden evolucionar con el tiempo y en muchos casos así sucede. Los dominios administrativos pueden interactuar y establecer acuerdos para proporcionar y/o consumir servicios a través de fronteras de dominios administrativos.

3.2.6 administrador: Persona que instala o mantiene un sistema, o lo utiliza para gestionar entidades del sistema, usuarios y/o contenido. Por lo general, un administrador está afiliado a un dominio administrativo específico pero puede afiliarse a varios.

3.2.7 afiliación; grupo de afiliación: Conjunto de entidades del sistema que comparten un solo espacio de nombre (en sentido federado) de identificadores para principales.

3.2.8 anonimidad: Calidad o estado anónimo, que equivale a la condición de tener una identidad o nombre desconocido u oculto.

3.2.9 parte asertante: Formalmente, el dominio administrativo que comprende una o varias autoridades del SAML. Informalmente, un ejemplar de una autoridad del SAML.

3.2.10 aserción: Pieza de datos producida por una autoridad del SAML en lo que concierne a un acto de autenticación ejecutado sobre un sujeto , la información del atributo acerca de un sujeto o los datos de autorización que se aplican al sujeto con relación a un recurso específico.

3.2.11 atributo: Característica que distingue a un objeto. En el caso de objetos reales, los atributos se especifican a menudo en términos de características físicas: tamaño, forma, peso y color. Los objetos en el ciberespacio pueden tener atributos que describen: tamaño, tipo de codificación, dirección de red y otros. A menudo los atributos se representan como pares de "attribute name" (nombre de atributo) y "attribute value(s)" (valores de atributo), por ejemplo, "foo" tiene el valor 'bar', "count" tiene el valor 1, "gizmo" tiene los valores "frob" y "2".

3.2.12 aserción de atributo: Aserción que comunica la información de los atributos de un sujeto.

3.2.13 autoridad de atributos: Entidad del sistema que genera aserciones de atributo.

3.2.14 autenticación: Proceso para determinar si alguien o algo es en realidad, con cierto grado de confianza, la persona o cosa que pretende ser.

3.2.15 aserción de autenticación: Aserción que comunica la información relativa a un acto de autenticación satisfactorio que se ejecutó para un sujeto.

3.2.16 autoridad de autenticación: Entidad del sistema que genera aserciones de autenticación.

3.2.17 autorización: Proceso para determinar, mediante la evaluación de la información de control de acceso aplicable, si un sujeto está autorizado a utilizar los tipos de acceso especificados a un recurso específico. Por lo general, la autorización se encuentra en el contexto de la autenticación. Tras la autenticación de un sujeto, éste podrá recibir la autorización para realizar diferentes tipos de acceso.

3.2.18 decisión de autorización: Resultado de un acto de autorización. El resultado puede ser negativo, es decir, puede indicar que el sujeto no tiene autorización alguna de acceso al recurso.

3.2.19 aserción de decisión de autorización: Aserción que comunica la información acerca de una decisión de autorización.

3.2.20 canal de retorno: Se refiere a la comunicación directa entre dos entidades del sistema sin mensajes de "redireccionamiento" a través de otra entidad del sistema tal como un cliente HTTP (por ejemplo, un agente del usuario).

3.2.21 vinculación; vinculación de protocolo: De manera genérica, una especificación de la correspondencia que se establece entre algunos mensajes de protocolo determinados y quizás patrones de intercambio de mensajes, y otro protocolo, de un modo concreto. Por ejemplo, la correspondencia entre el mensaje `<AuthnRequest>` SAML y HTTP es un ejemplo de vinculación. La correspondencia entre el mismo mensaje SAML y SOAP es otra vinculación. En el contexto de SAML, a cada vinculación se le asigna un nombre dentro del patrón "SAML xxx binding".

3.2.22 credenciales: Datos que se transfieren para establecer una identidad principal alegada.

3.2.23 usuario de extremo: Persona física que aplica los recursos.

3.2.24 entidad: Véase entidad del sistema.

3.2.25 federar: Enlazar o vincular dos o más entidades.

3.2.26 federación: Este término se emplea en dos sentidos:

1) Acto de establecer una relación entre dos entidades.
2) Asociación que incluye cualquier número de proveedores de servicio y de proveedores de identidad.

3.2.27 identidad federada: Una identidad del principal está federada entre un conjunto de proveedores cuando existe un acuerdo entre éstos con respecto al uso de un conjunto de identificadores y/o atributos para hacer referencia al principal.

3.2.28 canal frontal: Se trata del "canal de comunicación" que puede ser establecido entre dos servidores que aceptan HTTP empleando mensajes "HTTP redirect" y transfiriendo los mensajes entre ellos a través de un agente del usuario, por ejemplo, un explorador web o cualquier otro cliente HTTP.

3.2.29 identificador: Objeto de datos (por ejemplo, una cadena) que tiene una correspondencia con una entidad del sistema, que se refiere de manera inequívoca a dicha entidad. La entidad del sistema puede tener múltiples y distintos identificadores que se refieren a ella. Un identificador es esencialmente un "atributo distinguido" de una entidad.

3.2.30 identidad: Esencia de una entidad. Una identidad se describe por lo general a través de sus características, entre las cuales puede haber cualquier número de identificadores.

3.2.31 anulación de la federación de una identidad: Sucede cuando los operadores se ponen de acuerdo para dejar de referirse a un principal a través de un conjunto de identificadores y/o atributos.

3.2.32 federación de identidades: Creación de una identidad federada en nombre de un principal.

3.2.33 proveedor de identidad: Clase de proveedor de servicio que crea, conserva y gestiona información relativa a la identidad de los principales y proporciona la autenticación del principal a otros proveedores de servicio que pertenecen a una federación, como es el caso con los perfiles del explorador web.

3.2.34 proveedor de identidad parcial: Clase de proveedor de servicio que crea, conserva y gestiona información relativa a la identidad de los principales y proporciona la autenticación del principal a otros proveedores de servicio que pertenecen a una federación, utilizando sólo las partes necesarias del SAML.

3.2.35 inicio de sesión: Proceso que permite a un usuario presentar sus credenciales a una autoridad de autenticación, establecer una sesión simple y, facultativamente, establecer una sesión completa.

3.2.36 fin de sesión: Proceso que permite a un usuario indicar su deseo de dar por terminada una sola sesión simple o completa (rich).

3.2.37 lenguaje de marcaje: Conjunto de elementos y atributos XML que se deben aplicar a la estructura de un documento XML para una finalidad específica. Este lenguaje se define por lo general mediante un conjunto de diagramas XML y la documentación conexa.

3.2.38 calificador de nombre: Cadena que permite resolver la ambigüedad de un identificador que puede emplearse en varios espacios de nombre (en el sentido federado) para representar a diferentes principales.

3.2.39 parte: Informalmente, uno o varios principales que participan en algún proceso o comunicación, como en el caso de la recepción de una aserción o del acceso a un recurso.

3.2.40 seudónimo persistente: Identificador de nombre que protege la privacidad, el cual es asignado por un proveedor para identificar a un principal ante una parte confiante determinada durante un periodo de tiempo ampliado que abarca múltiples sesiones; puede utilizarse para representar a una federación de identidades.

3.2.41 punto de decisión de política (PDP, *policy decision point*): Entidad del sistema que toma decisiones de autorización para sí misma o para otras entidades del sistema que solicitan esas decisiones. Por ejemplo, un PDP del SAML acepta peticiones de decisión de autorización y como respuesta produce aserciones de decisión de autorización. Un PDP representa a una "autoridad de decisión de autorización".

3.2.42 punto de imposición de la política (PEP, *policy enforcement point*): Entidad del sistema que solicita decisiones de autorización y posteriormente las hace cumplir. Por ejemplo, un PEP del SAML envía peticiones de decisión de autorización a un PDP, y acepta las aserciones de decisión de autorización que le envía este último como respuesta.

3.2.43 identidad del principal: Representación de la identidad del principal, que por lo general se trata de un identificador.

3.2.44 perfil: Conjunto de reglas para una o varias finalidades; a cada conjunto se le asigna un nombre con el patrón "perfil xxx del SAML" o "perfil SAML xxx":

1) Reglas para incorporar aserciones en un protocolo u otro contexto de utilización, así como para extraerlas del mismo.
2) Reglas para emplear mensajes de protocolo SAML en un contexto de utilización específico.
3) Reglas para establecer la correspondencia entre atributos expresados en el SAML y otro sistema de representación de atributos. Este tipo de conjunto de reglas se denomina un "perfil de atributo".

3.2.45 vinculación de protocolo: Véase "vinculación".

3.2.46 proveedor: Forma genérica para referirse a los proveedores de identidad y a los proveedores de servicio.

3.2.47 parte confiante: Entidad del sistema que decide ejecutar una acción basándose en información de otra entidad del sistema. Por ejemplo, una parte confiante del SAML depende de las aserciones que recibe de una parte asertante (una autoridad del SAML) acerca de un sujeto.

3.2.48 peticionario: Entidad del sistema que emplea el protocolo SAML para solicitar servicios de otra entidad del sistema (una autoridad del SAML, un respondedor). No se emplea el término "client" para esta noción debido a que muchas entidades del sistema actúan simultáneamente o en serie como ambos, clientes y servidores. En el caso de que se esté utilizando la vinculación de SOAP para SAML, el peticionario del SAML será distinto desde el punto de vista de la arquitectura, del emisor inicial de SOAP.

3.2.49 recurso: Datos contenidos en un sistema de información (por ejemplo, en forma de ficheros, información en memoria, etc.), así como:

1) Un servicio proporcionado por un sistema.
2) Un ejemplar del equipo del sistema (en otras palabras, un componente del sistema como: hardware, firmware, software o documentación).

3.2.50 respondedor: Entidad del sistema (una autoridad del SAML) que aplica el protocolo del SAML para dar respuesta a una petición de servicios de otra entidad del sistema (un peticionario). No se emplea el término "server" para esta noción debido a que muchas entidades del sistema actúan simultáneamente o en serie como ambos, clientes y servidores. En el caso de que se esté utilizando la vinculación de SOAP para SAML, el respondedor SAML será distinto desde el punto de vista de la arquitectura, del receptor final de SOAP.

3.2.51 cometido, papel: En el diccionario se define un papel como "personaje o parte ejecutada por un intérprete" o "una función o posición". Las entidades del sistema desempeñan varios tipos de cometidos en serie y/o simultáneamente, por ejemplo, cometidos activos y pasivos. La noción de Administrador es a menudo un ejemplo de cometido.

3.2.52 artefacto SAML: Objeto de datos pequeño, de tamaño fijo, estructurado, que apunta, por lo general, a un mensaje de protocolo SAML de tamaño variable y grande. Los artefactos SAML se conciben para incorporarlos en URL y transportarlos en mensajes HTTP, tales como los mensajes de respuesta HTTP con códigos de condición "3xx Redirection", y los mensajes GET HTTP subsiguientes. De esta manera, un proveedor de servicio puede transportar un artefacto SAML indirectamente mediante un agente del usuario, a otro proveedor, quien ulteriormente podrá anular la referencia del artefacto SAML gracias a una interacción directa con el proveedor que lo suministró, para obtener el mensaje de protocolo SAML.

3.2.53 autoridad del SAML: Entidad de sistema abstracto en el modelo de dominio SAML que expide aserciones. Véase asimismo la autoridad de atributos, la autoridad de autenticaciones y el punto de decisión de política (PDP).

3.2.54 seguridad: Grupo de salvaguardas que garantiza la confidencialidad de la información, protege los sistemas o redes utilizados para procesarla y controla el acceso a ellos. Normalmente la seguridad abarca los conceptos de secreto, confidencialidad, integridad y disponibilidad. Su objetivo es garantizar que un sistema resiste los posibles ataques correlacionados.

3.2.55 aserción de seguridad: Aserción examinada en el contexto de la arquitectura de seguridad.

3.2.56 contexto de seguridad: El contexto de seguridad del mensaje, con relación a un mensaje de protocolo SAML individual, es la unión de la semántica de los bloques del encabezamiento de seguridad del mensaje (si la hubiere) y de otros mecanismos de seguridad que pueden ser empleados para la entrega del mensaje al destinatario. Un ejemplo, con respecto a este último, son los mecanismos de seguridad que se aplican en las capas inferiores de la pila de protocolos de red, tales como HTTP, TLS e IPSec.

3.2.57 dominio de seguridad: Entorno o contexto que se define mediante modelos y arquitectura de seguridad, incluyendo un conjunto de recursos y otro de entidades del sistema con autorización para acceder a los recursos. En un solo dominio administrativo pueden residir uno o varios dominios de seguridad. Las características que definen un dominio de seguridad determinado generalmente evolucionan con el paso del tiempo.

3.2.58 expresión de política de seguridad: Correspondencia que se establece entre las identidades del principal y/o sus atributos y las acciones válidas. A menudo, estas expresiones son esencialmente listas de control de acceso.

3.2.59 proveedor de servicio: Cometido que asume una entidad del sistema para proporcionar servicios a los principales u otras entidades del sistema.

3.2.60 proveedor de servicios parcial: Cometido que asume una entidad del sistema para proporcionar servicios a los principales u otras entidades del sistema aprovechando sólo la parte necesaria del protocolo del SAML.

3.2.61 sesión: Interacción duradera entre entidades del sistema, que incluyen a menudo un principal, tipificada por el mantenimiento de algún estado de la interacción durante toda la vida de ésta.

3.2.62 autoridad de sesiones: Cometido adoptado por una entidad del sistema cuando ésta mantiene un estado relacionado con las sesiones.

3.2.63 participante en la sesión: Cometido adoptado por una entidad del sistema cuando ésta participa en una sesión con al menos una autoridad de sesiones.

3.2.64 cancelación de firma: Véase "logout" (desconexión).

3.2.65 firma: Véase "login" (conexión).

3.2.66 emplazamiento: Término informal para un dominio administrativo en sentido geográfico o de nombre DNS. Puede hacer referencia a un tramo geográfico o topológico particular, o puede englobar múltiples dominios administrativos, como puede ser en un emplazamiento ASP.

3.2.67 sujeto: Principal en el contexto de un dominio de seguridad. Las aserciones SAML hacen declaraciones acerca de los sujetos.

3.2.68 entidad del sistema; entidad: Elemento activo de un sistema de ordenador/red. Por ejemplo, un proceso automatizado o un conjunto de procesos, un subsistema, una persona o grupo de personas que incorporan un conjunto de funcionalidades distinto.

3.2.69 fin de temporización: Periodo de tiempo tras el cual alguna condición se torna verdadera si un determinado evento no ha ocurrido. Por ejemplo, cuando una sesión se termina porque su estado ha permanecido inactivo durante un periodo de tiempo específico, se dice que alcanzó el fin de la temporización ("time out").

3.2.70 seudónimo transitorio: Identificador que protege la seguridad, el cual es asignado por un proveedor de identidades a fin de identificar a un principal ante una parte confiante determinada por un periodo de tiempo relativamente corto que no debe abarcar múltiples sesiones.

3.2.71 atributo XML: Estructura de datos XML que se incorpora en la etiqueta de inicio (start-tag) de un elemento XML y que tiene nombre y valor.

3.2.72 elemento XML: Estructura de datos XML que se organiza jerárquicamente entre otras estructuras similares en un documento XML y que se señala mediante una etiqueta de inicio (start-tag) y una etiqueta de fin (end-tag) o una etiqueta vacía (empty tag).

4 Abreviaturas, siglas o acrónimos

A los efectos de esta Recomendación se utilizan las siguientes abreviaturas, siglas o acrónimos.

AA	Autoridad de atributos (*attribute authority*)
ASN.1	Notación de sintaxis abstracta uno (*abstract syntax notation one*)
ASP	Proveedor de servicio de aplicación (*application service provider*)
CA	Autoridad de certificación (*certification authority*)
CMP	Protocolo de gestión de certificados (*certificate management protocol*)

CRL	Lista de revocación de certificados (*certificate revocation list*)
DCE	Entorno informático cómputo distribuido (*distributed computing environment*)
DDDS	Sistema de descubrimiento de delegación dinámico (*dynamic delegation discovery system*)
DNS	Sistema de nombre de dominio (*domain name system*)
ECP	Cliente/mandatario mejorado (*enhanced client/proxy*)
HTTP	Protocolo de transferencia de hipertexto (*hypertext transfer protocol*)
HTTPS	Protocolo de transporte de hipertexto seguro (*secure hypertext transport protocol*)
IdP	Proveedor de identidad (*identity provider*)
IdP Lite	Proveedor de identidad parcial (*identity provider lite*)
IP	Protocolo Internet (*Internet protocol*)
IPSec	Seguridad del protocolo Internet (*Internet protocol security*)
MD5	Algoritmo de compendio de mensajes 5 (*message digest algorithm 5*)
MIME	Ampliaciones multifunción del correo Internet (*multipurpose Internet mail extensions*)
NAPTR	Puntero autoridad de denominación (*naming authority pointer*)
OID	Identificador de objeto (*object identifier*)
PAC	Certificados de atributo de privilegio (*privilege attribute certificates*)
PAOS	SOAP inverso (*reverse SOAP*)
PDP	Punto de decisión de la política (*policy decision point*)
PEP	Punto de imposición de la política (*policy enforcement point*)
PGP	Privacidad bastante aceptable (*pretty good privacy*)
PKI	Infraestructura de clave pública (*public-key infrastructure*)
POP	Prueba de posesión (*proof of possession*)
RA	Autoridad de registro (*registration authority*)
RSA	Rivest Shamir Adleman (Algoritmo de clave pública) (*Rivest Shamir Adleman (public key algorithm)*)
SHA-1	Algoritmo de troceo seguro 1 (*secure hash algorithm 1*)
SP	Proveedor de servicio (*service provider*)
SPKI	Infraestructura de clave pública simple (*simple public key infrastructure*)
SP Lite	Proveedor de servicio parcial (*service provider lite*)
SSO	Inicio de sesión única (*single sign on*)
TLS	Protocolo de seguridad de la capa de transporte (*transport layer security protocol*)
URI	Identificador uniforme de recursos (*uniform resource identifier*)
UTC	Tiempo universal coordinado (*coordinated universal time*)
UUID	Identificador único universal (*universal unique identifier*)
XACML	Lenguaje de marcaje de control de acceso extensible (*eXtensible access control markup language*)
XML	Lenguaje de marcaje extensible (*eXtensible markup language*)

5 Convenios

En esta Recomendación se emplean las palabras clave "debe(n)" y "requerido" ("must", "shall" y "required"), "no debe(n)" ("must not", "shall not"), "debería(n)" ("should"), "no debería(n)" ("should not"), "recomendado" ("recommended"), "puede(n)" ("may") y "facultativo" ("optional"). Para los efectos de esta Recomendación dichos términos deben interpretarse conforme a RFC 2119 del IETF.

En esta Recomendación se emplean documentos de esquemas XML conformes con las partes 1 y 2 del esquema XML del W3C y el texto normativo de esas especificaciones para describir la sintaxis y la semántica de las aserciones SAML codificadas en XML y los mensajes de protocolo. Si hubiere desacuerdo entre los documentos de los esquemas del SAML y las listas de esquemas en esta Recomendación, tendrán precedencia los documentos de esquemas. Obsérvese

que en algunos casos, en esta Recomendación se imponen obligaciones más estrictas que las indicadas en los documentos de los esquemas.

6 Perspectiva general

Esta Recomendación tiene por objetivo especificar la versión 2 del lenguaje de marcaje de aserción de seguridad (SAML v2.0). En esta versión se definen la sintaxis y el procesamiento de la semántica de las aserciones que expide una entidad del sistema acerca de un sujeto. Las entidades del sistema SAML, durante la expedición o el examen de las aserciones, pueden aprovechar otros protocolos de comunicación en lo que concierne a la propia aserción o al sujeto de ésta. En esta Recomendación se definen la estructura de las aserciones del SAML y el conjunto de protocolos asociado, además de las reglas de procesamiento que intervienen en la gestión de un sistema SAML.

Las aserciones del SAML y los mensajes de protocolo se codifican en XML y emplean espacios de nombre XML. Por lo general, se incorporan en otras estructuras de transporte, tales como las peticiones POST HTTP o los mensajes SOAP codificados en XML. En la cláusula 7 se presenta la definición de los tipos de datos comunes que utiliza el SAML. En la cláusula 8 se introduce un marco para las aserciones y los protocolos del SAML. En la cláusula 9 se describe el modelo de metadatos del SAML. En la cláusula 10 se presentan marcos para la incorporación y el transporte de mensajes de protocolo del SAML. En la cláusula 11 se presenta un conjunto de perfiles básico para que las aserciones y los protocolos del SAML se empleen para lograr casos de uso específicos o el interfuncionamiento cuando se aplican las características del SAML. En la cláusula 12 se examina el contexto de autenticación del SAML. En particular, se especifican los siguientes contextos:

- esquema del contexto de autenticación del SAML;
- tipos de esquema del contexto de autenticación del SAML;
- esquema de clase de contexto del SAML para el protocolo Internet;
- esquema de clase de contexto del SAML para la contraseña del protocolo Internet;
- esquema de clase de contexto del SAML para Kerberos;
- esquema de clase de contexto del SAML para el servicio móvil con un factor no registrado;
- esquema de clase de contexto del SAML para el servicio móvil con dos factores no registrados;
- esquema de clase de contexto del SAML para el servicio móvil con un factor con contrato;
- esquema de clase de contexto del SAML para el servicio móvil con dos factores con contrato;
- esquema de clase de contexto del SAML para contraseña;
- esquema de clase de contexto del SAML para transporte protegido por contraseña;
- esquema de clase de contexto del SAML para la sesión anterior;
- esquema de clase de contexto del SAML para clave pública – X.509;
- esquema de clase de contexto del SAML para clave pública – PGP;
- esquema de clase de contexto del SAML para clave pública – SPKI;
- esquema de clase de contexto del SAML para clave pública – firma XML;
- esquema de clase de contexto del SAML para tarjeta inteligente (smartcard);
- esquema de clase de contexto del SAML para PKI de smartcard;
- esquema de clase de contexto del SAML para PKI de software;
- esquema de clase de contexto del SAML para telefonía;
- esquema de clase de contexto del SAML para telefonía (nomádica);
- esquema de clase de contexto del SAML para telefonía (personalizada);
- esquema de clase de contexto del SAML para telefonía (autenticada);
- esquema de clase de contexto del SAML para contraseña distante segura;
- esquema de clase de contexto del SAML para autenticación de cliente basado en certificado SL/TLS;
- esquema de clase de contexto del SAML para testigo de sincronía de tiempo.

En la cláusula 13 se propone un marco para supervisar a los implementadores de SAML a fin de garantizar la conformidad. En la cláusula 13 se examinan los requisitos de conformidad, incluidos los modos operacionales y los modelos de seguridad. En el anexo A se incluye una lista de todos los esquemas del SAML asociados.

7 Tipos de datos comunes

En las cláusulas siguientes se define la forma de utilización e interpretación de los tipos de datos comunes que aparecen en los esquemas del SAML.

7.1 Valores de cadena

Todos los valores de cadena del SAML tienen el tipo **xs:string**, que se construye con tipos de datos (Datatypes) XML W3C. Si no se indica otra cosa en esta Recomendación, todas las cadenas en los mensajes SAML deben constar de al menos un carácter que no sea un espacio en blanco (whitespace).

A menos que se indique otra cosa en esta Recomendación o se empleen perfiles particulares, todos los elementos en los documentos SAML que utilicen el esquema XML tipo **xs:string** o un tipo derivado de éste, deben compararse aplicando un método binario exacto. En particular, las aplicaciones y despliegues del SAML deben ser independientes de las comparaciones de cadenas insensibles a mayúsculas y minúsculas, de la normalización o supresión de los espacios en blanco o de la conversión de formatos específicos de aplicación local como es el caso de los números o las unidades monetarias. El propósito de este requisito es la conformidad con la cadena del W3C.

Si en alguna implementación se comparan valores que se representan mediante codificaciones de caracteres diferentes, se debe aplicar un método que arroje el mismo resultado que se obtiene al convertir ambos valores a la codificación de caracteres Unicode, la forma C de normalización, y a continuación se debe llevar a cabo una comparación binaria exacta. El propósito de este requisito es la conformidad con el modelo de caracteres del W3C y en particular las reglas relativas al texto normalizado en Unicode.

Las aplicaciones que comparan los datos que se reciben en documentos SAML con los datos de fuentes externas deben tener en cuenta las reglas de normalización especificadas para XML. El texto incluido en los elementos se normaliza de tal manera que las terminaciones de las líneas se representan mediante caracteres de cambio de línea ($10_{DECIMAL}$ del CÓDIGO ASCII). Los valores del atributo XML que se definen como cadenas (o sus tipos derivados) se normalizan conforme a XML W3C 1.0, 3.3.3. Todos los caracteres de espacio en blanco serán sustituidos por espacios (blanks) ($32_{DECIMAL}$ del CÓDIGO ASCII).

En esta Recomendación no se define el orden de compaginación o clasificación de los valores de atributo XML o el contenido del elemento. Las implementaciones de SAML deben ser independientes del orden de clasificación específico de los valores, ya que éstos pueden diferir en función de las configuraciones locales de los anfitriones que intervienen.

7.2 Valores de URI

Todos los valores de referencia de URI de SAML tienen el tipo **xs:anyURI**, que se construye con Datatypes XML W3C.

Si no se indica otra cosa en esta Recomendación, todos los valores de referencia de URI utilizados en los elementos o atributos definidos en SAML habrán de constar de al menos un carácter que no sea un espacio en blanco, y es necesario que sean absolutos.

En esta Recomendación se emplean ampliamente las referencias de URI como identificadores, tales como los códigos de situación, los tipos de formato, los nombres de entidad de atributo y sistema, etc. Por consiguiente, es esencial que los valores sean únicos y consistentes, de manera que el mismo URI no pueda ser utilizado nunca en diferentes momentos para representar información subyacente distinta.

7.3 Valores de tiempo

Todos los valores de tiempo del SAML tienen el tipo **xs:dateTime**, que se construye con Datatypes XML W3C y deben expresarse en formato UTC, sin componente de huso horario.

Las entidades del sistema SAML no deberían aplicar una resolución de tiempo con mayor precisión que los milisegundos. Las aplicaciones no deben generar instantes de tiempo que especifiquen segundos intercalar.

7.4 Identificador (ID) y valores de referencia de ID

El tipo **xs:ID** único se emplea para declarar identificadores del SAML para aserciones, peticiones y respuestas. Los valores de tipo **xs:ID** que se declaren en esta Recomendación deben cumplir con las siguientes propiedades, además de las impuestas por la propia definición del tipo **xs:ID**:

- Cualquier parte que asigne un identificador debe garantizar que hay una probabilidad ínfima de que ella misma o cualquier otra parte podrán asignar accidentalmente el mismo identificador a un objeto de datos distinto.

- Cuando un objeto de datos declare que tiene un identificador particular, debe existir exactamente una declaración correspondiente.

El mecanismo mediante el cual una entidad del sistema SAML puede garantizar que el identificador es único será determinado durante la implementación. Si se emplea una técnica aleatoria o seudoaleatoria, la probabilidad de que dos identificadores elegidos aleatoriamente sean idénticos ha de ser menor o igual a 2^{-128} y habría de ser menor o igual a 2^{-160}. Este requisito puede satisfacerse codificando un valor escogido aleatoriamente entre 128 y 160 bits de longitud. La codificación debe ser conforme a las reglas con que se define el tipo de datos (datatype) **xs:ID**. A fin de garantizar las propiedades inequívocas deseadas entre diferentes sistemas, habrá que alimentar un generador seudoaleatorio con material único.

El tipo **xs:NCName** único se emplea en el SAML para los identificadores de referencia de tipo **xs:ID** ya que no es posible usar el tipo **xs:IDREF** para esta finalidad. En el sistema SAML, el elemento al que se hace referencia mediante un identificador del SAML puede ser definido en la práctica en un documento separado de aquel en el que se utiliza la referencia del identificador. La aplicación de **xs:IDREF** podría contravenir el requisito de que su valor debe corresponder con el valor de un atributo de ID en algún elemento del mismo documento XML.

8 Aserciones y protocolos del SAML

El sistema SAML define la sintaxis y el procesamiento de la semántica de las aserciones expedidas por una entidad del sistema con respecto a un sujeto. Las entidades del sistema SAML, durante el proceso de expedición o examen de las aserciones, puede aprovechar otros protocolos para la comunicación teniendo en cuenta la propia aserción o el sujeto de la misma. En esta cláusula se define la estructura de las aserciones del SAML, un conjunto de protocolos asociado, además de las reglas de procesamiento que intervienen en la gestión de un sistema SAML.

Las aserciones del SAML y los mensajes de protocolo se codifican en XML (véase XML W3C 1,0) y emplean espacios de nombre XML (véanse los espacios de nombre de W3C). Éstos, por lo general, se incorporan en otras estructuras para efectos de transporte, como las peticiones POST HTTP o los mensajes SOAP codificados en XML. En la cláusula 10 se presentan los marcos necesarios para la incorporación y el transporte de mensajes de protocolo del SAML. En la cláusula 11 se propone un conjunto de perfiles básico para que las aserciones y los protocolos del SAML se empleen para lograr casos de uso específicos o el interfuncionamiento cuando se aplican las características del SAML.

8.1 Aserciones del SAML

Una aserción constituye un lote de información que suministra cero o varios enunciados expedidos por una **autoridad del SAML**; algunas veces se hace referencia a esas autoridades como **partes asertantes** (***asserting parties***) en los estudios de generación e intercambio de aserciones, y las entidades del sistema que aplican las aserciones recibidas se conocen como **partes confiantes** (***relying parties***). (Estos términos son diferentes de **peticionario** (***requester***) y **respondedor** (***responder***), que se reservan para los estudios del intercambio de mensajes de protocolo del SAML.)

Por lo general, las aserciones del SAML se expiden acerca de un **sujeto** (subject), representado por el elemento `<Subject>`. No obstante, el elemento `<Subject>` es facultativo, y por consecuencia otras especificaciones y perfiles pueden utilizar la estructura de aserciones del SAML para emitir enunciados similares sin especificar un sujeto, o posiblemente especificando el sujeto de una manera alternativa. Normalmente, hay varios **proveedores de servicio** que pueden aprovechar las aserciones acerca de un sujeto para controlar el acceso y ofrecer un servicio personalizado, y por consiguiente, se convierten en las partes confiantes de una parte afirmadora denominada **proveedor de identidad**.

En esta Recomendación se definen tres clases diferentes de enunciados de aserción que pueden ser creadas por una autoridad del SAML. Todos los enunciados definidos por el SAML están asociados con un sujeto. Las tres clases de enunciados son:

- **Authentication** (**Autenticación**): El sujeto de la aserción ha sido autenticado por un medio particular en un momento particular.
- **Attribute** (**Atributo**): El sujeto de la aserción está asociado con los atributos suministrados.
- **Authorization decision** (**Decisión de autorización**): Se ha concedido o denegado una petición de autorización para que el sujeto de la aserción acceda a un recurso específico.

NOTA (informativa) – En PE13 (véase OASIS PE:2006) se recomienda agregar "o aún no se ha determinado" al párrafo anterior.

La estructura exterior de una aserción es genérica, es decir, proporciona información que es común a todos los enunciados contenidos en ella. En una aserción, una serie de elementos internos describe la autenticación, el atributo, la decisión de autorización o los enunciados definidos por el usuario que contienen los datos específicos.

Como se describe en 8.6, el esquema de aserciones del SAML autoriza las extensiones, lo que posibilita complementar las aserciones y los enunciados con extensiones definidas por el usuario, y la definición de nuevas clases de aserciones y enunciados.

8.1.1 Esquema de declaraciones de encabezamiento y espacio de nombre

En el siguiente fragmento de esquema se definen los espacios de nombre XML y otra información de encabezamiento para el esquema de aserción:

```
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
20021210/xenc-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-assertion-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New assertion schema for SAML V2.0 namespace.
        </documentation>
    </annotation>
…
</schema>
```

8.1.2 Identificadores de nombre

En las siguientes cláusulas se definen las construcciones del SAML que contienen los identificadores descriptivos de los sujetos y los expedidores de aserciones y mensajes de protocolo.

En el SAML existe una diversidad de circunstancias en las cuales resulta útil que dos entidades del sistema se comuniquen con respecto a una tercera parte; por ejemplo, el protocolo de petición de autenticación del SAML posibilita que un sujeto sea autenticado por una tercera parte. Por consiguiente, es conveniente establecer un mecanismo para asociar a las partes con identificadores que sean significativos para cada una de ellas. En algunos casos, será necesario limitar el alcance en el que se puede aplicar un identificador a un pequeño conjunto de entidades del sistema (por ejemplo, a fin de conservar la privacidad de un sujeto). Asimismo, pueden utilizarse identificadores similares para hacer referencia al expedidor de una aserción o mensaje de protocolo del SAML.

Existe la posibilidad de que dos o más entidades del sistema puedan usar el mismo valor del identificador de nombre para referirse a diferentes identidades. Así, cada entidad puede comprender de manera diferente ese mismo nombre. El sistema SAML emplea **calificadores de nombre** (***name qualifiers*)** para eliminar la ambigüedad de un identificador de nombre, colocándolo de manera eficaz en un **espacio de nombre** (***namespace***) relacionado con los calificadores de nombre. SAML V2.0 permite calificar a un identificador en virtud de una parte afirmante y una parte confiante o afiliación particular, permitiendo que los identificadores muestren una semántica por pares, cuando proceda.

Los identificadores de nombre también pueden ser criptados para mejorar aun más sus características de conservación de prevacía, especialmente en los casos cuando el identificador puede ser transmitido a través de un intermediario.

NOTA – Para evitar el uso de construcciones del esquema XML relativamente avanzados, los diversos tipos de elementos de identificador no comparten una jerarquía de tipo común.

8.1.2.1 Elemento <BaseID>

El elemento `<BaseID>` representa un punto de extensión que facilita que las aplicaciones añadan nuevas clases de identificadores. Su tipo complejo **BaseIDAbstractType** es abstracto y por lo tanto puede ser aplicado sólo como la base de un tipo derivado. Incluye los siguientes atributos para que sean utilizados por las representaciones del identificador ampliado:

– `NameQualifier` (Calificador de nombre) [Facultativo]

  Dominio de seguridad o administrativo que califica el identificador. Este atributo ofrece un medio para federar identificadores de grupos de usuarios diferentes sin colisiones.

– `SPNameQualifier` (Calificador de nombre de SP) [Facultativo]

  Este atributo califica adicionalmente a un identificador con el nombre de un proveedor de servicio o afiliación de proveedores. Ofrece un medio adicional para federar identificadores basándose en la parte o partes confiantes.

Los atributos `NameQualifier` y `SPNameQualifier` deberían ser omitidos a menos que en la definición del tipo de identificador se describa explícitamente su uso y semántica.

En el siguiente fragmento de esquema se define el elemento `<BaseID>` y su tipo complejo **BaseIDAbstractType**:

```
<attributeGroup name="IDNameQualifiers">
    <attribute name="NameQualifier" type="string" use="optional"/>
    <attribute name="SPNameQualifier" type="string" use="optional"/>
</attributeGroup>
<element name="BaseID" type="saml:BaseIDAbstractType"/>
<complexType name="BaseIDAbstractType" abstract="true">
    <attributeGroup ref="saml:IDNameQualifiers"/>
</complexType>
```

8.1.2.2 Tipo de identificador de nombre (NameIDType) de tipo complejo

Se emplea cuando un elemento es útil para representar a una entidad mediante un nombre con valor de cadena. Se trata de un formato de identificador más restringido que el elemento `<BaseID>` y es el tipo subyacente a los elementos `<NameID>` e `<Issuer>`. Además del contenido de la cadena que incluye el identificador real, proporciona los siguientes atributos facultativos:

– `NameQualifier` (Calificador de nombre) [Facultativo]

  Dominio de seguridad o administrativo que califica el nombre. Este atributo ofrece un medio para federar nombres de grupos de usuarios diferentes sin colisiones.

– `SPNameQualifier` (Calificador de nombre del SP) [Facultativo]

  Este atributo califica adicionalmente a un nombre con el nombre de un proveedor de servicio o afiliación de proveedores. Ofrece un medio adicional para federar nombres basándose en la parte o partes confiantes.

– `Format` (Formato) [Facultativo]

  Referencia de URI que representa la clasificación de la información del identificador basada en cadena. Véase 8.7.3 por lo que concierne a las referencias de URI definidas en el SAML que pueden ser utilizadas como valor del atributo `Format` y sus descripciones y reglas de procesamiento asociadas. Si no se proporciona un valor `Format`, a menos que un elemento basado en este tipo especifique otra cosa, estará vigente el valor `urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified` (véase 8.7.3.1).

  Cuando se emplea un valor de `Format` distinto del especificado en 8.7.3, el contenido de un elemento de este tipo debe interpretarse de acuerdo con la definición de ese formato proporcionada fuera de esta Recomendación. Si no se indican en la definición del formato, las cuestiones relativas a anonimidad, seudoanonimidad y la persistencia del identificador con respecto a las partes asertante y confiante serán específicas de la aplicación.

– `SPProvidedID` (Identificador proporcionado por el SP) [Facultativo]

  Identificador de nombre establecido por un proveedor de servicio o afiliación de proveedores para la entidad, si difiere del identificador de nombre principal proporcionado en el contenido del elemento. Este atributo ofrece un medio para integrar el empleo del SAML con los identificadores existentes que ya están siendo utilizados por un proveedor de servicio. Por ejemplo, un identificador existente puede estar "agregado" a la entidad mediante el protocolo de gestión del identificador de nombre que se define en 8.2.8.

Los elementos que utilizan este tipo y las definiciones de `Format` específicas pueden definir reglas adicionales para el contenido (o la omisión) de estos atributos. Los atributos `NameQualifier` y `SPNameQualifier` deberían ser omitidos a menos que el elemento o formato defina su uso y semántica explícitamente.

El siguiente fragmento de esquema define el tipo complejo **NameIDType**:

```
<complexType name="NameIDType">
    <simpleContent>
         <extension base="string">
              <attributeGroup ref="saml:IDNameQualifiers"/>
              <attribute name="Format" type="anyURI" use="optional"/>
              <attribute name="SPProvidedID" type="string" use="optional"/>
         </extension>
    </simpleContent>
</complexType>
```

8.1.2.3 Elemento <NameID> (Identificador de nombre)

El elemento `<NameID>` es del tipo **NameIDType** (véase 8.1.2.2), y se emplea en varias construcciones de aserción del SAML tales como los elementos `<Subject>` y `<SubjectConfirmation>` y en varios mensajes de protocolo (véase 8.2).

El siguiente fragmento de esquema define el elemento `<NameID>`:

```
<element name="NameID" type="saml:NameIDType"/>
```

8.1.2.4 Elemento <EncryptedID> (Identificador criptado)

El elemento `<EncryptedID>` es del tipo **EncryptedElementType** y transporta el contenido de un elemento identificador no criptado en una modalidad criptada, conforme a las reglas de criptación del W3C. El elemento `<EncryptedID>` incluye los siguientes elementos:

– `<xenc:EncryptedData>` [Obligatorio]

  El contenido criptado y los detalles de criptación asociados, conforme a las reglas de criptación del W3C. El atributo Type debería estar presente, y si lo está, debe contener un valor de `http://www.w3.org/2001/04/xmlenc#Element`. El contenido criptado debe incluir un elemento con un tipo de **NameIDType** o **AssertionType**, o un tipo que se derive de **BaseIDAbstractType**, **NameIDType** o **AssertionType**.

– `<xenc:EncryptedKey>` [Cero o varios]

  Claves de descriptación encapsuladas conforme a las reglas de criptación del W3C. Cada una de las claves encapsuladas debería incluir un atributo de destinatario en el que se especifique la entidad para la que se ha criptado la clave. El valor de dicho atributo debería ser el identificador URI de una entidad del sistema SAML, según se define en 8.4.

Los identificadores criptados están previstos como un mecanismo de protección de privacidad cuando el valor del texto explícito (plain-text) pasa por un intermediario. Por tal razón, el texto criptado debe ser único para cualquier operación de criptación dada; véanse las reglas de criptación de XML del W3C, 6.3.

En este elemento es posible criptar una aserción completa y utilizarla como un identificador. En ese caso, el elemento `<Subject>` de la aserción criptada suministra el "identificador" del sujeto de la aserción circundante. Por consiguiente, si la identificación de la aserción no es válida, la aserción circundante tampoco lo será.

En el siguiente fragmento de esquema se define el elemento `<EncryptedID>` y su tipo complejo **EncryptedElementType**:

```
<complexType name="EncryptedElementType">
    <sequence>
         <element ref="xenc:EncryptedData"/>
         <element ref="xenc:EncryptedKey" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
<element name="EncryptedID" type="saml:EncryptedElementType"/>
```

8.1.2.5 Elemento <Issuer> (Expedidor)

El elemento `<Issuer>` y el tipo complejo **NameIDType** proporcionan información acerca del expedidor de una aserción o de un mensaje de protocolo del SAML. El elemento necesita emplear una cadena para transportar el nombre del expedidor, pero se pueden incluir varios trozos de datos descriptivos (véase 8.1.2.2).

La invalidación de la regla habitual para este tipo de elemento, en caso de que no se proporcione el valor `Format` con este elemento, ocasiona que el valor `urn:oasis:names:tc:SAML:2.0:nameid-format:entity` esté vigente (véase 8.1.2.2).

En el siguiente fragmento de esquema se define el elemento `<Issuer>`:

```
<element name="Issuer" type="saml:NameIDType"/>
```

8.1.3 Aserciones

En las cláusulas siguientes se definen las construcciones del SAML que contienen información sobre la aserción u ofrecen un mecanismo para hacer referencia a una aserción existente.

8.1.3.1 Elemento <AssertionIDRef> (Referencia del identificador de la aserción)

El elemento `<AssertionIDRef>` hace referencia a una aserción del SAML gracias a su identificador único. La autoridad específica que expide la aserción o de la que puede obtenerse ésta, no se especifica como parte de la referencia. Véase 8.2.3 por lo que se refiere a un elemento de protocolo que utiliza esa referencia para preguntar por la aserción correspondiente.

En el siguiente fragmento de esquema se define el elemento `<AssertionIDRef>`:

```
<element name="AssertionIDRef" type="NCName"/>
```

8.1.3.2 Elemento <AssertionURIRef> (Referencia de URI de aserción)

El elemento `<AssertionURIRef>` hace referencia a una aserción del SAML gracias a la referencia URI. Esta última puede ser empleada para recuperar la aserción correspondiente de una forma específica conforme a la referencia URI. Véase 7.3 por lo que se refiere al modo de uso de este elemento en una vinculación de protocolo para lograr el objetivo en cuestión.

En el siguiente fragmento de esquema se define el elemento `<AssertionURIRef>`:

```
<element name="AssertionURIRef" type="anyURI"/>
```

8.1.3.3 Elemento <Assertion> (Aserción)

El elemento `<Assertion>` es del tipo complejo **AssertionType**, que permite especificar la información básica común para todas las aserciones, incluyendo los siguientes elementos y atributos:

– `Version` (Versión) [Obligatorio]

  Versión de esta aserción. El identificador para la versión de SAML que se define en esta Recomendación es "2.0". Las versiones del SAML se examinan en 8.3.

– `ID` (Identificador) [Obligatorio]

  Identificador de esta aserción. Es del tipo **xs:ID** y debe tener conformidad con los requisitos especificados en 7.3 para la singularidad del identificador.

– `IssueInstant` (Instante de expedición) [Obligatorio]

  Instante de expedición expresado en UTC, conforme a 7.3.

– `<Issuer>` (Expedidor) [Obligatorio]

  Autoridad del SAML que reivindica la expedición de la aserción. No debería existir ambigüedad alguna del expedidor ante las partes confiantes objetivo.

  En esta Recomendación no se define ninguna relación particular entre la entidad representada por este elemento y el firmante de la aserción (si lo hubiere). Este tipo de requisitos impuestos por una parte confiante que acepta la aserción o bien mediante perfiles específicos, serán específicos de la aplicación.

– `<ds:Signature>` [Facultativo]

  Firma XML que protege la integridad del expedidor de la aserción y lo autentica, conforme a 8.4.

– `<Subject>` (Sujeto) [Facultativo]

Sujeto de los enunciados en la aserción.

– `<Conditions>` (Condiciones) [Facultativo]

  Condiciones que deben ser evaluadas cuando se confirma la validez de la aserción y/o cuando se emplea la misma. Véase 8.1.5 en cuanto a la información adicional sobre el modo de evaluación de las condiciones.

– `<Advice>` (Asesoramiento) [Facultativo]

  Información adicional relativa a la aserción que apoya el proceso en determinadas situaciones, pero que podrá ser ignorada por algunas aplicaciones que no comprenden el asesoramiento o no desean utilizarlo.

  Cero o más de los siguientes elementos de enunciado:

– `<Statement>`

  Enunciado de un tipo que se define en un esquema de extensión. Es necesario utilizar un atributo **xsi:type** para indicar el tipo de enunciado real.

– `<AuthnStatement>`

  Enunciado de autenticación.

– `<AuthzDecisionStatement>`

  Enunciado de decisión de autorización.

– `<AttributeStatement>`

  Enunciado de atributo.

Una aserción sin enunciados debe contener un elemento `<Subject>`. Este tipo de aserción permite identificar a un principal de tal manera que pueda ser referenciado o confirmado aplicando métodos del SAML, pero no afirma ninguna información adicional asociada con ese principal.

De lo contrario, si el elemento `<Subject>` está incluido, permite identificar el sujeto de todos los enunciados en la aserción. Si se omite, los enunciados en la aserción se aplican a un sujeto o sujetos que se identifican de una manera específica de aplicación o perfil. El propio SAML no define esos enunciados, y en esta Recomendación una aserción sin sujeto no tiene un significado definido.

A menudo, es posible que, en función de los requisitos de los protocolos o perfiles particulares, sea necesario autenticar el expedidor de una aserción SAML y que se requiera protección de la integridad. La autenticación e integridad de los mensajes puede obtenerse mediante mecanismos ofrecidos por una vinculación de protocolo que esté siendo utilizada durante la entrega de una aserción (véase la cláusula 10). La aserción SAML puede estar firmada, lo que proporciona tanto la autenticación del expedidor como la protección de la integridad.

Si se emplea una firma, el elemento `<ds:Signature>` tiene que estar incluido, y la parte confiante debe verificar la validez de la firma (es decir, que la aserción no ha sido alterada) de conformidad con las reglas de firma XML del W3C. Si no es válida, la parte confiante no debe utilizar su contenido. Si es válida, la parte confiante debe evaluar la firma a fin de determinar la identidad y la aplicabilidad del expedidor conforme se juzgue pertinente (por ejemplo, condiciones de evaluación, asesoramiento, seguimiento de reglas específicas de perfil, etc.).

La inclusión de múltiples enunciados, firmados o no, dentro de una sola aserción es equivalente, desde el punto de vista semántico, a un conjunto de aserciones que contienen esos enunciados individualmente (siempre que el sujeto, condiciones, etc., sean idénticos).

En el siguiente fragmento de esquema se define el elemento `<Assertion>` y su tipo complejo **AssertionType**:

```
<element name="Assertion" type="saml:AssertionType"/>
<complexType name="AssertionType">
       <sequence>
              <element ref="saml:Issuer"/>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="saml:Subject" minOccurs="0"/>
              <element ref="saml:Conditions" minOccurs="0"/>
              <element ref="saml:Advice" minOccurs="0"/>
              <choice minOccurs="0" maxOccurs="unbounded">
                     <element ref="saml:Statement"/>
                     <element ref="saml:AuthnStatement"/>
                     <element ref="saml:AuthzDecisionStatement"/>
                     <element ref="saml:AttributeStatement"/>
              </choice>
       </sequence>
```

```
      <attribute name="Version" type="string" use="required"/>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>
```

8.1.3.4 Elemento <EncryptedAssertion> (Aserción criptada)

El elemento <EncryptedAssertion> representa una aserción criptada según las reglas de criptación del W3C, y contiene los elementos:

– <xenc:EncryptedData> [Obligatorio]

  Contenido criptado y detalles de la criptación asociada según las reglas de criptación del W3C. El atributo Type debería estar presente y, de ser así, habrá de contener un valor de http://www.w3.org/2001/04/xmlenc#Element. El contenido criptado ha de incluir un elemento con un tipo de **AssertionType** o un derivado del mismo.

– <xenc:EncryptedKey> [Cero o varios]

  Claves de descriptación encapsuladas conforme a las reglas de criptación del W3C. Cada una de las claves encapsuladas debería incluir un atributo de destinatario en el que se especifique la entidad para la que se ha criptado la clave. El valor de dicho atributo debería ser el identificador URI de una entidad del sistema SAML, según se define en 8.7.

Las aserciones criptadas están previstas como un mecanismo de protección de confidencialidad cuando el valor del texto explícito pasa por un intermediario.

En el siguiente fragmento de esquema se define el elemento <EncryptedAssertion>:

```
<element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
```

8.1.4 Sujetos

En esta cláusula se definen las construcciones del SAML que se emplean para describir el sujeto de una aserción. El elemento facultativo <Subject> permite especificar el principal que es el sujeto de todos (cero o varios) los enunciados en la aserción. Contiene un identificador, una serie de una o varias confirmaciones de sujeto, o ambos:

– <BaseID>, <NameID> o <EncryptedID> [Facultativo]

  Identifica el sujeto.

– <SubjectConfirmation> [Cero o varios]

  Información que posibilita confirmar el sujeto. Si se proporciona más de una confirmación de sujeto, en ese caso al satisfacer cualquiera de ellas será suficiente para confirmar el sujeto para fines de aplicación de la aserción.

Un elemento <Subject> puede contener un identificador y cero o varias confirmaciones de sujeto, las cuales pueden ser verificadas por una parte confiante durante el proceso de una aserción. Si se logra verificar una de las confirmaciones de sujeto incluidas, la parte confiante podrá tratar a la entidad que presenta la aserción como una entidad que ha sido asociada por la parte asertante con el principal identificado en el identificador de nombre y con los enunciados en la aserción. Esta entidad atestiguadora y el sujeto real pueden ser o no la misma entidad.

Si no se incluyen confirmaciones de sujeto, cualquier relación entre el presentador de la aserción y el sujeto real no estará especificada.

Un elemento <Subject> no debería identificar a más de un principal.

En el siguiente fragmento de esquema se define el elemento <Subject> y su tipo complejo **SubjectType**:

```
<element name="Subject" type="saml:SubjectType"/>
<complexType name="SubjectType">
      <choice>
      <sequence>
            <choice>
                  <element ref="saml:BaseID"/>
                  <element ref="saml:NameID"/>
                  <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmation" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
```

```
      <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
     </choice>
</complexType>
```

8.1.4.1 Elemento <SubjectConfirmation> (Confirmación de sujeto)

El elemento <SubjectConfirmation> ofrece un mecanismo para que una parte confiante pueda verificar la correspondencia entre el sujeto de la aserción y la parte con la que se está comunicando la parte confiante. Contiene los siguientes atributos y elementos:

– Method (Método) [Obligatorio]

Referencia de URI que permite identificar un protocolo o mecanismo que ha de utilizarse para confirmar el sujeto. En la cláusula 11 se definen las referencias de URI que identifican los métodos de confirmación definidos en el SAML. Es posible añadir métodos adicionales mediante la definición de nuevos URI y perfiles o gracias a un acuerdo privado.

– <BaseID>, <NameID> o <EncryptedID> [Facultativo]

Permite identificar a la entidad que podrá satisfacer los requisitos de confirmación del sujeto circundante.

– <SubjectConfirmationData> (Datos de confirmación del sujeto) [Facultativo]

Información de confirmación adicional que será utilizada por un método de confirmación específico. Por ejemplo, el contenido convencional de este elemento podría ser un elemento <ds:KeyInfo> según las reglas de criptación del W3C, que identifica una clave criptográfica (véase también 8.1.4.3). Los métodos de confirmación particulares pueden definir un tipo de esquema conveniente para describir los elementos, atributos o el contenido que pueden aparecer en el elemento <SubjectConfirmationData>.

En el siguiente fragmento de esquema se define el elemento <SubjectConfirmation> y su tipo complejo **SubjectConfirmationType**:

```
<element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
<complexType name="SubjectConfirmationType">
      <sequence>
            <choice minOccurs="0">
                  <element ref="saml:BaseID"/>
                  <element ref="saml:NameID"/>
                  <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
      </sequence>
      <attribute name="Method" type="anyURI" use="required"/>
</complexType>
```

8.1.4.2 Elemento <SubjectConfirmationData> (Datos de confirmación de sujeto)

El elemento <SubjectConfirmationData> tiene el tipo complejo **SubjectConfirmationDataType**. Este elemento permite especificar datos adicionales para confirmar el sujeto o restringir las circunstancias en las que se realizará dicha confirmación. La confirmación del sujeto se llevará a cabo cuando una parte confiante trate de verificar la relación entre la entidad que presenta la aserción (entidad atestiguadora) y el sujeto reivindicado por la aserción. Contiene los siguientes atributos facultativos que pueden ser aplicados a cualquier método:

– NotBefore (No antes de) [Facultativo]

Instante de tiempo antes del cual el sujeto no puede ser confirmado. El valor de tiempo se codifica en UTC conforme a 7.3.

– NotOnOrAfter (Ni en ese instante ni después) [Facultativo]

Instante de tiempo en el que el sujeto ya no puede ser confirmado. El valor de tiempo se codifica en UTC conforme a 7.3.

– Recipient (Destinatario) [Facultativo]

URI que permite especificar la entidad o la ubicación en la que una entidad atestiguadora puede presentar la aserción. Este atributo podría indicar, por ejemplo, que la aserción debe ser entregada a un punto extremo de una red particular a fin de impedir que un intermediario pueda redirigirla a algún otro sitio.

– `InResponseTo` (En respuesta a) [Facultativo]

  `ID` de un mensaje de protocolo SAML gracias al cual una entidad atestiguadora puede responder para presentar la aserción. Este atributo podría ser utilizado, por ejemplo, para establecer una correlación entre la aserción y una petición del SAML que haya resultado de su presentación.

– `Address` (Dirección) [Facultativo]

  Dirección o ubicación de la red desde la cual una entidad atestiguadora puede presentar la aserción. Este atributo podría ser utilizado, por ejemplo, para establecer una vinculación entre la aserción y direcciones de clientes particulares a fin de evitar que un atacante pueda sustraer la aserción fácilmente y presentarla desde otra ubicación. Las direcciones IPv4 deben representarse en el formato habitual con decimales separados por puntos (por ejemplo, "1.2.3.4"). Las direcciones IPv6 deben representarse conforme a la norma RFC 3513, 2.2 del IETF (por ejemplo, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210").

– `Atributos arbitrarios`

  Este tipo de atributo complejo emplea un punto de extensión `<xs:anyAttribute>` para agregar fácilmente atributos XML arbitrarios calificados por espacio de nombre a las construcciones `<SubjectConfirmationData>` sin necesidad de una extensión de esquema explícita. Esto permite añadir campos adicionales conforme se requiera para proporcionar información adicional relativa a la confirmación. Las extensiones del SAML no deben añadir atributos XML locales (que no son calificados por espacio de nombre) o atributos XML calificados por un espacio de nombre definido por el SAML al tipo complejo **SubjectConfirmationDataType** o a uno de sus derivados; esos atributos se reservan para mejorar y mantener el propio SAML en el futuro.

– `Elementos arbitrarios`

  Este tipo de atributo complejo emplea un punto de extensión `<xs:any>` para agregar fácilmente elementos XML arbitrarios a las construcciones `<SubjectConfirmationData>` sin necesidad de una extensión de esquema explícita. Esto permite añadir elementos adicionales conforme se requiera para proporcionar información adicional relativa a la confirmación.

Es posible que los métodos y perfiles de confirmación particulares que utilizan dichos métodos exijan la utilización de uno o varios de los atributos que se definen en este tipo complejo. En la cláusula 13 se presentan ejemplos sobre las opciones de utilización de estos atributos (y la confirmación del sujeto en general).

El periodo de tiempo especificado por los atributos facultativos `NotBefore` y `NotOnOrAfter`, si están incluidos, debería mantenerse dentro del periodo de validez total de la aserción especificado por los atributos `NotBefore` y `NotOnOrAfter` del elemento `<Conditions>`. Si los dos atributos están presentes, el valor de `NotBefore` ha de ser menor que (o anterior a) el valor de `NotOnOrAfter`.

En el siguiente fragmento de esquema se define el elemento `<SubjectConfirmationData>` y su tipo complejo **SubjectConfirmationDataType**:

```
<element name="SubjectConfirmationData"
type="saml:SubjectConfirmationDataType"/>
<complexType name="SubjectConfirmationDataType" mixed="true">
      <complexContent>
            <restriction base="anyType">
                  <sequence>
                        <any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
                  </sequence>
                  <attribute name="NotBefore" type="dateTime"
use="optional"/>
                  <attribute name="NotOnOrAfter" type="dateTime"
use="optional"/>
                  <attribute name="Recipient" type="anyURI"
use="optional"/>
                  <attribute name="InResponseTo" type="NCName"
use="optional"/>
                  <attribute name="Address" type="string"
use="optional"/>
                  <anyAttribute namespace="##other"
processContents="lax"/>
            </restriction>
      </complexContent>
</complexType>
```

8.1.4.3 Tipo complejo KeyInfoConfirmationDataType (Tipo de datos de confirmación de información de clave)

El tipo complejo **KeyInfoConfirmationDataType** exige que un elemento `<SubjectConfirmationData>` incluya uno o varios elementos `<ds:KeyInfo>` que identifican las claves criptográficas empleadas de alguna manera para autenticar una entidad atestiguadora. El método de confirmación específico debe definir el mecanismo exacto para utilizar los datos de confirmación. Es posible que también aparezcan los atributos facultativos definidos por el tipo complejo **SubjectConfirmationDataType**.

Cualquier método de confirmación que defina sus datos de confirmación en virtud del elemento `<ds:KeyInfo>` debería aplicar este tipo complejo o uno de sus derivados.

De conformidad con las reglas de criptación del W3C, cada elemento `<ds:KeyInfo>` debe identificar una clave criptográfica única. Se pueden identificar múltiples claves mediante elementos `<ds:KeyInfo>` independientes, como es el caso cuando un principal utiliza diferentes claves para confirmarse el mismo ante distintas partes confiantes.

En el siguiente fragmento de esquema se define el tipo complejo **KeyInfoConfirmationDataType**:

```
<complexType name="KeyInfoConfirmationDataType" mixed="false">
      <complexContent>
            <restriction base="saml:SubjectConfirmationDataType">
                  <sequence>
                        <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
                  </sequence>
            </restriction>
      </complexContent>
</complexType>
```

8.1.4.4 Ejemplo de un <Subject> confirmado por clave

Para ilustrar la forma en la que encajan los diversos tipos y elementos, a continuación se presenta un ejemplo de un elemento `<Subject>` que contiene un identificador de nombre y una confirmación de sujeto basados en la prueba de posesión de una clave. En este caso, el empleo del **KeyInfoConfirmationDataType** para identificar la sintaxis de datos de confirmación es un elemento `<ds:KeyInfo>`:

```
<Subject>
      <NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">
      scott@example.org
      </NameID>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-
of-key">
            <SubjectConfirmationData
xsi:type="saml:KeyInfoConfirmationDataType">
                  <ds:KeyInfo>
                        <ds:KeyName>Scott's Key</ds:KeyName>
                  </ds:KeyInfo>
            </SubjectConfirmationData>
      </SubjectConfirmation>
</Subject>
```

8.1.5 Condiciones

En esta cláusula se definen las construcciones del SAML que imponen restricciones al uso aceptable de las aserciones SAML. El elemento `<Conditions>` puede incluir los siguientes elementos y atributos:

– `NotBefore` (No antes de) [Facultativo]

  Especifica el primer instante en el que la aserción es válida. El valor de tiempo se codifica en UTC conforme a 7.3.

– `NotOnOrAfter` (Ni en ese instante ni después) [Facultativo]

  Especifica el instante en el que la aserción deja de ser válida. El valor de tiempo se codifica en UTC conforme a 7.3.

– `<Condition>` (Condición) [Cualquier número]

  Condición de un tipo que se define en un esquema de extensión. Podrá emplearse un atributo `xsi:type` para indicar el tipo de condición real.

– `<AudienceRestriction>` (Restricción de la audiencia) [Cualquier número]

  Especifica que la aserción está dirigida a una audiencia particular.

– `<OneTimeUse>` (Se usa una sola vez) [Facultativo]

  Especifica que la aserción debería utilizarse inmediatamente y no conservarse para uso posterior. Aunque el esquema permite múltiples ocurrencias, habrá como máximo un ejemplar de este elemento.

– `<ProxyRestriction>` (Restricción de mandatario) [Facultativo]

  Especifica las limitaciones que impone la parte asertante a las partes confiantes que desean actuar posteriormente como partes asertantes y expedir sus propias aserciones basándose en la información contenida en la aserción original. Aunque el esquema permite múltiples ocurrencias, habrá como máximo un ejemplar de este elemento.

Debido a que la utilización del atributo `xsi:type` podría permitir que una aserción contenga más de un ejemplar de un subtipo de **ConditionsType** (tal como **OneTimeUseType**) definido en el SAML, el esquema no limita explícitamente el número de veces que se pueden incluir condiciones particulares. Un tipo de condición particular puede definir límites para ese tipo de utilización, como se mostró antes.

En el siguiente fragmento de esquema se define el elemento `<Conditions>` y su tipo complejo **ConditionsType**:

```
<element name="Conditions" type="saml:ConditionsType"/>
<complexType name="ConditionsType">
      <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Condition"/>
            <element ref="saml:AudienceRestriction"/>
            <element ref="saml:OneTimeUse"/>
            <element ref="saml:ProxyRestriction"/>
      </choice>
      <attribute name="NotBefore" type="dateTime" use="optional"/>
      <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
</complexType>
```

8.1.5.1 Reglas de procesamiento generales

Si una aserción contiene un elemento `<Conditions>`, su validez dependerá de los subelementos y atributos proporcionados, al aplicar las siguientes reglas en el orden indicado más adelante.

Una aserción que tiene el estado de validez de condición Valid puede, no obstante, no ser digna de confianza o no ser válida por motivos tales como no estar bien formada o no contar con un esquema válido, no haber sido expedida por una autoridad del SAML de confianza o no haber sido autenticada por un mecanismo de confianza.

Es posible que algunas de esas condiciones no repercutan directamente en la validez de la aserción contenedora (siempre se evalúan como Valid), pero pueden restringir el comportamiento de las partes confiantes con respecto a la utilización de la aserción.

- Si no se incluyen subelementos o atributos en el elemento `<Conditions>`, la aserción se considera Valid con respecto al procesamiento de la condición.
- Si se determina que algún subelemento o atributo del elemento `<Conditions>` no es válido, la aserción se considera Invalid (no válida).
- Si no es posible evaluar algún subelemento o atributo del elemento `<Conditions>` o si se encuentra un elemento que no se puede comprender, no se podrá determinar la validez de la aserción y será considerada como Indeterminate.
- Si se determina que todos los subelementos y atributos del elemento `<Conditions>` son Valid, la aserción se considera Valid con respecto al procesamiento de la condición.

El procesamiento de la condición termina al aplicar la primera regla; por lo tanto, la determinación de que una aserción es Invalid tendrá precedencia sobre la correspondiente a Indeterminate.

Cuando se determina que una aserción es Invalid o Indeterminate, ésta debe ser rechazada por una parte confiante (en cualquier contexto o perfil que esté siendo procesado), exactamente como si la aserción estuviese mal construida o se considerase inútil.

8.1.5.2 Atributos NotBefore (No antes de) y NotOnOrAfter (Ni en ese instante ni después)

Los atributos `NotBefore` y `NotOnOrAfter` especifican los límites temporales relativos a la validez de la aserción dentro del contexto de su perfil o perfiles de utilización. No garantizan que los enunciados en la aserción serán correctos o precisos durante el periodo de validez.

El atributo `NotBefore` especifica el momento de inicio del intervalo de validez. El atributo `NotOnOrAfter` especifica el momento en el que termina el intervalo de validez.

Si se omite el valor de cualquiera de estos dos atributos, se considerará no especificado. Si no se especifica el atributo `NotBefore` (y si las demás condiciones proporcionadas equivalen a Valid), en ese caso, la aserción será Valid con respecto a las condiciones en cualquier momento antes del instante especificado por el atributo `NotOnOrAfter`. Si no se especifica el atributo `NotOnOrAfter` (y si las demás condiciones proporcionadas equivalen a Valid), la aserción será Valid con respecto a las condiciones a partir del instante especificado por el atributo `NotBefore` sin expiración. Si no se especifica ninguno de los dos (y si cualquiera de las demás condiciones equivale a Valid), la aserción será Valid en cualquier momento con respecto a las condiciones.

Si ambos atributos están presentes, el valor de `NotBefore` debe ser menor que (o anterior a) el valor de `NotOnOrAfter`.

8.1.5.3 Elemento <Condition> (Condición)

El elemento `<Condition>` es útil como punto de extensión de las nuevas condiciones. Su tipo complejo **ConditionAbstractType** es abstracto y por lo tanto sólo se puede utilizar como base de un tipo derivado.

En el siguiente fragmento de esquema se define el elemento `<Condition>` y su tipo complejo **ConditionAbstractType**:

```
<element name="Condition" type="saml:ConditionAbstractType"/>
<complexType name="ConditionAbstractType" abstract="true"/>
```

8.1.5.4 Elementos <AudienceRestriction> (Restricción de audiencia) y <Audience> (audiencia)

El elemento `<AudienceRestriction>` especifica que la aserción está dirigida a una o más audiencias específicas que se identifican mediante elementos `<Audience>`. Aunque una parte confiante del SAML externa a las audiencias especificadas dispone de capacidad para sacar conclusiones de una aserción, la parte asertante del SAML no lleva a cabo ninguna representación explícita en cuanto a la precisión o confianza de esa parte. Contiene el siguiente elemento:

– `<Audience>` (Audiencia)

  Referencia de URI que permite identificar una audiencia objetivo. Esta referencia de URI puede identificar un documento que describe los términos y condiciones de los miembros de una audiencia. Asimismo, puede contener el URI del identificador único a partir de un identificador de nombre del SAML que describe una entidad del sistema.

La condición de restricción de audiencia será evaluada como Valid únicamente si la parte confiante del SAML es miembro de una o más de las audiencias especificadas.

La parte atestiguadora del SAML no puede impedir que una parte que se entere de la aserción realice una acción basándose en la información disponible. No obstante, el elemento `<AudienceRestriction>` posibilita que la parte atestiguadora del SAML establezca explícitamente que no se otorgará ninguna garantía a esa parte en un formato legible por las máquinas y las personas. Aunque no existe la seguridad de que un tribunal mantendrá firme ese tipo de exclusión de garantía en cualquier circunstancia, la probabilidad de mantenerla firme ha mejorado considerablemente.

En una misma aserción pueden incluirse múltiples elementos `<AudienceRestriction>` y cada uno de ellos debe ser evaluado de forma independiente. El efecto de este requisito y la definición anterior es que dentro de una condición dada, las audiencias forman una disyunción ("OR") mientras que múltiples condiciones forman una conjunción ("AND").

En el siguiente fragmento de esquema se define el elemento <AudienceRestriction> y su tipo complejo **AudienceRestrictionType**:

```
<element name="AudienceRestriction"
      type="saml:AudienceRestrictionType"/>
<complexType name="AudienceRestrictionType">
      <complexContent>
            <extension base="saml:ConditionAbstractType">
                  <sequence>
                        <element ref="saml:Audience"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="Audience" type="anyURI"/>
```

8.1.5.5 Elemento <OneTimeUse> (Se usa una sola vez)

En general, las partes confiantes pueden decidir que van a conservar las aserciones o la información contenida en ellas de alguna otra forma, para volver a utilizarlas. El elemento de condición <OneTimeUse> permite que una autoridad indique que es probable que la información en la aserción cambie a corto plazo y que se debería obtener información reciente para cada uso. Un ejemplo podría ser una aserción que contiene un <AuthzDecisionStatement> que es el resultado de una política mediante la cual se especificó control de acceso en función de la hora del día.

Si los relojes del sistema en un entorno distribuido pudieran sincronizarse con precisión, se podría satisfacer este requisito mediante el empleo cuidadoso del intervalo de validez. No obstante, ya que siempre existirá un desalineamiento entre los sistemas que se combinará con los posibles retardos de transmisión, el expedidor no dispone de una forma conveniente de limitar adecuadamente el tiempo de vida de una aserción sin correr un riesgo considerable de que expire antes de arribar.

El elemento <OneTimeUse> indica que la parte confiante debería utilizar inmediatamente la aserción y que no debe conservarla para su utilización ulterior. Las partes confiantes pueden solicitar una aserción fresca para cada uso en cualquier momento. Sin embargo, las implementaciones que deciden conservar aserciones para utilizarlas posteriormente deben respetar el elemento <OneTimeUse>. Esta condición es independiente de la información de condición NotBefore y NotOnOrAfter.

Para que una parte confiante pueda soportar la restricción de uso único, debería mantener una memoria asociada (caché) de las aserciones que ha procesado y que contienen dicha condición. Cuando se procesa una aserción con esta condición, se debería verificar la memoria asociada para asegurar que la misma aserción no ha sido recibida y procesada previamente por la parte confiante.

Una autoridad de SAML no debe incluir más de un elemento <OneTimeUse> en un elemento <Conditions> de una aserción.

A fin de determinar la validez del elemento <Conditions>, el elemento <OneTimeUse> siempre se considera válido. Es decir, esta condición no afecta la validez sino que se trata de una condición de utilización.

En el siguiente fragmento de esquema se define el elemento <OneTimeUse> y su tipo complejo **OneTimeUseType**:

```
<element name="OneTimeUse" type="saml:OneTimeUseType"/>
<complexType name="OneTimeUseType">
      <complexContent>
            <extension base="saml:ConditionAbstractType"/>
</complexContent>
</complexType>
```

8.1.5.6 Elemento <ProxyRestriction> (Restricción de mandatario)

Especifica las limitaciones que impone la parte asertante a las partes confiantes que desean actuar posteriormente como partes asertantes y expedir sus propias aserciones basándose en la información contenida en la aserción original. Una parte confiante que actúa como una parte asertante no debe expedir una aserción que contravenga por si misma las restricciones especificadas en esta condición en virtud de una aserción que contenga dicha condición.

El elemento <ProxyRestriction> incluye los siguientes elementos y atributos:

– Count (Cómputo) [Facultativo]

  Especifica el número máximo de direccionamientos autorizados por la parte asertante que pueden existir entre esta aserción y otra que ha sido expedida recientemente en virtud de ella.

– <Audience> (Audiencia) [Cero o varios]

Especifica el conjunto de audiencias a las que, con autorización de la parte asertante, se pueden expedir nuevas aserciones en virtud de esta aserción.

Un valor Count de cero indica que una parte confiante no debe expedir una aserción a otra parte confiante en virtud de esta aserción. Si el valor es mayor que cero, las aserciones emitidas de este modo deben contener un elemento <ProxyRestriction> con un valor Count que como máximo tendrá un número menos que este valor.

Si no se especifican elementos <Audience>, no se impondrán restricciones de audiencia a las partes confiantes a las que se podrán seguir expidiendo aserciones. Si se especifican, las aserciones expedidas han de contener un elemento <AudienceRestriction> con al menos uno de los elementos <Audience> incluido en el elemento <ProxyRestriction> anterior, y no deben contener elementos <Audience> que no estaban incluidos en el elemento <ProxyRestriction> anterior.

Una autoridad del SAML no debe incluir más de un elemento <ProxyRestriction> en un elemento <Conditions> de una aserción.

La condición <ProxyRestriction> se considera siempre válida para poder determinar la validez del elemento <Conditions>. Es decir, esta condición no afecta la validez pero representa una condición de utilización.

En el siguiente fragmento de esquema se define el elemento <ProxyRestriction> y su tipo complejo **ProxyRestrictionType**:

```
<element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>
<complexType name="ProxyRestrictionType">
      <complexContent>
            <extension base="saml:ConditionAbstractType">
                  <sequence>
                        <element ref="saml:Audience" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
                  <attribute name="Count" type="nonNegativeInteger"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

8.1.6 Asesoramiento

En esta cláusula se definen las construcciones del SAML que contienen información adicional relativa a una aserción que una parte asertante desea proporcionar a una parte confiante.

El elemento <Advice> incluye cualquier información adicional que la autoridad del SAML desee proporcionar. Las aplicaciones pueden ignorar esta información sin afectar ni la semántica ni la validez de la aserción.

El elemento <Advice> contiene una combinación de cero o varios elementos <Assertion>, <EncryptedAssertion>, <AssertionIDRef> y <AssertionURIRef> y elementos calificados por espacio de nombre en otros espacios de nombre que no pertenecen al SAML.

A continuación se presentan algunas aplicaciones posibles del elemento <Advice>:

- Incluir prueba que soporte las reivindicaciones de la aserción que serán citadas, bien sea directamente (mediante incorporación de las reivindicaciones) o indirectamente (por referencia a las aserciones de soporte).
- Establecer una prueba de las reivindicaciones de la aserción.
- Especificar los puntos de temporización y distribución necesarios para las actualizaciones de la aserción.

En el siguiente fragmento de esquema se define el elemento <Advice> y su tipo complejo **AdviceType**:

```
<element name="Advice" type="saml:AdviceType"/>
<complexType name="AdviceType">
      <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
            <any namespace="##other" processContents="lax"/>
      </choice>
</complexType>
```

8.1.7 Enunciados

Todos los enunciados que se definen en el SAML están asociados con un sujeto. Por lo general, las aserciones del SAML se refieren a un **subject**, que se representa mediante el elemento `<Subject>`. No obstante, este último es facultativo y por lo tanto, otras especificaciones y perfiles pueden hacer uso de la estructura de la aserción SAML para emitir enunciados similares sin necesidad de especificar un sujeto, o quizás especificándolo de un modo alternativo. En las siguientes subcláusulas se definen las construcciones que contienen la información del enunciado.

8.1.7.1 Elemento <Statement> (Enunciado)

El elemento `<Statement>` representa un punto de extensión para que otras aplicaciones basadas en aserción puedan reutilizar el marco de aserciones del SAML. El mismo SAML deduce sus enunciados principales de este punto de extensión. Su tipo complejo **StatementAbstractType** es abstracto y por consiguiente sólo puede ser empleado como la base de un tipo derivado.

En el siguiente fragmento de esquema se define el elemento `<Statement>` y su tipo complejo **StatementAbstractType**:

```
<element name="Statement" type="saml:StatementAbstractType"/>
<complexType name="StatementAbstractType" abstract="true"/>
```

8.1.7.2 Elemento <AuthnStatement> (Enunciado de autenticación)

El elemento `<AuthnStatement>` describe un enunciado mediante la autoridad del SAML afirmando que el sujeto de la aserción fue autenticado mediante un mecanismo particular en un momento específico. Las aserciones que contienen elementos `<AuthnStatement>` han de incluir un elemento `<Subject>`.

Se trata del tipo **AuthnStatementType**, que extiende el **StatementAbstractType** con la adición de los siguientes elementos y atributos:

NOTA – En la versión V2.0 del SAML se suprimieron del `<AuthnStatement>` el elemento `<AuthorityBinding>` (vinculación de autoridad) y su tipo correspondiente.

– `AuthnInstant` (Instante de autenticación) [Obligatorio]

  Especifica el instante en el que se realiza la autenticación. El valor de tiempo se codifica en UTC, conforme a 7.3.

– `SessionIndex` (Índice de la sesión) [Facultativo]

  Especifica el índice de una sesión particular entre el principal identificado por el sujeto y la autoridad de autenticación.

– `SessionNotOnOrAfter` (Ni en ese instante ni después de la sesión) [Facultativo]

  Especifica un instante en el que debe considerarse finalizada la sesión entre el principal identificado por el sujeto y la autoridad que emite el enunciado. El valor de tiempo se codifica en UTC, conforme a 7.3. No es necesaria ninguna relación entre este atributo y un atributo de condición `NotOnOrAfter` que pueda estar incluido en la aserción.

– `<SubjectLocality>` (Localidad del sujeto) [Facultativo]

  Especifica el nombre de dominio DNS y la dirección IP del sistema desde el que se autenticó, aparentemente, el sujeto de la aserción.

– `<AuthnContext>` (Contexto de autenticación) [Obligatorio]

  Contexto utilizado por la autoridad de autenticación que incluye el evento de autenticación que produjo este enunciado. Contiene una referencia de clase de contexto de autenticación, una declaración de contexto de autenticación o referencia de declaración, o ambas. Véase la cláusula 12 (Contexto de autenticación) para encontrar una descripción completa de la información relativa al contexto de autenticación.

En general, cualquier valor de cadena puede ser empleado como un valor `SessionIndex`. Ahora bien, cuando la privacidad es un factor importante, se debe tener especial precaución para garantizar que el valor de `SessionIndex` no invalida otros mecanismos de privacidad. Consecuentemente, no es recomendable utilizar este valor para correlacionar la actividad de un principal a través de diferentes participantes en la sesión. A continuación se recomiendan dos soluciones para lograr este cometido:

- Utilizar enteros positivos pequeños (o las constantes que aparecen varias veces en una lista) para el `SessionIndex`. La autoridad del SAML debe escoger la gama de valores de tal forma que la cardinalidad de cualquier entero sea suficientemente alta para impedir que las acciones de un principal particular tengan que ser correlacionadas a través de múltiples participantes en la sesión, y debe elegir

aleatoriamente los valores del `SessionIndex` de dicho rango (salvo cuando se exige asegurar valores únicos para los enunciados subsiguientes que se proporcionan al mismo participante en la sesión pero como parte de una sesión distinta).

- Utilizar el valor del `ID` de la aserción circundante en el `SessionIndex`.

En el siguiente fragmento de esquema se define el elemento `<AuthnStatement>` y su tipo complejo **AuthnStatementType**:

```
<element name="AuthnStatement" type="saml:AuthnStatementType"/>
<complexType name="AuthnStatementType">
      <complexContent>
            <extension base="saml:StatementAbstractType">
                  <sequence>
                        <element ref="saml:SubjectLocality"
minOccurs="0"/>
                        <element ref="saml:AuthnContext"/>
                  </sequence>
                  <attribute name="AuthnInstant" type="dateTime"
use="required"/>
                  <attribute name="SessionIndex" type="string"
use="optional"/>
                  <attribute name="SessionNotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

8.1.7.2.1 Elemento <SubjectLocality> (Localidad del sujeto)

El elemento `<SubjectLocality>` especifica el nombre de dominio DNS y la dirección IP del sistema desde el que se autenticó el sujeto de la aserción. Tiene los siguientes atributos:

– `Address` (Dirección) [Facultativo]

Dirección de red del sistema desde la que se autenticó el principal identificado por el sujeto. Las direcciones IPv4 deben representarse en el formato con decimales separados por puntos (por ejemplo, "1.2.3.4"). Las direcciones IPv6 deben representarse conforme a RFC 3513, 2.2 del IETF (por ejemplo, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210").

– `DNSName` (Nombre de dominio DNS) [Facultativo]

Nombre de dominio DNS desde el que se autenticó el principal identificado por el sujeto.

Este elemento es meramente consultivo, ya que los dos campos pueden ser "suplantados" fácilmente, pero puede aportar información útil para algunas aplicaciones.

En el siguiente fragmento de esquema se define el elemento `<SubjectLocality>` y su tipo complejo **SubjectLocalityType**:

```
<element name="SubjectLocality" type="saml:SubjectLocalityType"/>
<complexType name="SubjectLocalityType">
      <attribute name="Address" type="string" use="optional"/>
      <attribute name="DNSName" type="string" use="optional"/>
</complexType>
```

8.1.7.2.2 Elemento <AuthnContext> (Contexto de autenticación)

El elemento `<AuthnContext>` especifica el contexto de un evento de autenticación. Puede contener una referencia de clase de contexto de autenticación, una declaración de contexto de autenticación o referencia de declaración, o ambas. Su tipo complejo **AuthnContextType** tiene los siguientes elementos:

– `<AuthnContextClassRef>` (Referencia de clase de contexto de autenticación) [Facultativo]

Referencia de URI que permite identificar una clase de contexto de autenticación que describe la declaración de contexto de autenticación que sigue.

– `<AuthnContextDecl>` (Declaración de contexto de autenticación) o `<AuthnContextDeclRef>` (Referencia de declaración de contexto de autenticación) [Facultativo]

Declaración de contexto de autenticación proporcionada por el valor, o una referencia de URI que permite identificar dicha declaración. Esta referencia puede ser determinada directamente en un documento XML que contenga la declaración referenciada.

– `<AuthenticatingAuthority>` (Autoridad de autenticación) [Cero o varios]

Cero o varios identificadores únicos de autoridades de autenticación que participaron en el proceso de autenticación del principal (sin incluir el expedidor de la aserción, quien se supone que participó sin haber sido nombrado explícitamente aquí).

Véase la cláusula 12 para encontrar una descripción completa de la información relativa al contexto de autenticación.

En el siguiente fragmento de esquema se define el elemento `<AuthnContext>` y su tipo complejo **AuthnContextType**:

```
<element name="AuthnContext" type="saml:AuthnContextType"/>
<complexType name="AuthnContextType">
      <sequence>
            <choice>
                  <sequence>
                        <element ref="saml:AuthnContextClassRef"/>
                        <choice minOccurs="0">
                              <element ref="saml:AuthnContextDecl"/>
                              <element ref="saml:AuthnContextDeclRef"/>
                        </choice>
                  </sequence>
                  <choice>
                        <element ref="saml:AuthnContextDecl"/>
                        <element ref="saml:AuthnContextDeclRef"/>
                  </choice>
            </choice>
            <element ref="saml:AuthenticatingAuthority" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
</complexType>
<element name="AuthnContextClassRef" type="anyURI"/>
<element name="AuthnContextDeclRef" type="anyURI"/>
<element name="AuthnContextDecl" type="anyType"/>
<element name="AuthenticatingAuthority" type="anyURI"/>
```

8.1.7.3 Elemento <AttributeStatement> (Enunciado de atributo)

El elemento `<AttributeStatement>` describe un enunciado de la autoridad del SAML afirmando que el sujeto de la aserción está asociado con los atributos especificados. Las aserciones que contienen elementos `<AttributeStatement>` deben incluir un elemento `<Subject>`.

Se trata del tipo **AttributeStatementType**, que extiende **StatementAbstractType** con la adición de los siguientes elementos:

– `<Attribute>` (Atributo) o `<EncryptedAttribute>` (Atributo criptado) [Uno o varios]

El elemento `<Attribute>` especifica un atributo del sujeto de la aserción. Se puede incluir un atributo SAML criptado con el elemento `<EncryptedAttribute>`.

En el siguiente fragmento de esquema se define el elemento `<AttributeStatement>` y su tipo complejo **AttributeStatementType**:

```
<element name="AttributeStatement" type="saml:AttributeStatementType"/>
<complexType name="AttributeStatementType">
      <complexContent>
            <extension base="saml:StatementAbstractType">
                  <choice maxOccurs="unbounded">
                        <element ref="saml:Attribute"/>
                        <element ref="saml:EncryptedAttribute"/>
                  </choice>
            </extension>
      </complexContent>
</complexType>
```

8.1.7.3.1 Elemento <Attribute> (Atributo)

El elemento `<Attribute>` permite identificar un atributo por nombre y facultativamente incluye su valor o valores. Tiene el tipo complejo **AttributeType**. Se emplea en un enunciado de atributo para expresar atributos y valores particulares asociados con un sujeto de aserción, como se describe en la cláusula anterior. Asimismo, se utiliza en una consulta de atributo para solicitar que se devuelvan los valores de los atributos SAML específicos. El elemento `<Attribute>` contiene los siguientes atributos XML:

– `Name` (Nombre) [Obligatorio]

  Nombre del atributo.

– `NameFormat` (Formato del nombre) [Facultativo]

  Referencia del URI que representa la clasificación del nombre del atributo para poder interpretar el nombre. En 8.7.2 se pueden encontrar algunas referencias de URI que pueden ser empleadas como el valor del atributo `NameFormat`, así como sus descripciones asociadas y las reglas de procesamiento. Si no se proporciona el valor de `NameFormat`, estará vigente el identificador `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`.

– `FriendlyName` (Nombre cómodo) [Facultativo]

  Cadena que ofrece un formato de nombre de atributo que puede ser leído con mayor facilidad por las personas, lo que puede ser útil cuando el nombre real es complejo u opaco, tal como un OID (definido en la Rec. UIT-T X.660) o un UUID (definido en la Rec. UIT-T X.667). Este valor de atributo no debe utilizarse como base de identificación formal de los atributos del SAML.

– Atributos arbitrarios

  Este tipo complejo aprovecha un punto de extensión `<xs:anyAttribute>` para poder añadir atributos XML arbitrarios a las construcciones `<Attribute>` sin necesidad de una extensión de esquema explícita. Esto permite añadir campos adicionales conforme se requiera para generar parámetros adicionales que pueden ser utilizados, por ejemplo, en una consulta de atributo. Las extensiones del SAML no deben añadir atributos XML locales (que no son calificados por espacio de nombre) o atributos XML calificados mediante un espacio de nombre definido por el SAML, al tipo complejo AttributeType o a uno de sus derivados; esos atributos se reservan para mejorar y mantener el propio SAML en el futuro.

– `<AttributeValue>` (Valor de atributo) [Cualquier número]

  Contiene un valor del atributo. Si un atributo incluye más de un valor discreto, se recomienda que cada valor aparezca en su propio elemento `<AttributeValue>`. Si un atributo recibe más de un elemento `<AttributeValue>`, y cualquiera de ellos tiene un datatype asignado mediante `xsi:type`, en ese caso, todos los elementos `<AttributeValue>` deben tener asignado el mismo datatype.

El significado de un elemento de atributo `<Attribute>` que no contiene elementos `<AttributeValue>` depende de su contexto. Si el atributo SAML está presente en un `<AttributeStatement>`, pero no tiene valores, se debe omitir el elemento `<AttributeValue>`. La ausencia de valores en una `<samlp:AttributeQuery>` indica que el peticionario está interesado en alguno o todos los valores de atributo nombrados (véase también 8.2).

Si los perfiles u otras especificaciones llevan a cabo otros usos del elemento `<Attribute>`, éstos deben definir la semántica de la especificación u omisión de los elementos `<AttributeValue>`.

En el siguiente fragmento de esquema se define el elemento `<Attribute>` y su tipo complejo **AttributeType**:

```
<element name="Attribute" type="saml:AttributeType"/>
<complexType name="AttributeType">
      <sequence>
            <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Name" type="string" use="required"/>
      <attribute name="NameFormat" type="anyURI" use="optional"/>
      <attribute name="FriendlyName" type="string" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

El elemento `<AttributeValue>` suministra el valor de un atributo SAML especificado. Se trata del tipo **xs:anyType**, el cual permite que en el contenido del elemento aparezca cualquier XML bien estructurado.

Si el contenido de datos de un elemento <AttributeValue> corresponde con un tipo único de esquema XML (tal como **xs:integer** o **xs:string**), se puede declarar el datatype explícitamente mediante una declaración xsi:type en el elemento <AttributeValue>. Si el valor del atributo contiene datos estructurados, los elementos de datos necesarios pueden ser definidos en un esquema de extensión.

NOTA – La especificación en <AttributeValue> de un datatype distinto de un tipo único de esquema XML utilizando xsi:type exige la presencia del esquema de extensión que define el datatype para que pueda continuar el procesamiento del esquema.

Si un atributo del SAML incluye un valor vacío, como es el caso de la cadena vacía, el elemento <AttributeValue> correspondiente debe estar vacío (por lo general esto se puede serializar como <AttributeValue/>). Lo anterior invalida el requisito en 7.1 que establece que los valores de cadena en el contenido del SAML deben incluir al menos un carácter que no sea un espacio en blanco (whitespace).

Si un atributo del SAML incluye un valor "null" (nulo), el elemento <AttributeValue> correspondiente debe estar vacío y ha de contener el atributo XML xsi:nil reservado con un valor "true" (verdadero) o "1".

En el siguiente fragmento de esquema se define el elemento <AttributeValue>:

```
<element name="AttributeValue" type="anyType" nillable="true"/>
```

8.1.7.3.2 Elemento <EncryptedAttribute> (Atributo criptado)

El elemento <EncryptedAttribute> representa un atributo del SAML en modo criptado conforme a las reglas de criptación del W3C. Este elemento contiene los siguientes elementos:

– <xenc:EncryptedData> [Obligatorio]

  El contenido criptado y los detalles de criptación asociados son conformes con las reglas de criptación del W3C. El atributo Type debe estar presente, y de ser así, ha de contener un valor de http://www.w3.org/2001/04/xmlenc#Element. El contenido criptado debe contener un elemento con un tipo **AttributeType** o derivado del mismo.

– <xenc:EncryptedKey> [Cero o varios]

  Claves de descriptación encapsuladas conforme a las reglas de criptación del W3C. Cada una de las claves encapsuladas debería incluir un atributo de destinatario en el que se especifique la entidad para la que se ha criptado la clave. El valor de dicho atributo debería ser el identificador URI de una entidad del sistema con un identificador de nombre del SAML, según se define en 8.7.

Los atributos criptados están previstos como una protección de la confidencialidad cuando el valor del texto explícito pasa por un intermediario.

En el siguiente fragmento de esquema se define el elemento <EncryptedAttribute>:

```
<element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
```

8.1.7.4 Elemento <AuthzDecisionStatement> (Enunciado de decisión de autorización)

El elemento <AuthzDecisionStatement> describe un enunciado mediante la autoridad del SAML afirmando que una petición de acceso del sujeto de la aserción al recurso señalado ha dado por resultado la decisión de autorización especificada basándose en alguna prueba especificada facultativamente. Las aserciones que contienen elementos <AuthzDecisionStatement> deben incluir un elemento <Subject>.

El recurso se identifica gracias a una referencia de URI. Para poder interpretar la aserción correctamente y con seguridad, la autoridad del SAML y la parte confiante del SAML han de interpretar cada referencia de URI de una manera congruente. Si no se logra una interpretación congruente de dicha referencia se pueden producir diferentes decisiones de autorización en función de la codificación de la referencia del URI del recurso. En RFC 2396, cláusula 6, del IETF se pueden encontrar reglas de normalización de las referencias de URI.

A fin de evitar posibles ambigüedades resultantes de las variaciones en la codificación del URI, las entidades del sistema SAML deberían aplicar, siempre que sea posible, el formato normalizado del URI como sigue:

- Las autoridades del SAML deberían codificar todas las referencias de URI de recursos en el formato normalizado.
- Las partes confiantes deberían convertir las referencias de URI de recursos al formato normalizado antes del procesamiento.

Las diferencias entre la sintaxis de la referencia de URI y la semántica de un sistema de ficheros subyacente puede producir la interpretación incongruente de la referencia de URI. Si las referencias de URI se emplean para especificar

un lenguaje de política de control de acceso, se tendrán que tomar precauciones especiales. El sistema que emplea aserciones del SAML debería cumplir con las siguientes condiciones de seguridad:

- Algunas partes de la sintaxis de la referencia de URI son sensibles a mayúsculas y minúsculas. Si el sistema de ficheros subyacente también lo es, un peticionario no debería lograr el acceso a un recurso denegado cambiando las minúsculas a mayúsculas o viceversa de una parte de la referencia del URI del recurso.
- Muchos de los sistemas de ficheros aceptan mecanismos tales como los trayectos lógicos y los enlaces simbólicos, que permiten a los usuarios establecer equivalencias lógicas entre los asientos en el sistema de ficheros. Un peticionario no debería lograr el acceso a un recurso denegado creando ese tipo de equivalencia.

El elemento `<AuthzDecisionStatement>` es del tipo **AuthzDecisionStatementType**, que extiende el **StatementAbstractType** al añadir los siguientes elementos y atributos:

– `Resource` (Recurso) [Obligatorio]

Referencia de URI que permite identificar el recurso al que se pretende una autorización de acceso. Este atributo puede tener el valor de la referencia de URI vacía (""), y el significado se define como "el inicio del documento actual", conforme a RFC 2396, 4.2, del IETF.

– `Decision` (Decisión) [Obligatorio]

Decisión adoptada por la autoridad del SAML con respecto al recurso específico. El valor es del tipo único **DecisionType**.

– `<Action>` (Acción) [Una o varias]

Conjunto de acciones cuya ejecución está autorizada en el recurso específicado.

– `<Evidence>` (Prueba) [Facultativo]

Conjunto de aserciones en las que se apoyó la autoridad del SAML para adoptar una decisión.

En el siguiente fragmento de esquema se define el elemento `<AuthzDecisionStatement>` y su tipo complejo **AuthzDecisionStatementType**:

```
<element name="AuthzDecisionStatement"
 type="saml:AuthzDecisionStatementType"/>
<complexType name="AuthzDecisionStatementType">
 <complexContent>
  <extension base="saml:StatementAbstractType">
        <sequence>
              <element ref="saml:Action" maxOccurs="unbounded"/>
              <element ref="saml:Evidence" minOccurs="0"/>
        </sequence>
        <attribute name="Resource" type="anyURI" use="required"/>
        <attribute name="Decision" type="saml:DecisionType" use="required"/>
  </extension>
 </complexContent>
</complexType>
```

8.1.7.4.1 Tipo único DecisionType (Tipo de decisión)

El tipo único **DecisionType** permite definir los posibles valores que han de comunicarse como progreso de un enunciado de decisión de autorización.

– `Permit` (Permiso)

La acción especificada está permitida.

– `Deny` (Denegación)

La acción especificada está denegada.

– `Indeterminate` (Indeterminado)

La autoridad del SAML no puede determinar si la acción especificada está permitida o denegada.

El valor decisión `indeterminada` se emplea cuando la autoridad del SAML necesita emitir un enunciado afirmativo y no puede tomar una decisión. La información adicional como por ejemplo el motivo del rechazo o la incapacidad de proporcionar una decisión puede ser devuelta como elementos `<StatusDetail>` en la `<Response>` circundante.

En el siguiente fragmento de esquema se define el tipo único **DecisionType**:

```
<simpleType name="DecisionType">
      <restriction base="string">
            <enumeration value="Permit"/>
            <enumeration value="Deny"/>
            <enumeration value="Indeterminate"/>
      </restriction>
</simpleType>
```

8.1.7.4.2 Elemento <Action> (Acción)

El elemento `<Action>` permite especificar una acción relativa al recurso especificado para la que se pretende un permiso. Su contenido de datos de cadena proporciona la etiqueta relativa a la acción que se pretende ejecutar en el recurso especificado, la cual tiene el siguiente atributo:

– `Namespace` (Espacio de nombre) [Facultativo]

Referencia de URI que representa el espacio de nombre en el que será interpretado el nombre de la acción especificada. Si este elemento está ausente, estará vigente el espacio de nombre `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation` especificado en 8.7.

NOTA (informativa) – En PE 36 (véase OASIS PE:2006) se sugiere reemplazar el texto anterior por:

`Namespace` [Obligatorio]

Referencia de URI que representa el espacio de nombre en el que será interpretado el nombre de la acción especificada.

En el siguiente fragmento de esquema se define el elemento `<Action>` y su tipo complejo **ActionType**:

```
<element name="Action" type="saml:ActionType"/>
<complexType name="ActionType">
 <simpleContent>
       <extension base="string">
             <attribute name="Namespace" type="anyURI" use="required"/>
       </extension>
 </simpleContent>
</complexType>
```

8.1.7.4.3 Elemento <Evidence> (Prueba)

El elemento `<Evidence>` contiene una o más aserciones o referencias de aserción en las que se apoyó la autoridad del SAML al emitir la decisión de autorización. Tiene el tipo complejo **EvidenceType** e incluye una combinación de uno o varios de los elementos:

– `<AssertionIDRef>` (Referencia de identificador de aserción) [Cualquier número]

Especifica una aserción por referencia al valor del atributo de `ID` de la aserción.

– `<AssertionURIRef>` (Referencia de URI de aserción) [Cualquier número]

Especifica una aserción mediante una referencia de URI.

– `<Assertion>` (Aserción) [Cualquier número]

Especifica una aserción mediante un valor.

– `<EncryptedAssertion>` (Aserción criptada) [Cualquier número]

Especifica una aserción criptada mediante un valor.

Cuando se proporciona una aserción como prueba se puede afectar el acuerdo de confianza existente entre la parte confiante del SAML y la autoridad del SAML que toma la decisión de autorización. Por ejemplo, si la parte confiante del SAML presenta una aserción a la autoridad del SAML en una petición, dicha autoridad puede utilizarla como prueba para tomar su decisión de autorización sin tener que endosar la validez de la aserción del elemento `<Evidence>` a la parte confiante o a cualquier otro tercero.

En el siguiente fragmento de esquema se define el elemento <Evidence> y su tipo complejo **EvidenceType**:

```
<element name="Evidence" type="saml:EvidenceType"/>
<complexType name="EvidenceType">
      <choice maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
      </choice>
</complexType>
```

8.2 Protocolos del SAML

Los mensajes de protocolo del SAML pueden ser generados e intercambiados gracias a una diversidad de protocolos. En la cláusula 10, Vinculaciones del SAML, se describen medios específicos de transporte de mensajes de protocolo utilizando protocolos de transporte ampliamente desplegados. En la cláusula 11, Perfiles del SAML, se describen varias aplicaciones de los protocolos que se definen en esta cláusula, así como reglas de procesamiento, restricciones y requisitos que facilitan el interfuncionamiento.

La petición del SAML específica y los mensajes de respuesta se deducen de los tipos comunes. El peticionario envía un elemento derivado de **RequestAbstractType** a un respondedor SAML, y éste genera un elemento que se adhiere al **StatusResponseType** o se deriva del mismo, como se ilustra en la figura 8-1.

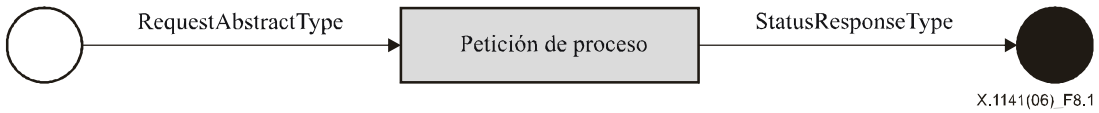

Figura 8-1/X.1141 – Protocolo de petición-respuesta del SAML

En algunos casos, cuando así lo permiten los perfiles, es posible generar una respuesta y enviarla aun cuando el respondedor no haya recibido la petición correspondiente.

Los protocolos definidos por el SAML pueden realizar las siguientes acciones:

- Devolver una o varias aserciones solicitadas. Esto puede suceder en respuesta a una petición directa de aserciones específicas o a una consulta de aserciones que satisfacen criterios particulares.
- Realizar una autenticación en virtud de una consulta y devolver la aserción correspondiente.
- Registrar un identificador de nombre o dar por terminado un registro de nombre en virtud de una petición.
- Recuperar un mensaje de protocolo que ha sido solicitado mediante un artefacto.
- Realizar un fin de sesión casi simultáneo de una colección de sesiones relacionadas ("fin de sesión única") en virtud de una petición.
- Proporcionar una correspondencia de identificadores de nombre en virtud de una petición.

En esta cláusula no se muestran las descripciones textuales de los elementos y tipos en el espacio de nombre del protocolo del SAML con el prefijo de espacio de nombre convencional `samlp`:. Para facilitar la comprensión, las descripciones textuales de los elementos y tipos en el espacio de nombre de la aserción SAML se indican con el prefijo de espacio de nombre convencional `saml`:.

8.2.1 Declaraciones de encabezamiento y espacio de nombre de esquema

En el siguiente fragmento de esquema se definen los espacios de nombre XML e información diversa de encabezamiento relativa al esquema de protocolo:

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
```

```
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-protocol-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New protocol schema based in a SAML V2.0 namespace.
     </documentation>
    </annotation>
…
</schema>
```

8.2.2 Peticiones y respuestas

En las subcláusulas a continuación se definen las construcciones y los requisitos básicos del SAML subyacentes a todos los mensajes de petición y respuesta que se emplean en los protocolos del SAML.

8.2.2.1 Tipo complejo RequestAbstractType (Tipo abstracto de petición)

Todas las peticiones del SAML son de los tipos derivados del tipo complejo **RequestAbstractType** abstracto. Este tipo permite definir atributos y elementos comunes que están asociados con todas las peticiones del SAML:

NOTA – En la versión V2.0 del SAML se ha suprimido el elemento `<RespondWith>` del tipo **`RequestAbstractType`**.

– `ID` (Identificador) [Obligatorio]

  Identificador de la petición de tipo **xs:ID** que debe seguir los requisitos especificados en 7.4 en cuanto a la singularidad del identificador. Es necesario que concuerden los valores del atributo `ID` en una petición y el atributo `InResponseTo` en la respuesta correspondiente.

– Version (Versión) [Obligatorio]

  Versión de esta petición. El identificador de la versión del SAML que se define en esta Recomendación es "2.0".

– `IssueInstant` (Instante de emisión) [Obligatorio]

  Instante en el que se emite la petición. El valor de tiempo se codifica en UTC conforme a 7.3.

– `Destination` (Destino) [Facultativo]

  Referencia de URI que indica la dirección a la que se ha enviado esta petición. Resulta útil para impedir la retransmisión malintencionada de peticiones a otros destinatarios, ya que se trata de una protección exigida por algunas vinculaciones de protocolo. Si está disponible, el destinatario objetivo verificará que la referencia de URI identifica el emplazamiento en el que se recibió el mensaje. Si no está disponible, la petición debe ser descartada. Algunas vinculaciones de protocolo pueden exigir la utilización de este atributo (véase la cláusula 10).

– `Consent` (Consentimiento) [Facultativo]

  Indica si se obtuvo o no el consentimiento (y las condiciones correspondientes) de un principal para el envío de esta petición. En 8.7.4 se proponen algunas referencias de URI que pueden ser aprovechados como valores del atributo `Consent`, así como de sus descripciones asociadas. Si no se proporciona un valor de `Consent`, el identificador `urn:oasis:names:tc:SAML:2.0:consent:unspecified` estará vigente.

– `<saml:Issuer>` [Facultativo]

  Identifica la entidad que generó el mensaje.

– `<ds:Signature>` [Facultativo]

  Firma XML que permite autenticar al peticionario y que proporciona integridad del mensaje, conforme a 8.4.

– `<Extensions>` (Extensiones) [Facultativo]

Este punto de extensión contiene elementos de extensión facultativos del mensaje de protocolo que han sido acordados entre las partes en comunicación. No se necesita un esquema de extensión para poder emplear este tipo de extensión, y si se dispone de uno de ellos, la configuración de validación lax no impone requisito alguno en cuanto a la validez de la extensión. Los elementos de extensión del SAML tienen que estar calificados por espacios de nombre definidos en un sistema ajeno al SAML.

Un peticionario del SAML, en función de los requisitos de protocolos o perfiles particulares, puede necesitar autoautenticarse a menudo, y es posible que también se requiera integridad del mensaje con cierta frecuencia. La autenticación y la integridad del mensaje pueden lograrse mediante mecanismos aportados por la vinculación de protocolo (véase la cláusula 10). La petición del SAML puede conllevar una firma que proporciona tanto la autenticación del peticionario como la integridad del mensaje.

Si se emplea una firma, el elemento `<ds:Signature>` debe estar presente, y el respondedor del SAML habrá de verificar la validez de la firma (es decir, que el mensaje no ha sido manipulado) conforme a las reglas de firma del W3C. Si la firma no es válida, el respondedor desconfiará del contenido de la petición y debería responder con un mensaje de error. Si la firma es válida, el respondedor debería evaluarla para determinar la identidad y la aplicabilidad del firmante, y poder continuar con el proceso de la petición o responder con un mensaje de error (si la petición no es válida por cualquier otro motivo).

Si está incluido un atributo `Consent` y su valor indica que se ha obtenido alguna forma de consentimiento del principal, la petición debería ser firmada.

Si un respondedor del SAML juzga que una petición no es válida de conformidad con la sintaxis del SAML o las reglas de procesamiento, si responde, debe devolver un mensaje de respuesta SAML con un elemento `<StatusCode>` con el valor `urn:oasis:names:tc:SAML:2.0:status:Requester`. En algunos casos, por ejemplo cuando existe la sospecha de un ataque por denegación de servicio, se puede garantizar que no habrá ninguna respuesta.

En el siguiente fragmento de esquema se define el tipo complejo **RequestAbstractType**:

```
<complexType name="RequestAbstractType" abstract="true">
      <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
      </sequence>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="Version" type="string" use="required"/>
      <attribute name="IssueInstant" type="dateTime" use="required"/>
      <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
<element name="Extensions" type="samlp:ExtensionsType"/>
<complexType name="ExtensionsType">
      <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
      </sequence>
</complexType>
```

8.2.2.2 Tipo complejo StatusResponseType (Tipo de respuesta de situación)

Todas las respuestas del SAML pertenecen a tipos derivados del tipo complejo **StatusResponseType**. Este tipo permite definir atributos y elementos comunes asociados con todas las respuestas del SAML:

– `ID` (Identificador) [Obligatorio]

Identificador de la respuesta del tipo **xs:ID** que debe cumplir con los requisitos especificados en 7.4 en cuanto a la singularidad del identificador.

– `InResponseTo` (En respuesta a) [Facultativo]

Referencia al identificador de la petición a la que corresponde la respuesta, si la hubiere. Si no se genera la respuesta correspondiente a una petición, o no se puede determinar el valor del atributo `ID` de una petición (por ejemplo, cuando la petición está mal estructurada), este atributo no debe estar presente. De lo contrario, el atributo debe estar presente y su valor ha de concordar con el valor del atributo `ID` de la petición correspondiente.

– `Version` (Versión) [Obligatorio]

  Versión de esta respuesta. El identificador de la versión del SAML que se define en esta Recomendación es "2.0".

– `IssueInstant` (Instante de emisión) [Obligatorio]

  Instante de emisión de la respuesta. El valor temporal se codifica en UTC, conforme a 7.3.

– `Destination` (Destino) [Facultativo]

  Referencia de URI que indica la dirección a la que se ha de enviar esta respuesta. Resulta útil para impedir la retransmisión malintencionada de respuestas a otros destinatarios, ya que se trata de una protección exigida por algunas vinculaciones de protocolo. Si está disponible, el destinatario objetivo verificará que la referencia de URI identifica el emplazamiento en el que se recibió el mensaje. Si no lo está, la petición debe ser descartada. Algunas vinculaciones de protocolo pueden exigir la utilización de este atributo (véase la cláusula 10).

– `Consent` (Consentimiento) [Facultativo]

  Indica si se obtuvo o no el consentimiento (y las condiciones correspondientes) de un principal para el envío de esta respuesta. En 8.7.4 se proponen algunas referencias de URI que pueden aprovecharse como valores del atributo `Consent`, así como sus descripciones asociadas. Si no se proporciona un valor de `Consent`, el identificador `urn:oasis:names:tc:SAML:2.0:consent:unspecified` (véase 8.7.4) estará vigente.

– `<saml:Issuer>` [Facultativo]

  Identifica la entidad que generó el mensaje. (Véase 8.1.2.5 para obtener más información relativa a este elemento).

– `<ds:Signature>` [Facultativo]

  Firma XML que permite autenticar el respondedor y que protege la integridad del mensaje, conforme a 8.4.

– `<Extensions>` (Extensiones) [Facultativo]

  Este punto de extensión contiene elementos de extensión facultativos del mensaje de protocolo que han sido acordados entre las partes en comunicación. No se necesita un esquema de extensión para poder emplear este tipo de extensión, y si se dispone de uno de ellos, la configuración de validación lax no impone requisito alguno en cuanto a la validez de la extensión. Los elementos de extensión del SAML tienen que estar calificados por espacio de nombre en un espacio de nombre definido en un sistema ajeno al SAML.

– `<Status>` (Situación) [Requerido]

  Código que representa la situación de la petición correspondiente.

Un respondedor del SAML, en función de los requisitos de protocolos o perfiles particulares, puede necesitar autoautenticarse a menudo, y es posible que también se requiera integridad del mensaje con cierta frecuencia. La autenticación y la integridad del mensaje pueden lograrse mediante mecanismos aportados por la vinculación de protocolo. La respuesta del SAML puede conllevar una firma que protege tanto la autenticación del respondedor como la integridad del mensaje.

Si se emplea una firma, el elemento `<ds:Signature>` debe estar presente, y el peticionario del SAML que recibe la respuesta habrá de verificar la validez de la firma (es decir, que el mensaje no ha sido manipulado) conforme a las reglas de firma XML del W3C. Si la firma no es válida, el respondedor desconfiará del contenido de la respuesta y debería tratarla como un error. Si la firma es válida, el peticionario debería evaluarla para determinar la identidad y la aplicabilidad del firmante, y poder continuar con el proceso de la respuesta conforme juzgue conveniente.

Si está incluido un atributo `Consent` y su valor indica que se ha obtenido alguna forma de consentimiento del principal, la respuesta debería estar firmada.

En el siguiente fragmento de esquema se define el tipo complejo **StatusResponseType**:

```
<complexType name="StatusResponseType">
      <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
            <element ref="samlp:Status"/>
      </sequence>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="InResponseTo" type="NCName" use="optional"/>
      <attribute name="Version" type="string" use="required"/>
```

```
      <attribute name="IssueInstant" type="dateTime" use="required"/>
      <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
```

1) Elemento <Status> (Situación)

El elemento <Status> contiene los elementos:

– <StatusCode> (Código de situación) [Obligatorio]

  Código que representa la condición de la actividad realizada en respuesta a la petición correspondiente.

– <StatusMessage> (Mensaje de situación) [Facultativo]

  Mensaje que puede ser devuelto a un operador.

– <StatusDetail> (Detalle de la situación) [Facultativo]

  Información adicional relativa a la condición de la petición.

En el siguiente fragmento de esquema se define el elemento <Status> y su tipo complejo **StatusType**:

```
<element name="Status" type="samlp:StatusType"/>
<complexType name="StatusType">
   <sequence>
      <element ref="samlp:StatusCode"/>
      <element ref="samlp:StatusMessage" minOccurs="0"/>
      <element ref="samlp:StatusDetail" minOccurs="0"/>
   </sequence>
</complexType>
```

2) Elemento <StatusCode> (Código de situación)

El elemento <StatusCode> especifica un código o un conjunto de códigos anidados que representan la situación de la petición correspondiente. Este elemento tiene los siguientes elementos y atributos:

– Value (Valor) [Obligatorio]

  Valor del código de situación. Este atributo contiene una referencia de URI. El valor más alto del elemento <StatusCode> debe obtenerse de la lista de nivel superior en esta cláusula.

– <StatusCode> (Código de situación) [Facultativo]

  Código de situación subordinado que proporciona información más específica relativa a una situación de error. Los respondedores pueden omitir estos códigos a fin de impedir ataques con los que se pretende sondear información adicional presentando intencionalmente peticiones erróneas.

Los valores de <StatusCode> válidos de nivel superior son:

```
urn:oasis:names:tc:SAML:2.0:status:Success
```

La petición ha tenido éxito. La información adicional puede ser devuelta en los elementos <StatusMessage> y/o <StatusDetail>.

```
urn:oasis:names:tc:SAML:2.0:status:Requester
```

La petición no pudo llevarse a cabo debido a un error cometido por el peticionario.

```
urn:oasis:names:tc:SAML:2.0:status:Responder
```

La petición no pudo llevarse a cabo debido a un error cometido por el respondedor del SAML o la autoridad del SAML.

```
urn:oasis:names:tc:SAML:2.0:status:VersionMismatch
```

El respondedor del SAML no pudo procesar la petición porque la versión del mensaje de petición era incorrecto.

En diversas partes de esta Recomendación se hace referencia a los siguientes códigos de situación de segundo nivel. Es posible que en versiones subsiguientes de la Recomendación del SAML se definan códigos de situación de segundo nivel adicionales. Las entidades del sistema podrán proponer libremente más códigos de situación específicos mediante la definición de las referencias de URI pertinentes.

```
urn:oasis:names:tc:SAML:2.0:status:AuthnFailed
```

El proveedor respondedor no pudo autenticar el principal satisfactoriamente.

```
urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue
```

En un elemento <saml:Attribute> o <saml:AttributeValue> se encontró contenido inesperado o no válido.

```
urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy
```

El proveedor respondedor no puede o no podrá aceptar la política del identificador de nombre solicitada.

```
urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext
```

El respondedor no puede satisfacer los requisitos del contexto de autenticación especificado.

```
urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP
```

Un intermediario puede utilizar este código para señalar que no se puede determinar ninguno de los elementos <Loc> de proveedor de identidad soportados en una <IDPList> o que no está disponible ninguno de los proveedores de identidad soportados.

```
urn:oasis:names:tc:SAML:2.0:status:NoPassive
```

Indica que el proveedor respondedor no puede autenticar al principal pasivamente, tal y como fue solicitado.

```
urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP
```

Un intermediario puede utilizar este código para señalar que no puede soportar a ninguno de los proveedores incluidos en <IDPList>.

```
urn:oasis:names:tc:SAML:2.0:status:PartialLogout
```

La autoridad de la sesión puede utilizar este código para indicar a un participante en la misma que no le fue posible distribuir el fin de sesión a los demás participantes en esa sesión.

```
urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded
```

Indica que un proveedor respondedor no puede autenticar al principal directamente y que no está autorizado para seguir retransmitiendo la petición a través de un mandatario (proxy).

```
urn:oasis:names:tc:SAML:2.0:status:RequestDenied
```

El respondedor o la autoridad del SAML tiene la capacidad para procesar la petición pero ha decidido no responder. Este código de situación puede aplicarse cuando el contexto de seguridad del mensaje de petición o la secuencia de los mensajes de petición recibidos de un peticionario específico plantee una preocupación importante.

```
urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported
```

El respondedor o la autoridad del SAML no soporta la petición.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated
```

El respondedor del SAML no puede procesar ninguna petición con la versión de protocolo especificada en la petición.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh
```

El respondedor del SAML no puede procesar la petición porque la versión del protocolo especificada en el mensaje de petición es una actualización muy importante de la versión más alta del protocolo que puede soportar el respondedor.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow
```

El respondedor no puede procesar la petición porque la versión del protocolo especificada en el mensaje de petición es demasiado baja.

```
urn:oasis:names:tc:SAML:2.0:status:ResourceNotRecognized
```

El valor del recurso proporcionado en el mensaje de petición no es válido o no se puede reconocer.

```
urn:oasis:names:tc:SAML:2.0:status:TooManyResponses
```

El mensaje de respuesta debería contener más elementos que los que puede devolver el respondedor del SAML.

```
urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile
```

A una entidad que no tiene conocimiento de un perfil de atributo particular se le ha presentado un atributo extraído de ese perfil.

```
urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal
```

El proveedor respondedor no reconoce al principal especificado en la petición o implícito en ella.

```
urn:oasis:names:tc:SAML:2.0:status:UnsupportedBinding
```

El respondedor del SAML no puede satisfacer adecuadamente la petición con la vinculación de protocolo especificada en la petición.

En el siguiente fragmento de esquema se define el elemento <StatusCode> y su tipo complejo **StatusCodeType**:

```
<element name="StatusCode" type="samlp:StatusCodeType"/>
<complexType name="StatusCodeType">
   <sequence>
      <element ref="samlp:StatusCode" minOccurs="0"/>
   </sequence>
   <attribute name="Value" type="anyURI" use="required"/>
</complexType>
```

3) Elemento <StatusMessage> (Mensaje de situación)

El elemento <StatusMessage> permite especificar un mensaje que puede ser devuelto a un operador:

En el siguiente fragmento de esquema se define el elemento <StatusMessage>:

```
<element name="StatusMessage" type="string"/>
```

4) Elemento <StatusDetail> (Detalle de la situación)

El elemento <StatusDetail> es útil para especificar información adicional relativa a la situación de la petición. La información adicional consta de cero o varios elementos de cualquier espacio de nombre, y no necesita la inclusión de un esquema o la validación del esquema del contenido de <StatusDetail>.

En el siguiente fragmento de esquema se define el elemento <StatusDetail> y su tipo complejo **StatusDetailType**:

```
<element name="StatusDetail" type="samlp:StatusDetailType"/>
<complexType name="StatusDetailType">
   <sequence>
      <any namespace="##any" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

8.2.3 Consulta de aserción y protocolo de petición

En esta cláusula se definen los mensajes y las reglas de procesamiento necesarias para solicitar las aserciones las aserciones existentes por referencia o consultándolas por sujeto y tipo de enunciado .

8.2.3.1 Elemento <AssertionIDRequest> (Petición de identificador de aserción)

Si el peticionario conoce el identificador único de una o varias aserciones, puede emplearse el elemento de mensaje `<AssertionIDRequest>` para solicitar que sean devueltos en un mensaje `<Response>`. El elemento `<saml:AssertionIDRef>` se usa para especificar cada una de las aserciones que será devuelta.

En el siguiente fragmento de esquema se define el elemento `<AssertionIDRequest>`:

```
<element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>
<complexType name="AssertionIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:AssertionIDRef"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

8.2.3.2 Consultas

En las siguientes subcláusulas se definen los mensajes de petición de consulta del SAML.

8.2.3.2.1 Elemento <SubjectQuery> (Consulta de sujeto)

El elemento del mensaje `<SubjectQuery>` representa un punto de extensión que define nuevas consultas del SAML en las que se especifica un sujeto SAML único. Su tipo complejo **SubjectQueryAbstractType** es abstracto y por consiguiente sólo puede ser utilizado como la base de un tipo derivado. El tipo **SubjectQueryAbstractType** añade el elemento `<saml:Subject>` (definido en 8.1.4) al tipo **RequestAbstractType**.

En el siguiente fragmento de esquema se define el elemento `<SubjectQuery>` y su tipo complejo **SubjectQueryAbstractType**:

```
<element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>
<complexType name="SubjectQueryAbstractType" abstract="true">
     <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:Subject"/>
                  </sequence>
            </extension>
     </complexContent>
</complexType>
```

8.2.3.2.2 Elemento <AuthnQuery> (Consulta de autenticación)

El elemento de mensaje `<AuthnQuery>` se emplea para realizar la consulta "¿Cuales son las aserciones que contienen enunciados de autenticación y están disponibles para este sujeto?" Una `<Response>` satisfactoria ha de incluir una o varias aserciones que contienen enunciados de autenticación.

No se debe emplear el mensaje `<AuthnQuery>` como una petición de una nueva autenticación mediante las referencias de identidad proporcionadas en la petición. `<AuthnQuery>` representa una petición de enunciados acerca de actos de autenticación que han tenido lugar durante una interacción anterior entre el sujeto indicado y la autoridad de autenticación.

Este elemento es del tipo **AuthnQueryType**, que extiende el tipo **SubjectQueryAbstractType** con la adición de los siguientes elementos y atributos:

– `SessionIndex` (Índice de sesión) [Facultativo]

  Si está presente, especifica un filtro adecuado para las posibles respuestas. Este tipo de consulta formula la pregunta "¿Cuáles son las aserciones que contienen enunciados de autenticación y que están disponibles para este sujeto en el contexto de la información suministrada relativa a la sesión?"

– `<RequestedAuthnContext>` (Contexto de autenticación solicitado) [Facultativo]

  Si está presente, especifica un filtro adecuado para las posibles respuestas. Este tipo de consulta formula la pregunta "¿Cuáles son las aserciones que contienen enunciados de autenticación, que están disponibles para este sujeto, y que pueden colmar los requisitos del contexto de autenticación de este elemento?"

Una autoridad del SAML, en respuesta a una consulta de autenticación, devuelve aserciones con enunciados de autenticación de la siguiente manera:

- Reglas establecidas en 8.2.3.4 relativas a la adaptación con el elemento `<Subject>` de la consulta para identificar las aserciones que pueden ser devueltas.
- Si el atributo `SessionIndex` está presente en la consulta, al menos un elemento `<AuthnStatement>` en el conjunto de las aserciones devueltas debe contener un atributo `SessionIndex` que concuerde con el mismo atributo en la consulta. Este atributo es facultativo para todo el conjunto de esas aserciones concordantes que han de ser devueltas en la respuesta.
- Si el elemento `<RequestedAuthnContext>` está presente en la consulta, al menos un elemento `<AuthnStatement>` en el conjunto de las aserciones devueltas debe contener un elemento `<AuthnContext>` que satisfaga el elemento en la consulta. Este atributo es facultativo para todo el conjunto de esas aserciones concordantes que han de ser devueltas en la respuesta.

En el siguiente fragmento de esquema se define el elemento `<AuthnQuery>` y su tipo complejo **AuthnQueryType**:

```
<element name="AuthnQuery" type="samlp:AuthnQueryType"/>
<complexType name="AuthnQueryType">
      <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                  <sequence>
                        <element ref="samlp:RequestedAuthnContext"
minOccurs="0"/>
                  </sequence>
                  <attribute name="SessionIndex" type="string"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

1) Elemento <RequestedAuthnContext> (Contexto de autorización solicitado)

El elemento `<RequestedAuthnContext>` permite especificar los requisitos del contexto de autenticación de los enunciados de autenticación devueltos en la respuesta que corresponde a una petición o consulta. Su tipo complejo **RequestedAuthnContextType** define los siguientes elementos y atributos:

– `<saml:AuthnContextClassRef>` o `<saml:AuthnContextDeclRef>` [Uno o varios]

Especifica una o varias referencias de URI que identifican las clases o declaraciones de contexto de autenticación. Estos elementos se definen en 8.1.7.2.2. En la cláusula 12 figura más información acerca de las clases de contexto de autenticación.

– `Comparison` (Comparación) [Facultativo]

Especifica el método de comparación aplicado para evaluar las clases o enunciados de contexto solicitados, es decir, uno de: "`exact`" (exacto), "`minimum`" (mínimo), "`maximum`" (máximo) o "`better`" (mejor). El criterio por defecto es "`exact`".

Se puede emplear un conjunto de referencias de clase o de referencias de declaración. El conjunto de las referencias proporcionadas debe evaluarse como un conjunto ordenado, donde el primer elemento es la clase o declaración de contexto de autenticación preferida. Si no es posible satisfacer alguna de las clases o declaraciones especificadas conforme a las reglas a continuación, el respondedor devolverá un mensaje `<Response>` con el siguiente `<StatusCode>` de segundo nivel: `urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext`.

Si el atributo `Comparison` se fija a "`exact`" o se omite, el contexto de autenticación resultante en el enunciado de autenticación debe concordar exactamente con al menos uno de los contextos de autenticación especificados.

Si el atributo `Comparison` se fija a "`minimum`", el contexto de autenticación resultante en el enunciado de autenticación debe ser al menos tan estricto (a juicio del respondedor) como uno de los contextos de autenticación especificados.

Si el atributo `Comparison` se fija a "`better`", el contexto de autenticación resultante en el enunciado de autenticación debe ser más estricto (a juicio del respondedor) que cualquiera de los contextos de autenticación especificados.

Si el atributo `Comparison` se fija a "`maximum`", el contexto de autenticación resultante en el enunciado de autenticación debe ser tan estricto como sea posible (a juicio del respondedor) sin exceder la rigidez de al menos uno de los contextos de autenticación especificados.

En el siguiente fragmento de esquema se define el elemento <RequestedAuthnContext> y su tipo complejo **RequestedAuthnContextType**:

```
<element name="RequestedAuthnContext" type="samlp:RequestedAuthnContextType"/>
<complexType name="RequestedAuthnContextType">
  <choice>
        <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
        <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
  </choice>
  <attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
</complexType>
<simpleType name="AuthnContextComparisonType">
  <restriction base="string">
        <enumeration value="exact"/>
        <enumeration value="minimum"/>
        <enumeration value="maximum"/>
        <enumeration value="better"/>
  </restriction>
</simpleType>
```

8.2.3.2.3 Elemento <AttributeQuery> (Consulta de atributo)

El elemento <AttributeQuery> se emplea para formular la consulta "Devuélvanse los atributos solicitados para este sujeto". Una respuesta satisfactoria tendrá un formato con aserciones que contienen enunciados de atributo, en la medida permitida por la política. Este elemento es de tipo **AttributeQueryType**, que extiende **SubjectQueryAbstractType** con la adición del siguiente elemento:

– <saml:Attribute> [Cualquier número]

  Cada elemento <saml:Attribute> especifica un atributo cuyos valores serán devueltos. Si no se especifican atributos, ello indica que se solicitan todos los atributos autorizados por la política. Si en la respuesta se devuelve un elemento del atributo <saml:Attribute> determinado que contiene uno o varios elementos <saml:AttributeValue>, éste no debe contener ningún valor diferente de los valores especificados en la consulta. Si los perfiles o atributos particulares no han especificado reglas de igualdad, ésta se define como una representación XML idéntica al valor. En 8.1.7.3.1 figura más información acerca de <saml:Attribute>.

Una consulta única no debe contener dos elementos <saml:Attribute> con los mismos valores de Name (nombre) y NameFormat (formato de nombre) (es decir, un atributo determinado sólo debe ser nombrado una vez en una consulta).

Una autoridad del SAML, en respuesta a una consulta de atributo, devuelve aserciones con enunciados de atributo como se indica a continuación:

- Reglas establecidas en 8.2.3.4 relativas a la adaptación con el elemento <Subject> de la consulta para identificar las aserciones que pueden ser devueltas.
- Si hay uno o varios elementos <Attribute> presentes en la consulta, éstos restringen o filtran los atributos y facultativamente los valores devueltos, como se mencionó antes.
- Los atributos y los valores devueltos también pueden estar restringidos por consideraciones de una política específica de la aplicación.

Los códigos de situación urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile y urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue de segundo nivel pueden ser empleados para señalar problemas referentes a la interpretación de información de atributo o valor en una consulta.

En el siguiente fragmento de esquema se define el elemento <AttributeQuery> y su tipo complejo **AttributeQueryType**:

```
<element name="AttributeQuery" type="samlp:AttributeQueryType"/>
<complexType name="AttributeQueryType">
      <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                  <sequence>
                        <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

8.2.3.2.4 Elemento <AuthzDecisionQuery> (Consulta de decisión de autorización)

El elemento <AuthzDecisionQuery> se emplea para formular la consulta "Dada esta prueba, ¿deberían permitirse estas acciones en este recurso para este sujeto?" Una respuesta satisfactoria tendrá un formato con aserciones que contienen enunciados de decisión de autorización.

NOTA – La característica <AuthzDecisionQuery> se ha mantenido como definitiva a partir de la V2.0 del SAML, y no hay planes para mejorarla en el futuro. Los usuarios que necesiten funcionalidad adicional deberían considerar el lenguaje de etiquetas de control de acceso extensible (véase la Rec. UIT-T X.1142), que ofrece características de decisión de autorización mejoradas.

Este elemento es de tipo **AuthzDecisionQueryType**, que extiende **SubjectQueryAbstractType** con la adición de los siguientes elementos y atributos:

– `Resource` (Recurso) [Obligatorio]

  Referencia de URI que indica el recurso para el que se solicitó la autorización.

– `<saml:Action>` [Uno o varios]

  Acciones para las que se solicitó la autorización. En 8.1.7.4.2 figura más información acerca de este elemento.

– `<saml:Evidence>` [Facultativo]

  Conjunto de aserciones en las que puede confiar una autoridad del SAML para poder tomar su decisión de autorización. En 8.1.7.4.3 figura más información acerca de este elemento.

Una autoridad del SAML, en respuesta a una consulta de decisión de autorización, devuelve aserciones con enunciados de decisión de autorización, de la siguiente manera:

- Reglas establecidas en 8.2.3.4 relativas a la adaptación con el elemento <Subject> de la consulta para identificar las aserciones que pueden ser devueltas.

En el siguiente fragmento de esquema se define el elemento <AuthzDecisionQuery> y su tipo complejo **AuthzDecisionQueryType**:

```
<element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>
<complexType name="AuthzDecisionQueryType">
  <complexContent>
        <extension base="samlp:SubjectQueryAbstractType">
              <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
              </sequence>
              <attribute name="Resource" type="anyURI" use="required"/>
        </extension>
  </complexContent>
</complexType>
```

8.2.3.3 Elemento <Response> (Respuesta)

El elemento de mensaje <Response> se emplea cuando una respuesta consiste en una lista de cero o varias aserciones que satisfacen la petición. Tiene el tipo complejo **ResponseType**, que extiende **StatusResponseType** y añade los siguientes elementos:

– `<saml:Assertion>` o `<saml:EncryptedAssertion>` [Cualquier número]

  Especifica una aserción mediante un valor o facultativamente una aserción criptada también por medio de un valor. En 8.1.3.3 figura más información acerca de estos elementos.

En el siguiente fragmento de esquema se define el elemento <Response> y su tipo complejo **ResponseType**:

```
<element name="Response" type="samlp:ResponseType"/>
<complexType name="ResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                  <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="saml:Assertion"/>
                        <element ref="saml:EncryptedAssertion"/>
                  </choice>
            </extension>
      </complexContent>
</complexType>
```

8.2.3.4 Reglas de procesamiento

Cada aserción devuelta por una autoridad del SAML, en respuesta a un mensaje de consulta definido en el SAML, debe contener un elemento <saml:Subject> que **concuerda estrictamente** con el elemento <saml:Subject> en la consulta.

Un elemento <saml:Subject> S1 concuerda estrictamente con S2 únicamente si se cumplen las dos condiciones siguientes:

- Si S2 incluye un elemento identificador (<BaseID>, <NameID>, o <EncryptedID>), S1 ha de incluir un elemento identificador idéntico, aunque el elemento puede criptarse (o no) en S1 o S2. En otras palabras, la forma descriptada del identificador debe ser idéntica en S1 y S2. "Idéntica" significa que el contenido del elemento identificador y los valores de atributo deben ser los mismos. De acuerdo con esta definición, un identificador criptado, una vez descriptado, será idéntico al original.
- Si S2 incluye uno o varios elementos <saml:SubjectConfirmation>, S1 ha de incluir al menos un elemento <saml:SubjectConfirmation> de modo que S1 pueda ser confirmado en la forma descrita por al menos un elemento <saml:SubjectConfirmation> en S2.

Como un ejemplo de lo que está permitido y lo que no lo está, S1 podría contener un elemento <saml:NameID> con un valor Format particular, y S2 podría contener un elemento <saml:EncryptedID> que resulta al criptar el elemento <saml:NameID> de S1. No obstante, S1 y S2 no pueden contener un elemento <saml:NameID> con diferentes valores de Format y contenido de elemento, aun cuando se considera que ambos identificadores hacen referencia al mismo principal.

Si la autoridad del SAML no puede proporcionar una aserción con algún enunciado que satisfaga las restricciones expresadas mediante una referencia de consulta o aserción, el elemento <Response> no debe contener un elemento <Assertion> y ha de incluir un elemento <StatusCode> con el valor urn:oasis:names:tc:SAML:2.0:status:Success.

Es necesario respetar el resto de las reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

8.2.4 Protocolo de petición de autenticación

Cuando un principal (o un agente que actúa en su nombre) desea obtener aserciones que contienen enunciados de autenticación para establecer un contexto de seguridad en una o varias partes confiantes, puede aprovechar el protocolo de petición de autenticación para emitir un elemento de mensaje <AuthnRequest> a una autoridad del SAML y solicitarle que devuelva un mensaje <Response> que incluya una o varias aserciones de ese tipo. Dichas aserciones pueden contener enunciados adicionales de cualquier tipo, pero al menos una de ellas debe incluir como mínimo un enunciado de autenticación. Si una autoridad del SAML soporta este protocolo se le denomina también proveedor de identidad.

Aparte de dicho requisito, el contenido específico de las aserciones devueltas dependerá del perfil o contexto de uso. Asimismo, no se especifican los medios exactos necesarios para que el proveedor de identidad pueda autenticar al principal o al agente; por consecuencia, dichos medios pueden repercutir en el contenido de la respuesta. Además, quedan fuera del alcance de este protocolo otras cuestiones relacionadas con la validación de las referencias de autenticación por parte del proveedor de identidad o cualquier comunicación entre éste y algunas otras entidades que participan en el proceso de autenticación.

En las subcláusulas siguientes, las descripciones y reglas de procesamiento hacen referencia a los siguientes participantes, muchos de los cuales pueden ser la misma entidad en un perfil de uso particular:

– Peticionario

  Entidad que origina la petición de autenticación y recibe la respuesta devuelta.

– Presentador

  Entidad que presenta la petición al proveedor de identidad y que se autoautentica durante la transmisión del mensaje, o bien se apoya en un contexto de seguridad determinado para establecer su identidad. El peticionario o el presentador actúa como intermediario entre el primero y el proveedor de identidad que responde.

– Sujeto solicitado

  Entidad para la que se solicitan una o varias aserciones.

– Entidad asertante

  Entidad o entidades que probablemente podrán satisfacer uno de los elementos `<SubjectConfirmation>` de la aserción o aserciones resultantes.

– Parte confiante

  Entidad o entidades que probablemente podrán aceptar la aserción o aserciones para dar cumplimiento a una finalidad definida por el perfil o contexto de uso, por lo general para establecer un contexto de seguridad.

– Proveedor de identidad

  Entidad que recibe la petición del presentador y que envía la respuesta al presentador.

Elemento <AuthnRequest> (Petición de autorización)

Un presentador autentica a un proveedor de identidad para poder solicitarle que expida una aserción con un enunciado de autenticación (o se apoya en un contexto de seguridad disponible) y le envía un mensaje `<AuthnRequest>` en el que se describen las propiedades que exige la aserción resultante para poder cumplir con su finalidad. Entre dichas propiedades puede encontrarse información relacionada con el contenido de la aserción y/o relacionada con la forma en la que se debería entregar el mensaje `<Response>` resultante al peticionario. El proceso de autenticación del presentador puede realizarse antes, durante o después de la entrega inicial del mensaje `<AuthnRequest>`.

Si, por ejemplo, el peticionario es una parte confiante que trata de utilizar la aserción resultante para autenticar o autorizar al sujeto solicitado de modo que la parte confiante pueda decidir si proporciona un servicio, el peticionario podría no ser el presentador de la petición.

El mensaje `<AuthnRequest>` debería estar firmado o autenticado y la integridad debería estar protegida por la vinculación de protocolo empleada para entregar el mensaje.

Este mensaje tiene el tipo complejo **AuthnRequestType**, que extiende el tipo **RequestAbstractType** y añade los siguientes elementos y atributos, todos los cuales son facultativos en general, pero algunos perfiles específicos podrían exigirlos:

– `<saml:Subject>` [Facultativo]

  Especifica el sujeto solicitado de la aserción o aserciones resultantes. Esto puede incluir uno o varios elementos `<saml:SubjectConfirmation>` para indicar como se pueden confirmar las aserciones resultantes y/o quien se encargará de ello. En 8.1.4 figura más información sobre este elemento.

  Si el identificador se omite o no está incluido, se supondrá que el presentador del mensaje es el sujeto solicitado. Si no se incluyen elementos `<saml:SubjectConfirmation>`, se supondrá que el presentador es la única entidad atestiguadora necesaria y que el método está implícito por el perfil de uso y/o las políticas del proveedor de identidad.

– `<NameIDPolicy>` (Política de identificador de nombre) [Facultativo]

  Especifica las restricciones relativas al identificador de nombre que se emplearán para representar el sujeto solicitado. Si se omiten, en ese caso podrá aplicarse al sujeto solicitado cualquier tipo de identificador soportado por el proveedor de identidad, restringido por alguna de las políticas específicas de despliegue pertinentes, por ejemplo, con relación a la privacidad.

– `<saml:Conditions>` [Facultativo]

Especifica las condiciones del SAML en virtud de las cuales el peticionario espera limitar la validez y/o el uso de las aserciones resultantes. El respondedor puede modificar o suplementar este conjunto conforme lo juzgue necesario. La información en este elemento se utiliza como entrada al proceso de construcción de la aserción y no como condiciones de uso de la petición. (En 8.1.5 figura más información relativa a este elemento.)

– `<RequestedAuthnContext>` (Contexto de autenticación solicitado) [Facultativo]

Especifica los requisitos, si los hubiere, que el peticionario impondrá al contexto de autenticación que se aplica a la autenticación del presentador del proveedor respondedor.

– `<Scoping>` (Sondeo) [Facultativo]

Especifica un conjunto de proveedores de identidad en los que el peticionario deposita su confianza para autenticar al presentador, así como las limitaciones y el contexto relacionados con el envío de un mensaje `<AuthnRequest>` mediante un mandatario, del respondedor a los proveedores de identidad subsiguientes.

– `ForceAuthn` (Autorización forzada) [Facultativo]

Valor booleano. Si está fijado a "true" (verdadero), el proveedor de identidad debe autenticar al presentador directamente en lugar de apoyarse en un contexto de seguridad anterior. Si no se proporciona un valor, el valor por defecto es "false" (falso). No obstante, si `ForceAuthn` e `IsPassive` están fijados a "true", el proveedor de identidad no debe autenticar al presentador a menos que no se pueda dar cumplimiento a las restricciones de `IsPassive`.

– `IsPassive` [Facultativo]

Valor booleano. Si está fijado a "true", el proveedor de identidad y el propio agente del usuario no deben quitarle visiblemente el control de la interfaz del usuario al peticionario e interactuar con el presentador de una manera evidente. Si no se proporciona un valor, el valor por defecto es "false".

– `AssertionConsumerServiceIndex` (Índice de servicio del consumidor de la aserción) [Facultativo]

Identifica indirectamente el emplazamiento del peticionario al que debería devolverse el mensaje `<Response>`. Se aplica sólo a los perfiles donde el peticionario es diferente del presentador, como en el perfil del explorador Web SSO en esta Recomendación. El proveedor de identidad debe disponer de medios de confianza para establecer una correspondencia entre el valor del índice en el atributo y un emplazamiento asociado con el peticionario. En la cláusula 9 se describe un posible mecanismo de este tipo. Si se omite, el proveedor de identidad devolverá el mensaje `<Response>` al emplazamiento por defecto asociado con el peticionario del perfil de uso. Si el índice especificado no es válido, el proveedor de identidad puede devolver un error `<Response>` o puede emplear el emplazamiento por defecto. Este atributo es mutuamente exclusivo con los atributos `AssertionConsumerServiceURL` y `ProtocolBinding`.

– `AssertionConsumerServiceURL` (URL de servicio del consumidor de la aserción) [Facultativo]

Especifica, mediante un valor, el emplazamiento del peticionario al que ha de devolverse el mensaje `<Response>`. El respondedor tiene que garantizar de alguna manera que el valor especificado está asociado en realidad con el peticionario. En la cláusula 9 se describe un posible mecanismo de este tipo; la firma del mensaje circundante `<AuthnRequest>` constituye otro mecanismo. Este atributo es mutuamente exclusivo con el atributo `AssertionConsumerServiceIndex` y por lo general está acompañado por el atributo `ProtocolBinding`.

– `ProtocolBinding` (Vinculación de protocolo) [Facultativo]

Referencia de URI que identifica una vinculación del protocolo SAML que posibilitará devolver el mensaje `<Response>`. En la cláusula 10 figura más información acerca de las vinculaciones de protocolo y las referencias de URI que se han definido para ellas. Este atributo es mutuamente exclusivo con el atributo `AssertionConsumerServiceIndex` y por lo general está acompañado por el atributo `AssertionConsumerServiceURL`.

– `AttributeConsumingServiceIndex` (Índice de servicio del consumidor del atributo) [Facultativo]

Identifica indirectamente la información asociada con el peticionario que describe los atributos del SAML que desea o requiere que le sean suministrados por el proveedor de identidad en el mensaje `<Response>`. El proveedor de identidad debe disponer de medios de confianza para establecer una correspondencia entre el valor del índice en el atributo y la información asociada con el peticionario. En la cláusula 9 se describe un posible mecanismo de este tipo. El proveedor de identidad puede aprovechar esta información para rellenar uno o varios elementos `<saml:AttributeStatement>` en la aserción o aserciones que devuelve.

– `ProviderName` (Nombre del proveedor) [Facultativo]

Especifica el nombre del peticionario legible por las personas que será utilizado por el agente del usuario del presentador o el proveedor de identidad.

Véase 8.2.4.4 en cuanto a las reglas de procesamiento generales que se aplican a este mensaje.

En el siguiente fragmento de esquema se define el elemento `<AuthnRequest>` y su tipo complejo **AuthnRequestType**:

```
<element name="AuthnRequest" type="samlp:AuthnRequestType"/>
<complexType name="AuthnRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:Subject" minOccurs="0"/>
                        <element ref="samlp:NameIDPolicy" minOccurs="0"/>
                        <element ref="saml:Conditions" minOccurs="0"/>
                        <element ref="samlp:RequestedAuthnContext"
minOccurs="0"/>
                        <element ref="samlp:Scoping" minOccurs="0"/>
                  </sequence>
                  <attribute name="ForceAuthn" type="boolean"
use="optional"/>
                  <attribute name="IsPassive" type="boolean"
use="optional"/>
                  <attribute name="ProtocolBinding" type="anyURI"
use="optional"/>
                  <attribute name="AssertionConsumerServiceIndex"
type="unsignedShort" use="optional"/>
                  <attribute name="AssertionConsumerServiceURL"
type="anyURI" use="optional"/>
                  <attribute name="AttributeConsumingServiceIndex"
type="unsignedShort" use="optional"/>
                  <attribute name="ProviderName" type="string"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

8.2.4.1 Elemento <NameIDPolicy> (Política de identificador de nombre)

El elemento `<NameIDPolicy>` permite adaptar el identificador de nombre en los sujetos de las aserciones resultantes de una `<AuthnRequest>`. Su tipo complejo **NameIDPolicyType** define los siguientes atributos:

– `Format` (Formato) [Facultativo]

Especifica la referencia de URI que corresponde a un formato de identificador de nombre definido en esta u otra Recomendación (véanse algunos ejemplos en 8.7.3). El valor adicional de `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` se define específicamente para utilizarse en este atributo a fin de señalar una petición en el sentido de que debe criptarse el identificador resultante.

– `SPNameQualifier` (Calificador de nombre del SP) [Facultativo]

Especifica facultativamente que el identificador del sujeto de la aserción ha de ser devuelto (o creado) en el espacio de nombre de un proveedor de servicio distinto del peticionario o en el de un grupo de afiliación de proveedores de servicio. Véase por ejemplo la definición de `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` en esta Recomendación.

– `AllowCreate` (Permitir la creación) [Facultativo]

Valor booleano que es útil para indicar si el proveedor de identidad está autorizado, durante el proceso de cumplimiento de la petición, a crear un nuevo identificador para representar al principal. El valor por defecto es "false", y cuando se aplica, el peticionario restringe al proveedor de identidad a expedirle una aserción sólo si ya se ha establecido un identificador aceptable para el principal. Esto no impide que el proveedor de identidad pueda crear esos identificadores fuera del contexto de esta petición específica (por ejemplo, con antelación para un número considerable de principales).

NOTA 1 (informativa) – En PE14 (véase OASIS PE:2006) se aclara la definición anterior como sigue:

Valor booleano que es útil para indicar si el peticionario concede permiso al proveedor de identidad, durante el proceso de cumplimiento de la petición, para crear un nuevo identificador o para asociar un identificador existente que representa al principal con la parte confiante. Si no está presente este valor se utiliza el valor por defecto "false" o se omite todo el elemento.

NOTA 2 (informativa) – En PE14 (véase OASIS PE:2006) se sugiere añadir el siguiente texto al párrafo más adelante:

En algunas instalaciones el atributo `AllowCreate` se puede emplear para influenciar la creación del estado mantenido por el proveedor de identidad que corresponde a la utilización de un identificador de nombre (o cualquier otro persistente, que identifica los atributos de manera inequívoca) mediante una parte confiante particular, para fines tales como la creación dinámica de un identificador o atributo, el seguimiento de un consentimiento, la aplicación subsiguiente del protocolo de gestión de identificadores de nombre u otros propósitos conexos.

Cuando el atributo se fija a "false", el peticionario trata de restringir al proveedor de identidad de modo que sólo pueda expedir una aserción si ya se ha establecido dicho estado o si este último no considera conveniente utilizar un identificador. Sin embargo, esto no impide que el proveedor de identidad suponga que existe dicha información fuera del contexto de esta petición específica (por ejemplo, estableciéndola con antelación para un número considerable de principales).

Un valor "true" permite que el proveedor de identidad adopte las acciones correspondientes necesarias para satisfacer la petición, supeditadas a otras restricciones impuestas por la petición y la política (por ejemplo, el atributo `IsPassive`).

Los peticionarios, por lo general, no pueden suponer un comportamiento específico de los proveedores de identidad en lo que concierne a la creación o asociación inicial de los identificadores en su nombre, ya que se trata de detalles que se dejan para las implementaciones o las instalaciones. Salvo en el caso de perfiles específicos que rigen la utilización de este atributo, podría emplearse como un indicio para los proveedores de identidad acerca de la intención del peticionario en cuanto a almacenar el identificador o enlazarlo con un valor local.

En el atributo podría utilizarse un valor "false" para señalar que el peticionario no está preparado o no tiene la capacidad para llevar a cabo lo anterior, evitando así la pérdida de esfuerzo del proveedor de identidad.

Los peticionarios que no tienen una aplicación específica para este atributo deben, por lo general, fijarlo a "true" a fin de aumentar al máximo el interfuncionamiento. El atributo `AllowCreate` no debe ser utilizado en conjunto con peticiones o aserciones que se expiden con identificadores de nombre con un valor `Format` de `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` y por lo tanto no debería tenerse en cuenta (éstas excluyen cualquier estado de ese tipo relacionado con ellas).

Cuando se emplea este elemento, si el proveedor de identidad no comprende o no acepta el contenido, se devolverá un elemento de mensaje `<Response>` con un error `<Status>`, y puede contener un `<StatusCode>` de segundo nivel como `urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy`.

Si el valor `Format` se omite o se fija a `urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified`, el proveedor de identidad podrá devolver cualquier clase de identificador, supeditado a las restricciones adicionales en el contenido de este elemento o a las políticas del proveedor de identidad o del principal.

El valor `Format` especial `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` indica que la aserción o aserciones resultantes deben contener elementos `<EncryptedID>` en lugar de texto explícito. La forma no criptada del identificador de nombre subyacente puede ser de cualquier tipo soportado por el proveedor de identidad para el sujeto solicitado.

NOTA 3 (informativa) – En PE6 (véase OASIS PE:2006) se sugiere añadir el siguiente texto al final del párrafo anterior:

Si el proveedor de servicio exige criptación, no puede solicitar específicamente la devolución de una clase particular de identificador. Se puede emplear el elemento de metadatos `<md:NameIDFormat>` que se describe en la cláusula 9 u otro medio fuera de banda para determinar qué clase de identificador se debe criptar y devolver.

NOTA 4 (informativa) – En PE15 (véase OASIS PE:2006) se sugiere añadir el siguiente párrafo:

Cuando se emplea un valor `Format` definido en 8.7.3.7 distinto de `urn:oasis:names:TC:SAML:2.0:nameid-format:unspecified` o `urn:oasis:names:TC:SAML:2.0:nameid-format:encrypted`, si el proveedor de identidad devuelve alguna o algunas aserciones:

- el valor `Format` del `<NameID>` en el `<Subject>` de cualquier `<Assertion>` ha de ser idéntico al valor `Format` proporcionado en `<NameIDPolicy>`; y
- si no se omite el `SPNameQualifier` en `<NameIDPolicy>`, el valor del `SPNameQualifier` del `<NameID>` en el `<Subject>` de cualquier `<Assertion>` ha de ser idéntico al valor del `SPNameQualifier` proporcionado en `<NameIDPolicy>`.

Independientemente del valor `Format` en `<NameIDPolicy>`, el proveedor de identidad puede devolver un `<EncryptedID>` en el sujeto de la aserción resultante si las políticas en vigor en el proveedor de identidad (posiblemente específicas para el proveedor de servicio) exigen la utilización de un identificador criptado.

Si el peticionario desea autorizar que el proveedor de identidad establezca un nuevo identificador para el principal en caso de que no exista ninguno, debe incluir este elemento con el atributo `AllowCreate` fijado a "true". De no ser así, un principal sólo podrá ser autenticado satisfactoriamente cuando el proveedor de identidad le haya establecido antes un identificador que pueda ser utilizado por el peticionario. Esto resulta útil en primer lugar en conjunto con el valor `Format urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`, (véase la cláusula 12).

NOTA 5 (informativa) – En PE14 (véase OASIS PE:2006) se sugiere no tener en cuenta el párrafo anterior.

En el siguiente fragmento de esquema se define el elemento `<NameIDPolicy>` y su tipo complejo **NameIDPolicyType**:

```
<element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
<complexType name="NameIDPolicyType">
      <attribute name="Format" type="anyURI" use="optional"/>
      <attribute name="SPNameQualifier" type="string" use="optional"/>
      <attribute name="AllowCreate" type="boolean" use="optional"/>
</complexType>
```

8.2.4.2 Elemento <Scoping> (Sondeo)

El elemento `<Scoping>` permite especificar los proveedores de identidad que son de confianza para el peticionario a fin de autenticar el presentador, y las limitaciones y el contexto relacionados con el envío del mensaje `<AuthnRequest>` mediante un mandatario del respondedor a los proveedores de identidad subsiguientes. Su tipo complejo **ScopingType** define los siguientes elementos y atributos:

– `ProxyCount` (Cómputo de mandatarios) [Facultativo]

Especifica el número de indirecciones (direccionamientos que emplean un mandatario, nombre, referencia, etc. en lugar de un valor) permitidas que se realizan a través de un mandatario entre el proveedor de identidad que recibe esta `<AuthnRequest>` y el proveedor de identidad que se encarga de autenticar el principal. Un cómputo igual a cero prohíbe el empleo de mandatarios, mientras que la omisión de este atributo no expresa esa restricción.

– `<IDPList>` (Lista de los IDP) [Facultativo]

Lista consultiva de los proveedores de identidad e información asociada en la que se establece que el peticionario acepta responder a la petición.

– `<RequesterID>` (Identificador del peticionario) [Cero o varios]

Identifica el conjunto de entidades peticionarias en cuyo nombre actúa el peticionario. Se emplea para comunicar la cadena de peticionarios cuando se utiliza un mandatario, como se describe en 8.2.4.5. En 8.7.3.6 figura una descripción de los identificadores de entidad.

En los perfiles que especifican un intermediario activo, éste puede examinar la lista y devolver un mensaje `<Response>` con un error `<Status>` y un valor de `<StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP` o `urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP` si no puede ponerse en contacto con alguno de los proveedores de identidad especificados o no soporta a ninguno de ellos.

En el siguiente fragmento de esquema se define el elemento `<Scoping>` y su tipo complejo **ScopingType**:

```
<element name="Scoping" type="samlp:ScopingType"/>
<complexType name="ScopingType">
      <sequence>
            <element ref="samlp:IDPList" minOccurs="0"/>
            <element ref="samlp:RequesterID" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="ProxyCount" type="nonNegativeInteger"
use="optional"/>
</complexType>
<element name="RequesterID" type="anyURI"/>
```

8.2.4.3 Elemento <IDPList> (Lista de IDP)

El elemento <IDPList> permite especificar los proveedores de identidad que son de confianza para el peticionario a fin de autenticar el presentador. Su tipo complejo **IDPListType** define los siguientes elementos:

– <IDPEntry> (Asiento de IDP) [Uno o varios]

Información relativa a un solo proveedor de identidad.

– <GetComplete> (Obtener la lista completa) [Facultativo]

Si la <IDPList> está incompleta, se puede recuperar completa mediante una referencia de URI especificada por este elemento. La recuperación del recurso asociado con el URI debe arrojar un ejemplar XML cuyo elemento raíz es una <IDPList> que no contiene por sí misma un elemento <GetComplete>.

En el siguiente fragmento de esquema se define el elemento <IDPList> y su tipo complejo **IDPListType**:

```
<element name="IDPList" type="samlp:IDPListType"/>
<complexType name="IDPListType">
      <sequence>
            <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
            <element ref="samlp:GetComplete" minOccurs="0"/>
      </sequence>
</complexType>
<element name="GetComplete" type="anyURI"/>
```

El elemento <IDPEntry> permite especificar un solo proveedor de identidad que es de confianza para el peticionario a fin de autenticar el presentador. Su tipo complejo **IDPEntryType** define los siguientes atributos:

– ProviderID (Identificador de proveedor) [Obligatorio]

Identificador único del proveedor de identidad. En 8.7.3.6 figura una descripción de esos identificadores.

– Name (Nombre) [Facultativo]

Nombre legible por las personas para el proveedor de identidad.

– Loc (Emplazamiento) [Facultativo]

Referencia de URI que representa el emplazamiento de un punto extremo específico de perfil que soporta el protocolo de petición de autenticación. El perfil de uso debe comprender la vinculación que se ha de utilizar.

En el siguiente fragmento de esquema se define el elemento <IDPEntry> y su tipo complejo **IDPEntryType**:

```
<element name="IDPEntry" type="samlp:IDPEntryType"/>
<complexType name="IDPEntryType">
      <attribute name="ProviderID" type="anyURI" use="required"/>
      <attribute name="Name" type="string" use="optional"/>
      <attribute name="Loc" type="anyURI" use="optional"/>
</complexType>
```

8.2.4.4 Reglas de procesamiento

El intercambio de <AuthnRequest> y <Response> soporta una diversidad de casos de utilización y, por consecuencia, generalmente se concibe de modo que se pueda utilizar en un contexto específico en el cual se restringe este carácter facultativo y se exigen o prohíben clases específicas de entradas y salidas. Las siguientes reglas de procesamiento se aplican como un comportamiento invariante en cualquier perfil de este intercambio de protocolo. También deben respetarse las demás reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

El respondedor debe, en última instancia, responder a una <AuthnRequest> con un mensaje <Response> que incluye una o varias aserciones para satisfacer las especificaciones que se definen en la petición, o con un mensaje <Response> que contiene un <Status> en el que se describe el error ocurrido. El respondedor puede llevar a cabo intercambios de mensajes adicionales con el presentador, según proceda, para iniciar o completar el proceso de autenticación, en función de la naturaleza de la vinculación de protocolo y del mecanismo de autenticación. Como se describe en la cláusula que sigue, esto incluye el envío de la petición a través de un mandatario, dirigiendo el presentador a otro proveedor de identidad mediante la emisión de su propio mensaje <AuthnRequest>, de modo que la aserción resultante pueda emplearse para autenticar el presentador ante el respondedor original, aprovechando, de hecho, el SAML como mecanismo de autenticación.

Si el respondedor no tiene la capacidad para autenticar al presentador, no puede reconocer el sujeto solicitado o las políticas en vigor en el proveedor de identidad le impiden expedir una aserción (por ejemplo, el sujeto objetivo ha prohibido que el proveedor de identidad expida aserciones a la parte confiante), éste debe devolver un mensaje `<Response>` con un error `<Status>`, y puede incluir un valor `<StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:AuthnFailed`; o `urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal`.

Si el elemento `<saml:Subject>` está incluido en la petición, las aserciones resultantes `<saml:Subject>` deben **coincidir estrictamente** con dicha petición, como se describe en 8.2.3.4, salvo que el identificador pueda tener un formato distinto si está especificado por `<NameIDPolicy>`. En ese caso, el contenido físico del identificador puede diferir, pero debe referirse al mismo principal.

Todo el contenido que se define específicamente en `<AuthnRequest>` es facultativo, aunque es posible que algunos perfiles requieran parte del mismo. En ausencia de contenido específico, queda implícito el siguiente comportamiento:

- La aserción o aserciones devueltas han de contener un elemento `<saml:Subject>` que representa al presentador. El proveedor de identidad determina el tipo y formato del identificador. Al menos un enunciado en al menos una aserción ha de ser un `<saml:AuthnStatement>` que describe la autenticación realizada por el respondedor o el servicio de autenticación asociado con él.
- El presentador de la petición debería ser, en la medida posible, la única entidad atestiguadora que puede satisfacer la `<saml:SubjectConfirmation>` de la aserción o aserciones. Si se emplean métodos de confirmación que no son estrictos, se aplicarán mecanismos específicos de vinculación u otros para ayudar a satisfacer este requisito.
- La aserción o aserciones resultantes han de contener un elemento `<saml:AudienceRestriction>` que hace referencia al peticionario como una parte confiante aceptable. A criterio del proveedor de identidad se pueden incluir otras audiencias.

8.2.4.5 Autorizar o Habilitar (Proxying)

Si un proveedor de identidad recibe una `<AuthnRequest>` sin haber autenticado aún al presentador o sin poder autenticarlo directamente, pero cree que otro proveedor de identidad o un equivalente externo al SAML ya ha llevado a cabo dicha autenticación, puede dar respuesta a la petición emitiendo una nueva `<AuthnRequest>` en su nombre que será presentada al otro proveedor de identidad, o una petición en cualquier formato externo al SAML que pueda reconocer la entidad. El proveedor de identidad original se denomina proveedor de identidad autorizado (proxying).

Cuando el proveedor autorizado recibe una respuesta `<Response>` (o su equivalente en un sistema ajeno al SAML) satisfactoria, la aserción incluida o el equivalente en un sistema ajeno al SAML puede ser utilizada para autenticar al presentador de manera que el proveedor habilitado pueda expedir una aserción propia en respuesta a la `<AuthnRequest>` original, concluyendo el intercambio de mensajes. Ambos proveedores de identidad, el de autenticación y el autorizado, pueden incluir restricciones en lo que concierne a la actividad de los mensajes y aserciones que expiden a través de mandatarios, como se describe en las subcláusulas anteriores y en las que siguen.

El peticionario puede influenciar el comportamiento del mandatario (proxy) incluyendo el elemento `<Scoping>` donde el proveedor fija un valor `ProxyCount` deseado y/o indica una lista de proveedores de identidad preferidos a los que se les puede habilitar incluyendo una `<IDPList>` ordenada de los proveedores preferidos.

Un proveedor de identidad puede controlar el empleo secundario de sus aserciones al habilitar proveedores de identidad mediante un elemento `<ProxyRestriction>` en las aserciones que expide.

Un proveedor de identidad puede autorizar una `<AuthnRequest>` si se omite el atributo `<ProxyCount>` o si es mayor que cero. La decisión de autorizar o no es una cuestión de política local. Un proveedor de identidad puede decidir habilitar a un proveedor especificado en la `<IDPList>`, si está incluido, pero no está obligado a ello.

Un proveedor de identidad no debe autorizar una petición donde `<ProxyCount>` tenga un valor cero. En este caso, el proveedor de identidad devolverá un error `<Status>` con un valor `<StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded`, a menos que pueda autenticar directamente al presentador.

Si, cuando se crea la nueva `<AuthnRequest>`, se decide habilitar a un proveedor de identidad del SAML, éste debe incluir formas equivalentes o más estrictas de toda la información incluida en la petición original (como el contexto de política de autenticación). Sin embargo, el proveedor habilitado puede especificar cualquier `<NameIDPolicy>` que desee para aumentar al máximo las oportunidades de obtener una respuesta satisfactoria.

Si el proveedor de identidad que autentica no pertenece al SAML, el proveedor habilitado debe disponer de algún otro medio para garantizar que los elementos que controlan la interacción del agente del usuario (por ejemplo, `<IsPassive>`) serán aceptados por el proveedor de autenticación.

La nueva <AuthnRequest> debe incluir un atributo <ProxyCount> con un valor máximo igual al valor original menos uno. Si la petición original no contiene un atributo <ProxyCount>, la nueva petición si debería incluirlo.

Si se especificó una <IDPList> en la petición original, la nueva petición también ha de incluirla. El proveedor de identidad habilitado puede añadir proveedores de identidad adicionales al final de la <IDPList>, pero no debe suprimir ninguno de la lista.

La petición y la respuesta de autenticación son procesadas en la forma normal, conforme a las reglas que figuran en esta cláusula y en el perfil de uso. Cuando el proveedor de identidad autorizado concluye la autenticación del presentador (en el caso del SAML se indica mediante un mensaje <Response>), se siguen estos pasos:

- El proveedor de identidad habilitado prepara una nueva aserción en su nombre insertando información pertinente copiada de la aserción original o del equivalente en el sistema ajeno al SAML.
- El nuevo <saml:Subject> de la aserción debe contener un identificador que satisfaga las preferencias del peticionario original, de conformidad con su elemento <NameIDPolicy>.
- En la nueva aserción, el enunciado <saml:AuthnStatement> ha de incluir un elemento <saml:AuthnContext> que contiene a su vez un elemento <saml:AuthenticatingAuthority> que señala el proveedor de identidad al cual fue referido el presentador por el proveedor de identidad habilitado. Si la aserción original contiene información <saml:AuthnContext> que incluye uno o varios elementos <saml:AuthenticatingAuthority>, éstos deberían ser incluidos en la nueva aserción con el elemento colocado atrás de ellos.
- Si el proveedor de identidad de autenticación no es un proveedor del SAML, el proveedor de identidad habilitado debe generar un valor de identificador único para el primero. Este valor debería ser congruente durante el tiempo entre diferentes peticiones. El valor no debe entrar en conflicto con valores utilizados o generados por otros proveedores del SAML.
- Cualquier otra información <saml:AuthnContext> podrá ser copiada, traducida u omitida de conformidad con las políticas del proveedor de identidad habilitado, siempre que se dé cumplimiento a los requisitos originales establecidos por el peticionario.

Si en el futuro se solicita al proveedor de identidad que autentique el mismo presentador para un segundo peticionario, y esta petición es igual o menos estricta que la petición original (determinado por el proveedor de identidad habilitado), el proveedor de identidad podrá omitir la creación de una nueva <AuthnRequest> para el proveedor de identidad de autenticación y expedir inmediatamente otra aserción (suponiendo que la aserción original o su equivalente en un sistema ajeno al SAML recibido por el proveedor de identidad, aún es válido).

8.2.5 Protocolo de resolución de artefacto

Este protocolo ofrece un mecanismo que facilita que los mensajes del protocolo SAML puedan ser transportados en una vinculación SAML por referencia y no por valor. Al emplear este protocolo especializado se pueden obtener tanto las peticiones como las respuestas por referencia. El emisor del mensaje, en lugar de vincular un mensaje a un protocolo de transporte, envía una pieza pequeña de datos denominada artefacto aprovechando la vinculación. El artefacto puede adoptar varias formas, pero debe soportar un medio que permita que el receptor pueda determinar quién lo envío. Si el receptor lo desea, puede aplicar este protocolo junto con un protocolo de vinculación SAML diferente (generalmente síncrono) para resolver (encontrar) el artefacto en el mensaje de protocolo original.

La aplicación más común de este mecanismo es en las vinculaciones que no son capaces de transportar un mensaje fácilmente debido a las restricciones de tamaño, o para posibilitar la transmisión de un mensaje a través de un canal seguro entre el peticionario del SAML y el respondedor, evitando la necesidad de una firma.

Dependiendo de las características del mensaje subyacente que se ha de pasar por referencia, es posible que el protocolo de resolución de artefacto pueda exigir protecciones tales como autenticación mutua, protección de la integridad, confidencialidad, etc. de la vinculación de protocolo que se emplea para resolver el artefacto. El artefacto, en todos los casos, debe exponer una semántica de uso único de manera que una vez resuelto, ya no pueda ser utilizado por alguna de las partes.

Independientemente del mensaje de protocolo obtenido, el resultado al resolver un artefacto debe tratarse exactamente como si el mensaje obtenido hubiera sido enviado originalmente en lugar del artefacto.

8.2.5.1 Elemento <ArtifactResolve> (Resolución de artefacto)

El mensaje <ArtifactResolve> es útil para solicitar la devolución de un mensaje de protocolo SAML en un mensaje <ArtifactResponse> mediante la especificación de un artefacto que representa el mensaje de protocolo SAML. La transmisión original del artefacto está controlada por la vinculación de protocolo específica que se esté utilizando.

El mensaje <ArtifactResolve> debería estar firmado o autenticado, y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

Este mensaje tiene el tipo complejo ArtifactResolveType, que extiende el tipo RequestAbstractType y añade el siguiente elemento:

– <Artifact> (Artefacto) [Obligatorio]

Valor del artefacto recibido por el peticionario que ha de ser traducido al mensaje de protocolo que representa.

En el siguiente fragmento de esquema se define el elemento <ArtifactResolve> y su tipo complejo **ArtifactResolveType**:

```
<element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>
<complexType name="ArtifactResolveType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="samlp:Artifact"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="Artifact" type="string"/>
```

8.2.5.2 Elemento <ArtifactResponse> (Respuesta de artefacto)

El destinatario de un mensaje <ArtifactResolve> debe responder con un elemento de mensaje <ArtifactResponse>. Este elemento tiene el tipo complejo **ArtifactResponseType**, que extiende el tipo **StatusResponseType** con un solo elemento de uso variable (comodín) facultativo que corresponde al mensaje de protocolo SAML que ha de ser devuelto. Este mensaje encapsulado puede ser una petición o una respuesta.

El mensaje <ArtifactResponse> debería estar firmado o autenticado y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

En el siguiente fragmento de esquema se define el elemento <ArtifactResponse> y su tipo complejo **ArtifactResponseType**:

```
<element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>
<complexType name="ArtifactResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                  <sequence>
                        <any namespace="##any" processContents="lax"
minOccurs="0"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

8.2.5.3 Reglas de procesamiento

Si el respondedor reconoce la validez del artefacto, responderá con el mensaje de protocolo asociado en un elemento de mensaje <ArtifactResponse>. En otros casos, responderá con un elemento <ArtifactResponse> sin incorporar un mensaje. El elemento <Status>, en ambos casos, debe incluir un elemento <StatusCode> con el valor de código urn:oasis:names:tc:SAML:2.0:status:Success. En el resto de esta cláusula, un mensaje de respuesta sin un mensaje incorporado se denomina respuesta vacía.

El respondedor debe hacer cumplir la propiedad de utilización del artefacto una sola vez, garantizando que cualquier petición subsiguiente con el mismo artefacto proveniente de cualquier peticionario dará por resultado una respuesta vacía, como se describió antes.

Algunos mensajes de protocolo SAML, en particular el mensaje <AuthnRequest> en algunos perfiles, pueden tener por objetivo ser aceptados por cualquier parte que los reciba y pueda responder adecuadamente. En la mayoría de los demás casos, sin embargo, un mensaje está dirigido a una entidad específica. En esas circunstancias, el artefacto emitido debe estar asociado con el destinatario objetivo del mensaje que representa el artefacto. Si el emisor del artefacto recibe un mensaje <ArtifactResolve> de un peticionario que no puede autenticarse como el destinatario objetivo inicial, en ese caso, el emisor devolverá una respuesta vacía.

El emisor del artefacto debería hacer cumplir el límite de tiempo práctico más corto relativo a la posibilidad de uso de un artefacto, de modo que se disponga de una ventana de tiempo aceptable (pero no más) para que el receptor del artefacto pueda obtenerlo y se lo devuelva en un mensaje `<ArtifactResolve>`.

El atributo `InResponseTo` del mensaje `<ArtifactResponse>` debe contener el valor del atributo de ID del mensaje `<ArtifactResolve>` correspondiente, pero el mensaje de protocolo incorporado incluirá su propio identificador de mensaje, y en el caso de una respuesta incorporada, puede contener un valor `InResponseTo` diferente que corresponda al mensaje de petición original al que se responde con el mensaje incorporado.

Se tienen que respetar las demás reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

8.2.6 Protocolo de gestión del identificador de nombre

Un proveedor de identidad que desee cambiar, tras establecer un identificador de nombre para un principal, el valor y/o formato de dicho identificador que utilizará para hacer referencia al principal o para indicar que ya no se empleará un identificador de nombre para la misma referencia, comunicará a los proveedores de servicio el cambio enviándoles un mensaje `<ManageNameIDRequest>`.

NOTA 1 (informativa) – En PE12 (véase OASIS PE:2006) se identifica el propósito del párrafo anterior escribiéndolo nuevamente de la siguiente manera:

> Un proveedor de identidad que desee cambiar, tras establecer un identificador de nombre para un principal, el valor de dicho identificador que utilizará para hacer referencia al principal o para indicar que ya no se empleará un identificador de nombre para la misma referencia, comunicará a los proveedores de servicio el cambio enviándoles un mensaje `<ManageNameIDRequest>`.

Un proveedor de servicio también puede usar este mensaje para registrar o cambiar el valor de `SPProvidedID` que ha de incluirse cuando se emplea el identificador de nombre subyacente para comunicarse con él, o para dar por terminada la utilización de un identificador de nombre entre él mismo y el proveedor de identidad.

Por lo general, el protocolo no se emplea con identificadores de nombre "transitorios", ya que no se pretende gestionar su valor a largo plazo.

NOTA 2 (informativa) – En PE14 (véase OASIS PE:2006) se aclara el texto anterior de la siguiente manera:

> Este protocolo no debe utilizarse junto con el `Format <NameID>` `urn:oasis:names:tc:SAML:2.0:nameidformat:transient`.

8.2.6.1 Elemento <ManageNameIDRequest> (Petición de identificador de gestión de nombre)

Un proveedor envía un mensaje `<ManageNameIDRequest>` para comunicar al destinatario un cambio de identificador de nombre o para señalar que se deja de utilizar un identificador de nombre.

El mensaje `<ManageNameIDRequest>` debería estar firmado o autenticado, y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

Este mensaje tiene el tipo complejo **ManageNameIDRequestType**, que extiende el tipo **RequestAbstractType**, y añade los siguientes elementos:

– `<saml:NameID>` o `<saml:EncryptedID>` [Obligatorio]

  Identificador de nombre y datos descriptivos asociados (en texto explícito o formato criptado) que especifican el principal tal y como lo reconocían los proveedores de identidad y de servicio antes de esta petición (en 8.1.2 figura más información sobre estos elementos).

– `<NewID>` o `<NewEncryptedID>` o `<Terminate>` [Obligatorio]

  Valor del nuevo identificador (en texto explícito o formato criptado) que ha de utilizarse durante la comunicación con el proveedor solicitante relativa a este principal, o una indicación de que se dejó de usar el antiguo identificador. En el primer caso, si el peticionario es el proveedor de servicio, el nuevo identificador debe aparecer en elementos `<NameID>` subsiguientes en el atributo `SPProvidedID`. Si el peticionario es el proveedor de identidad, el nuevo valor aparecerá en elementos `<NameID>` subsiguientes como el contenido del elemento.

  NOTA (informativa) – En PE12 (véase OASIS PE:2006) se sugiere agregar lo siguiente al párrafo anterior:

  > En cualquier caso, si se emplea `<NewEncryptedID>`, su contenido criptado es simplemente un elemento `<NewID>` que contiene sólo el nuevo valor del identificador (una vez establecidos el formato y los calificadores, no pueden ser modificados).

En el siguiente fragmento de esquema se define el elemento <ManageNameIDRequest> y su tipo complejo **ManageNameIDRequestType**:

```
<element name="ManageNameIDRequest" type="samlp:ManageNameIDRequestType"/>
<complexType name="ManageNameIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <choice>
                              <element ref="saml:NameID"/>
                              <element ref="saml:EncryptedID"/>
                        </choice>
                        <choice>
                              <element ref="samlp:NewID"/>
                              <element ref="samlp:NewEncryptedID"/>
                              <element ref="samlp:Terminate"/>
                        </choice>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="NewID" type="string"/>
<element name="NewEncryptedID" type="saml:EncryptedElementType"/>
<element name="Terminate" type="samlp:TerminateType"/>
<complexType name="TerminateType"/>
```

8.2.6.2 Elemento <ManageNameIDResponse> (Respuesta de identificador de gestión de nombre)

El destinatario de un mensaje <ManageNameIDRequest> debe responder con un mensaje <ManageNameIDResponse> del tipo **StatusResponseType** sin contenido adicional.

El mensaje <ManageNameIDResponse> debería estar firmado o autenticado y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

En el siguiente fragmento de esquema se define el elemento <ManageNameIDResponse>:

```
<element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
```

8.2.6.3 Reglas de procesamiento

Si la petición incluye un <saml:NameID> (o una versión criptada) que no puede ser reconocido por el destinatario, el proveedor respondedor debe contestar con un error <Status> y puede incluir un valor <StatusCode> de segundo nivel de urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

NOTA 1 (informativa) – En PE14 (véase OASIS PE:2006) se aclara con mayor detalle el párrafo anterior. Véase el apéndice VIII para obtener más detalles.

Si el elemento <Terminate> está incluido en la petición, el proveedor solicitante está señalando que (en el caso de un proveedor de servicio) ya no aceptará aserciones del proveedor de identidad o (en el caso de un proveedor de identidad) ya no expedirá aserciones al proveedor de servicio acerca del principal. El proveedor receptor puede ejecutar cualquier tipo de mantenimiento con el conocimiento de que ya ha terminado la relación representada por el identificador de nombre. Puede decidir invalidar la sesión o sesiones activas de un principal cuya relación ya ha terminado.

NOTA 2 (informativa) – En PE8 (véase OASIS PE:2006) se sugiere sustituir la última oración de este párrafo por:

En general, no debería invalidar ninguna sesión o sesiones activas del principal cuya relación ya ha terminado. Si el proveedor receptor es un proveedor de identidad, no debería invalidar ninguna sesión o sesiones activas del principal establecidas con otros proveedores de servicio. Un proveedor solicitante puede enviar un mensaje <LogoutRequest> antes de iniciar la terminación del identificador de nombre enviando un mensaje <ManageNameIDRequest>, si esa es la intención del proveedor solicitante (por ejemplo, la terminación del identificador de nombre se inicia a través de un administrador que desea dar por terminada toda la actividad del usuario). El proveedor solicitante no debe enviar un mensaje <LogoutRequest> después de que se emite un mensaje <ManageNameIDRequest>.

Si el proveedor de servicio solicita el cambio del identificador del principal incluyendo un elemento <NewID> (o <NewEncryptedID>), el proveedor de identidad debe incluir el contenido del elemento como SPProvidedID cuando se comunica posteriormente con el proveedor de servicio en relación con este principal.

Si el proveedor de identidad solicita el cambio del identificador del principal incluyendo un elemento <NewID> (o <NewEncryptedID>), el proveedor de servicio debe utilizar el contenido del elemento como el contenido del elemento <saml:NameID> cuando se comunica posteriormente con el proveedor de identidad en relación con este principal.

Es posible criptar ninguno, alguno o ambos de los identificadores original y nuevo (empleando los elementos <EncryptedID> y <NewEncryptedID>).

En cualquier caso, el contenido de <saml:NameID> en la petición y su atributo asociado SPProvidedID han de contener la información del identificador de nombre más reciente establecida entre los proveedores para el principal.

En el caso de un identificador con un Format de urn:oasis:names:tc:SAML:2.0:nameid-format:persistent, el atributo NameQualifier debe contener el identificador único del proveedor de identidad que creó el identificador. Si el identificador fue establecido entre el proveedor de identidad y un grupo de afiliación al cual pertenece el proveedor de servicio como miembro, el atributo SPNameQualifier ha de contener el identificador único de dicho grupo. En los demás casos, éste debe contener el identificador único del proveedor de servicio. Se pueden omitir estos atributos si no concuerdan con el valor del elemento <Issuer> del mensaje del protocolo contenedor, pero no se recomienda debido al riesgo de confusión.

La modificación de esos identificadores puede consumir una cantidad de tiempo significativa para efectos de difusión a través de los sistemas del peticionario y del respondedor. En las implementaciones sería deseable permitir que cada parte acepte cualquier identificador por algún periodo de tiempo una vez completada satisfactoriamente la modificación de un identificador de nombre. Al no hacerlo de esa manera, se puede traducir en la incapacidad del principal para acceder a los recursos.

Se deben respetar las demás reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

8.2.7 Protocolo de fin de sesión único

Se trata de un protocolo de intercambio de mensajes que posibilita que todas las sesiones proporcionadas por una autoridad de sesión particular se den por terminadas casi simultáneamente. Se utiliza cuando un principal finaliza la sesión desde un participante en la sesión o cuando lo lleva a cabo directamente desde una autoridad de sesión. Asimismo, puede emplearse para que un principal dé por terminada una sesión debido a un fin de temporización. El motivo correspondiente al evento de fin de temporización puede ser indicado a través del atributo Reason (motivo).

Es posible que el principal haya establecido sesiones autenticadas tanto con la autoridad de sesión como con los participantes en una sesión individual, basándose en aserciones que contienen enunciados de autenticación suministrados por la autoridad de la sesión.

Cuando el principal solicita utilizar el proceso de fin de sesión único desde un participante en la sesión, este último debe enviar un mensaje <LogoutRequest> a la autoridad de la sesión que proporcionó la aserción que contiene el enunciado de autenticación relacionado con esa sesión.

Cuando el principal solicita ejecutar un fin de sesión desde la autoridad de sesión, o bien un participante en la sesión envía una petición de fin de sesión a esa autoridad especificando ese principal, dicha autoridad debería enviar un mensaje <LogoutRequest> a cada uno de los participantes en la sesión a los que haya proporcionado aserciones que contienen enunciados de autenticación durante su sesión actual con el principal, salvo el participante en la sesión que envió el mensaje <LogoutRequest> a la autoridad de sesión. Éste debería tratar de entrar en contacto con tantos de esos participantes como sea posible mediante la aplicación de este protocolo, dar por terminada su propia sesión con el principal y, por último, devolver un mensaje <LogoutResponse> al participante en la sesión solicitante, si lo hubiere.

8.2.7.1 Elemento <LogoutRequest> (Petición de fin de sesión)

Un participante en la sesión o una autoridad de sesión envía un mensaje <LogoutRequest> para señalar que ha terminado una sesión.

El mensaje <LogoutRequest> debería estar firmado o autenticado, y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

Este mensaje tiene el tipo complejo **LogoutRequestType**, que extiende el tipo **RequestAbstractType** y añade los siguientes elementos y atributos:

– NotOnOrAfter (Ni en ese instante ni después) [Facultativo]

  Instante en el que deja de tener validez la petición, tras el cual el destinatario puede descartar el mensaje. El valor temporal se codifica en UTC, conforme a 7.3.

– Reason (Motivo) [Facultativo]

  Indicación del motivo del fin de sesión en forma de una referencia de URI.

  NOTA 1 (informativa) – En PE10 (véase OASIS PE:2006) se sugiere sustituir el texto anterior por:

  El atributo Reason se especifica como una cadena en el esquema. Esta especificación restringe aún más el esquema exigiendo que el atributo Reason tenga la forma de una referencia de URI.

– `<saml:BaseID>` o `<saml:NameID>` o `<saml:EncryptedID>` [Obligatorio]

El identificador y los atributos asociados (en texto explícito o formato criptado) que especifican el principal tal y como lo reconocían los proveedores de identidad y de servicio antes de esta petición (en 8.1.2 figura más información sobre este elementos).

– `<SessionIndex>` (Índice de la sesión) [Facultativo]

Identificador que indexa esta sesión en el destinatario del mensaje.

NOTA 2 (informativa) – En PE38 (véase OASIS PE:2006) se aclara el texto anterior como sigue:

Índice de la sesión entre el principal identificado por el elemento `<saml:BaseID>`, `<saml:NameID>` o `<saml:EncryptedID>` y la autoridad de la sesión. Esto debe tener correlación con el atributo `SessionIndex`, si lo hubiere, en el enunciado `<saml:AuthnStatement>` de la aserción que permitió establecer la sesión que ha de ser terminada.

En el siguiente fragmento de esquema se define el elemento `<LogoutRequest>` y el tipo complejo **LogoutRequestType** asociado:

```
<element name="LogoutRequest" type="samlp:LogoutRequestType"/>
    <complexType name="LogoutRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:SessionIndex" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="Reason" type="string" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SessionIndex" type="string"/>
```

8.2.7.2 Elemento <LogoutResponse> (Respuesta de fin de sesión)

El destinatario de un mensaje `<LogoutRequest>` debe responder con un mensaje `<LogoutResponse>` de tipo **StatusResponseType**, sin especificar contenido adicional.

El mensaje `<LogoutResponse>` debería estar firmado o autenticado y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

En el siguiente fragmento de esquema se define el elemento `<LogoutResponse>`:

```
<element name="LogoutResponse" type="samlp:StatusResponseType"/>
```

8.2.7.3 Reglas de procesamiento

El emisor del mensaje puede emplear el atributo `Reason` para indicar el motivo del envío de la petición `<LogoutRequest>`. En esta Recomendación se definen los siguientes valores que podrán ser utilizados por todos los emisores de mensajes; entre los participantes se pueden poner de acuerdo acerca de otros valores:

`urn:oasis:names:tc:SAML:2.0:logout:user`

Especifica que el mensaje se envía porque el principal desea dar por terminada la sesión indicada.

`urn:oasis:names:tc:SAML:2.0:logout:admin`

Especifica que el mensaje se envía porque un administrador desea dar por terminada la sesión indicada para ese principal.

Se deben respetar las demás reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

En las siguientes subcláusulas se presentan reglas de procesamiento adicionales.

1) Reglas que se aplican al participante en la sesión

Cuando un participante en la sesión recibe un mensaje `<LogoutRequest>`, tiene que autenticarlo. Si el emisor es la autoridad que proporcionó una aserción que contiene un enunciado de autenticación vinculado a la sesión en curso del principal, el participante en la sesión debe invalidar la sesión o sesiones del principal referidas por el elemento `<saml:BaseID>`, `<saml:NameID>` o `<saml:EncryptedID>`, y todos los elementos `<SessionIndex>` suministrados en el mensaje. Si no se proporcionan elementos `<SessionIndex>`, se tendrán que invalidar todas las sesiones asociadas con el principal.

El participante en la sesión debe aplicar el mensaje de petición de fin de sesión a toda aserción que satisfaga las siguientes condiciones, aun cuando la aserción sea recibida tras la petición de fin de sesión:

- El sujeto de la aserción **concuerda estrictamente** con el elemento `<saml:BaseID>`, `<saml:NameID>` o `<saml:EncryptedID>` en la petición `<LogoutRequest>`, conforme a 8.2.3.4.
- El atributo `SessionIndex` de uno de los enunciados de autenticación de la aserción concuerda con uno de los elementos `<SessionIndex>` especificados en la petición de fin de sesión, o la petición de fin de sesión no contiene elementos `<SessionIndex>`.
- En cualquier otro caso, la aserción será válida basándose en las condiciones temporales especificadas en la propia aserción (en particular, el valor de cualquier atributo `NotOnOrAfter` especificado en las condiciones o en los datos de confirmación del sujeto).

La validez de la petición de fin de sesión aún no ha expirado (se determina examinando el atributo `NotOnOrAfter` en el mensaje).

NOTA – Esta regla tiene por finalidad evitar una situación en la que un participante en la sesión recibe una petición de fin de sesión dirigida a una sola aserción o a múltiples aserciones (identificada por el elemento o elementos `<SessionIndex>`) *antes* de recibir la aserción o aserciones reales (y posiblemente aún válidas) dirigidas por la petición de fin de sesión. En este caso, el participante debería aceptar la petición de fin de sesión hasta que ella misma se autodescarte (el valor de `NotOnOrAfter` en la petición se ha sobrepasado) o hasta que se haya recibido la aserción dirigida por la petición de fin de sesión y haya sido tratada adecuadamente.

2) Reglas de la autoridad de sesión

Cuando una autoridad de sesión recibe un mensaje `<LogoutRequest>`, debe autenticar el emisor correspondiente. Si se trata de un participante en la sesión que recibió una aserción de la autoridad de sesión con un enunciado de autenticación para la sesión en curso, la autoridad de sesión debería llevar a cabo lo siguiente en el orden especificado:

– Enviar un mensaje `<LogoutRequest>` a cualquier autoridad de sesión en cuyo nombre la autoridad de sesión autorizó la autenticación del principal, a menos que la segunda autoridad haya originado la `<LogoutRequest>`.

– Enviar un mensaje `<LogoutRequest>` a cada participante en la sesión que haya recibido aserciones de la autoridad de sesión durante la sesión en curso, *siempre que no se trate* del originador de una `<LogoutRequest>` actual.

– Dar por terminada la sesión actual del principal conforme a la especificación del elemento `<saml:BaseID>`, `<saml:NameID>` o `<saml:EncryptedID>`, y cualquier elemento `<SessionIndex>` incluido en el mensaje de petición de fin de sesión.

Si la autoridad de sesión da por terminada satisfactoriamente la sesión del principal con respecto a ella misma, tiene que responder al peticionario original, si lo hubiera, con un mensaje `<LogoutResponse>` que contenga un valor de código de situación de nivel superior de `urn:oasis:names:tc:SAML:2.0:status:Success`. Si no puede llevarlo a cabo, ha de responder con un mensaje `<LogoutResponse>` que incluya un código de situación de nivel superior que señale el error. De esta manera, la situación de nivel superior indica el estado de la operación de fin de sesión sólo con respecto a la propia autoridad de sesión.

La autoridad de sesión debe tratar de ponerse en contacto con cada participante en la sesión mediante cualquier vinculación de protocolo aplicable/utilizable, aun cuando uno o varios de dichos intentos fracasen o no puedan ser intentados (por ejemplo, debido a que la petición original se realiza usando una vinculación de protocolo que no permite la difusión del fin de sesión a todos los participantes).

En caso de que no todos los participantes en la sesión respondan satisfactoriamente a estos mensajes `<LogoutRequest>` (o si no se puede entrar en contacto con todos los participantes), la autoridad de sesión debe incluir en su mensaje `<LogoutResponse>` un valor de código de situación de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:PartialLogout` a fin de señalar que no todos los demás participantes en la sesión respondieron satisfactoriamente confirmando el fin de sesión.

Una autoridad de sesión puede iniciar un fin de sesión por motivos distintos a la recepción de una <LogoutRequest> proveniente de un participante en la sesión; se incluyen los siguientes, aunque no son los únicos:

- Si se acordó algún periodo temporal fuera de banda con un participante en la sesión individual, la autoridad de sesión puede enviar un mensaje <LogoutRequest> exclusivamente a ese participante individual.
- Se sobrepasó un periodo temporal acordado globalmente.
- El principal o alguna otra entidad de confianza solicitó el fin de la sesión del principal directamente desde la autoridad de sesión.
- La autoridad de sesión ha determinado que los documentos de identidad del principal pueden tener un problema.

Cuando se construye un mensaje de petición de fin de sesión, la autoridad de sesión tiene la obligación de fijar el valor del atributo NotOnOrAfter del mensaje a un valor temporal, indicando una hora de fin de validez del mensaje, tras la cual el destinatario podrá descartar la petición correspondiente. Este valor debería ser fijado a un valor temporal igual o mayor que el de cualquier atributo NotOnOrAfter especificado en la aserción que haya sido expedida más recientemente como parte de la sesión objetivo (indicada por el atributo SessionIndex en la petición de fin de sesión).

Además de los valores especificados en 8.2.6.3 para el atributo Reason, también están disponibles los siguientes valores que sólo pueden ser utilizados por la autoridad de sesión:

urn:oasis:names:tc:SAML:2.0:logout:global-timeout

Especifica que se envía el mensaje debido a que se sobrepasó el periodo del intervalo de tiempo de la sesión global.

urn:oasis:names:tc:SAML:2.0:logout:sp-timeout

Especifica que se envía el mensaje debido a que se sobrepasó el periodo del intervalo de tiempo acordado entre un participante y la autoridad de sesión.

8.2.8 Protocolo de concordancia del identificador de nombre

Cuando una entidad que comparte un identificador de un principal con un proveedor de identidad desea obtener un identificador de nombre para el mismo principal en un formato particular o espacio de nombre de federación, puede enviar una petición al proveedor de identidad mediante este protocolo.

Por ejemplo, un proveedor de servicio que desea establecer comunicación con otro proveedor de servicio con quien no comparte un identificador del principal puede utilizar un proveedor de identidad que comparte un identificador del principal con ambos proveedores de servicio para establecer una correspondencia entre su propio identificador y uno nuevo, generalmente criptado, con la cual podrá establecer comunicación con el segundo proveedor de servicio.

Independientemente del tipo de identificador de que se trate, el identificador concordado debería criptarse en un elemento <saml:EncryptedID> a menos que en una instalación específica se establezca que no se necesita ese tipo de protección.

8.2.8.1 Elemento <NameIDMappingRequest> (Petición de correspondencia del identificador de nombre)

Para que un peticionario solicite un identificador de nombre alternativo para un principal desde un proveedor de identidad, debe enviar un mensaje <NameIDMappingRequest>. Este mensaje tiene el tipo complejo **NameIDMappingRequestType**, que extiende **RequestAbstractType** y añade los siguientes elementos:

– <saml:BaseID> o <saml:NameID> o <saml:EncryptedID> [Obligatorio]

Identificador y datos descriptivos asociados en los que se especifica que el principal está reconocido actualmente por el peticionario y el respondedor. (En 8.1.2 figura más información acerca de este elemento.)

– <NameIDPolicy> (Política de identificador de nombre) [Obligatorio]

Requisitos que han de ser devueltos con relación al formato y el calificador de nombre facultativo del identificador.

El mensaje debería estar firmado o autenticado y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

En el siguiente fragmento de esquema se define el elemento <NameIDMappingRequest> y su tipo complejo **NameIDMappingRequestType**:

```
<element name="NameIDMappingRequest"
type="samlp:NameIDMappingRequestType"/>
<complexType name="NameIDMappingRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <choice>
                              <element ref="saml:BaseID"/>
                              <element ref="saml:NameID"/>
                              <element ref="saml:EncryptedID"/>
                        </choice>
                        <element ref="samlp:NameIDPolicy"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

8.2.8.2 Elemento <NameIDMappingResponse> (Respuesta de concordancia del identificador de nombre)

El destinatario de un mensaje <NameIDMappingRequest> debe responder con un mensaje <NameIDMappingResponse>. Éste tiene el tipo complejo **NameIDMappingResponseType**, que extiende el tipo **StatusResponseType** y añade el siguiente elemento:

– <saml:NameID> o <saml:EncryptedID> [Obligatorio]

Identificador y atributos asociados que especifican el principal en la forma solicitada, por lo general en forma criptada. (En 8.1.2 figura más información acerca de este elemento.)

El mensaje debería estar firmado o autenticado y su integridad protegida por la vinculación de protocolo que se emplee para entregar el mensaje.

En el siguiente fragmento de esquema se define el elemento <NameIDMappingResponse> y su tipo complejo **NameIDMappingResponseType**:

```
<element name="NameIDMappingResponse"
type="samlp:NameIDMappingResponseType"/>
<complexType name="NameIDMappingResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                  <choice>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                  </choice>
            </extension>
      </complexContent>
</complexType>
```

8.2.8.3 Reglas de procesamiento

Si el respondedor no reconoce el principal identificado en la petición, puede responder con un error <Status> que contiene un valor <StatusCode> de segundo nivel de

urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal

A criterio del respondedor, se puede devolver el código de situación urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy para indicar incapacidad o falta de disposición para suministrar un identificador en el formato o espacio de nombre solicitado.

Se deben respetar las demás reglas de procesamiento asociadas con los mensajes de petición y respuesta subyacentes.

8.3 Versiones del SAML

Las versiones del conjunto de Recomendaciones del SAML se asignan de dos maneras independientes. Cada una de ellas se examina en las siguientes subcláusulas, junto con las reglas de procesamiento necesarias para detectar y tratar las diferencias entre las versiones. Asimismo se incluyen directrices acerca de las fechas en las que se prevé modificar la información de la versión específica, así como el motivo, en futuras revisiones del SAML.

Cuando la información de la versión se indica con dos versiones Major (mayor) y Minor (menor), se expresará en el formato *Major.Minor*. La versión número $Major_B.Minor_B$ es superior a la versión número $Major_A.Minor_A$ únicamente si:

$$(Major_B > Major_A) \text{ O } ((Major_B = Major_A) \text{ Y } (Minor_B > Minor_A))$$

8.3.1 Versión del conjunto de especificaciones del SAML

Cada publicación de la Recomendación del SAML incluirá una designación de versión mayor y menor que describe su relación con versiones anteriores y posteriores de la Recomendación. La versión será expresada en el contenido de la Recomendación. La magnitud y el alcance general de las modificaciones a la Recomendación establecerán de manera oficiosa si un conjunto de modificaciones constituye una revisión mayor o menor. En general, si las modificaciones acumulativas son compatibles con una versión anterior, la nueva versión constituirá una revisión menor. En los demás casos, los cambios o modificaciones representarán una revisión mayor.

En la presente Recomendación se define la versión V2.0.

8.3.1.1 Versión del esquema

Como un mecanismo de documentación no normativo, los documentos de esquema XML que se publiquen como parte del conjunto de especificaciones incluirán un atributo de versión en el elemento `<xs:schema>` cuyo valor tiene el formato *Major.Minor*, que refleja la versión del conjunto de especificaciones en el que se han publicado. Las aplicaciones de validación pueden utilizar el atributo como un medio para distinguir qué versión de un esquema se está usando para validar los mensajes o para soportar múltiples versiones del mismo esquema lógico.

8.3.1.2 Versión de la aserción del SAML

El elemento `<Assertion>` del SAML contiene un atributo para expresar la versión mayor o menor de la aserción en una cadena con formato Major.Minor. Cada versión del conjunto de especificaciones del SAML estará orientada a documentar la sintaxis, la semántica y las reglas de procesamiento de las aserciones de la misma versión. Es decir, la versión 1.0 del conjunto de especificaciones describe la versión 1.0, y así en adelante.

NO existe una relación explícita entre la versión de la aserción y el espacio de nombre XML objetivo especificado para las definiciones del esquema de esa versión de aserción.

Se aplican las siguientes reglas de procesamiento:

- Una parte asertante del SAML no debe expedir ninguna aserción con un número de versión Major.Minor general que no pueda ser aceptado por la autoridad.
- Una parte confiante del SAML no debe procesar ninguna aserción con un número de versión mayor que no pueda ser aceptado por la parte confiante.
- Una parte confiante del SAML puede procesar o rechazar una aserción cuyo número de versión menor sea superior al número de versión de aserción minor soportado por la parte confiante. No obstante, todas las aserciones que comparten un número de versión de aserción major deben compartir las mismas reglas y semánticas de procesamiento generales, y pueden ser tratadas de una manera uniforme por una implementación. Por ejemplo, si una aserción V1.1 comparte la sintaxis de una aserción V1.0, una implementación puede tratar la aserción como una aserción V1.0 sin ningún efecto desfavorable.

8.3.1.3 Versión del protocolo SAML

Los diversos elementos de petición y respuesta del protocolo del SAML contienen un atributo para expresar la versión mayor y menor del mensaje de petición o respuesta mediante la utilización de una cadena con el formato *Major.Minor*. Cada versión del conjunto de especificaciones del SAML estará orientada a documentar la sintaxis, la semántica y las reglas de procesamiento de los mensajes de protocolo de la misma versión. Es decir, la versión 1.0 del conjunto de especificaciones describe la versión V1.0 de la petición y la respuesta, y así en adelante.

NO existe una relación explícita entre la versión del protocolo y el espacio de nombre XML objetivo especificado para las definiciones del esquema de esa versión de protocolo.

Los números de versión empleados en los elementos de petición y respuesta del protocolo del SAML corresponderán entre sí para cualquier revisión particular del conjunto de especificaciones del SAML.

1) Versión de la petición

A las peticiones se les aplican las siguientes reglas de procesamiento:

- Un peticionario del SAML debería emitir peticiones con la versión más alta que sea aceptable para el peticionario y el respondedor del SAML.

- Si el peticionario del SAML no conoce las capacidades del respondedor del SAML, debería suponer que el respondedor acepta peticiones con la versión de petición más alta que puede soportar el mismo.
- Un peticionario del SAML no debe emitir un mensaje de petición con un número de versión de petición *Major.Minor* general que concuerde con un número de versión de respuesta que no sea aceptable para el peticionario.
- Un respondedor del SAML debe rechazar cualquier petición que tenga un número de versión mayor que no sea aceptable para el respondedor.

Un respondedor del SAML puede procesar o rechazar cualquier petición cuyo número de versión menor sea superior a la versión de petición más alta que puede soportar. No obstante, todas las peticiones que comparten un número de versión de petición major deben compartir las mismas reglas y semánticas de procesamiento generales, y pueden ser tratadas de una manera uniforme por una implementación. Es decir, si una petición V1.1 comparte la sintaxis de una petición V1.0, un respondedor puede tratar el mensaje de petición como una petición V1.0 sin ningún efecto desfavorable.

2) Versión de la respuesta

A las respuestas se les aplican las siguientes reglas de procesamiento:

- Un respondedor del SAML no debe enviar un mensaje de respuesta con un número de versión más alto que el número de versión de petición del mensaje de petición correspondiente.
- Un respondedor del SAML no debe enviar un mensaje de respuesta con un número de versión mayor inferior que el número de versión de petición mayor del mensaje de petición correspondiente salvo para comunicar el error `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`.
- Una respuesta de error debida a versiones incompatibles del protocolo del SAML debe dar por resultado la notificación de un valor `<StatusCode>` de nivel superior de `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch`, y asimismo puede provocar la notificación de uno de los siguientes valores de segundo nivel:
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`;
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow`; o
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated`.

3) Combinaciones de versión válidas

Las aserciones de una versión mayor particular aparecen sólo en mensajes de respuesta de la misma versión mayor, conforme lo permita la importación del espacio de nombre de la aserción del SAML en el esquema de protocolo del SAML. Por ejemplo, una aserción V1.1 puede aparecer en un mensaje de respuesta V1.0, y una aserción V1.0 en un mensaje de respuesta V1.1, si se hace referencia al esquema de aserción adecuado durante la importación del espacio de nombre. Pero una aserción V1.0 no debe aparecer en un mensaje de respuesta V2.0 porque ambos pertenecen a versiones mayores diferentes.

8.3.2 Versión del espacio de nombre del SAML

Los documentos del esquema XML publicados como parte del conjunto de especificaciones contienen uno o varios espacios de nombre objetivos en los que se colocan las definiciones del tipo, elemento y atributo. Los espacios de nombre son distintos entre ellos, y representan, de una manera conveniente, las definiciones estructurales y sintácticas que conforman esa parte de la especificación.

Las referencias de URI de espacio de nombre que se definen en el conjunto de especificaciones incluirán, por lo general, información relativa a la versión con el formato *Major.Minor* en alguna parte del URI. La versión mayor y menor en el URI ha de concordar con la versión mayor y menor del conjunto de especificaciones en el que se introdujo y definió el espacio de nombre originalmente. Normalmente, el procesador XML no aprovecha esta información ya que trata el espacio de nombre de una manera opaca, pero es útil para comunicar la relación entre el conjunto de especificaciones y los espacios de nombre que define. Este patrón también es seguido por los identificadores basados en URI que se definen en el SAML, los mismos que se enumeran en 8.7.

Los implementadores, por regla general, pueden prever que los espacios de nombre y las definiciones de esquema asociadas que han sido elaboradas gracias a una revisión mayor del conjunto de especificaciones, se mantendrán válidas y estables a través de las revisiones menores de la especificación. Se pueden introducir nuevos espacios de nombre, y de ser necesario, sustituir espacios de nombre antiguos, pero se prevé que será en muy pocas ocasiones. En esos casos, debería esperarse que los espacios de nombre antiguos y sus definiciones asociadas seguirán siendo válidos hasta que se lleve a cabo una revisión mayor del conjunto de especificaciones.

Generalmente, mantener la estabilidad del espacio de nombre y añadir o modificar el contenido de un esquema son objetivos opuestos. Si bien es cierto que algunas estrategias de diseño pueden facilitar las modificaciones o cambios, resulta complejo prever cuál será la reacción de las aplicaciones anteriores a un cambio determinado, dificultando alcanzar la compatibilidad con las versiones futuras. Sin embargo, se reserva el derecho a efectuar esos cambios en las revisiones menores, en beneficio de la estabilidad del espacio de nombre. Salvo en circunstancias especiales (por ejemplo, para corregir deficiencias mayores o errores), las aplicaciones deberían prever cambios en el esquema compatibles con versiones futuras durante las revisiones menores, facilitando la validación entre los nuevos mensajes y los esquemas anteriores.

Las aplicaciones deberían prever y estar preparadas para tratar nuevos tipos de extensiones y mensajes de conformidad con las reglas de procesamiento establecidas para esos tipos. Las revisiones menores pueden introducir nuevos tipos que soportan los recursos de extensión como se describe en esta Recomendación. Las implementaciones antiguas deberían rechazar esas extensiones cuando se las encuentran en contextos que establecen semánticas obligatorias. Algunos ejemplos son los tipos nueva petición, enunciado o condición.

8.4 Sintaxis y procesamiento de firmas XML y SAML

Las aserciones del SAML y los mensajes de petición y respuesta del protocolo del SAML pueden incluir una firma, con las siguientes ventajas. Una aserción firmada por la parte asertante soporta: integridad de aserción, autenticación de la parte asertante ante una parte confiante del SAML y, si la firma está basada en el par de claves pública-privada de la autoridad del SAML, también no repudio de origen. Un mensaje de petición o respuesta del protocolo SAML firmado por el originador del mensaje soporta: integridad del mensaje, autenticación del origen del mensaje ante un destino y, si la firma está basada en el par de claves pública-privada del originador, también no repudio de origen.

El SAML no siempre exige una firma digital. Por ejemplo, en algunas circunstancias, las firmas pueden "heredarse" como cuando una aserción sin firma aprovecha la protección de una firma en el mensaje de respuesta del protocolo. Las firmas de este tipo deben aplicarse con precaución cuando el objeto contenido (como una aserción) está previsto con un tiempo de vida no transitorio. El motivo es que se debe conservar todo el contexto para facilitar la validación, exponiendo el contenido XML y añadiendo una tara posiblemente innecesaria. En otro ejemplo, la parte confiante o el peticionario del SAML puede haber obtenido una aserción o un mensaje de protocolo de la parte asertante o del respondedor del SAML directamente (sin intermediarios) a través de un canal seguro, siendo que estos dos últimos se han autenticado ante la parte confiante o el respondedor del SAML por medios distintos de una firma digital.

Para establecer autenticación "directa" y un canal seguro entre dos partes se dispone de muchas técnicas diferentes. La lista incluye mecanismos basados en contraseña, TLS, HMAC, y otros. Además, los requisitos de seguridad aplicables dependen de las aplicaciones de comunicación y de la naturaleza de la aserción o del mensaje transportado. Se recomienda que en todos los demás contextos se empleen firmas digitales para las aserciones y los mensajes de petición y respuesta. Específicamente:

- Una aserción SAML que recibe una parte confiante del SAML de una entidad que no es la parte asertante del SAML debería estar firmada por esta última.
- Un mensaje de protocolo del SAML que llega a un destino proveniente de una entidad que no es el emisor originador debería estar firmado por este último.
- Los perfiles pueden especificar mecanismos de firma alternativos como S/MIME u objetos Java firmados que contienen documentos del SAML. Se aplican salvedades respecto a la conservación del contexto y al interfuncionamiento. Las firmas XML están previstas como el mecanismo primario de firma del SAML, pero en esta Recomendación se trata de garantizar la compatibilidad con perfiles que pueden necesitar otros mecanismos.
- Salvo que un perfil especifique un mecanismo de firma alternativo, las firmas digitales XML deben estar encapsuladas.

8.4.1 Firma de aserciones

Todas las aserciones del SAML podrán ser firmadas con la firma XML. Esto se refleja en el esquema de aserción que se describe en la cláusula 8.

8.4.2 Firma de petición/respuesta

Todos los mensajes de petición y respuesta del protocolo SAML podrán ser firmados con la firma XML. Esto se refleja en el esquema que se describe en el anexo A.

8.4.3 Herencia de firma

Una aserción SAML puede ser incorporada en otro elemento SAML, tal como una `<Assertion>` circundante, una petición o una respuesta, que puede estar firmada. Cuando una aserción SAML no contiene un elemento

<ds:Signature>, pero está contenida en un elemento SAML circundante que sí lo incluye, y la firma se aplica al elemento <Assertion> y a todos sus vástagos, puede considerarse que la aserción hereda la firma del elemento circundante. La interpretación resultante debería ser equivalente al caso en el que la propia aserción estuviera firmada con las mismas opciones de clave y firma.

Muchos casos de utilización del SAML incluyen datos XML del SAML incorporados en otras estructuras de datos protegidas como en los mensajes SOAP firmados, los lotes S/MIME y las conexiones TLS autenticadas. Los perfiles del SAML pueden definir reglas adicionales para interpretar elementos del SAML como firmas heredadas u otra información de autenticación a partir del contexto circundante, pero la herencia no debe ser inferida a menos que el perfil la identifique específicamente.

8.4.4 Perfil de la firma XML

En el documento W3X XML Signature:2002 se propone una sintaxis XML general para la firma de datos con flexibilidad y muchas alternativas. En esta cláusula se detallan las restricciones relativas a estas características de modo que los procesadores del SAML no tengan que tratar con toda la generalidad del procesamiento de firmas XML. Para ello se utilizan específicamente los atributos **xs:ID**-typed incluidos en los elementos raíz a los que se pueden aplicar las firmas, y en particular el atributo ID relativo a <Assertion> y los diversos elementos de petición y respuesta. En esta cláusula estos atributos se denominan colectivamente atributos de identificador.

Este perfil sólo se aplica al empleo de los elementos <ds:Signature> que se encuentran directamente en las aserciones, peticiones y respuestas del SAML. Los demás perfiles en los que aparecen firmas en cualquier parte pero que se aplican al contenido del SAML pueden definir libremente otros enfoques.

8.4.4.1 Firma de formatos y algoritmos

La firma XML considera tres formas para relacionar una firma con un documento: encapsulante, encapsulada y separada.

Cuando las aserciones y los protocolos del SAML firman aserciones y mensajes del protocolo deben emplear firmas encapsuladas. Los procesadores del SAML deberían poder utilizar las firmas y la verificación RSA para las operaciones de clave pública de conformidad con el algoritmo identificado en 6.4 de http://www.w3.org/2000/09/xmldsig#rsa-sha1.

8.4.4.2 Referencias

Las aserciones y los mensajes del protocolo del SAML deben proporcionar un valor para el atributo ID en el elemento raíz de la aserción o mensaje del protocolo que ha de ser firmado. El elemento raíz de la aserción o del mensaje de protocolo puede ser o no ser el elemento raíz del documento XML que contiene la aserción o el mensaje de protocolo firmado (por ejemplo, podría estar incluido en un sobre SOAP).

Las firmas han de contener una sola <ds:Reference> con una referencia "mismo-documento" al valor del atributo ID del elemento raíz de la aserción o mensaje del protocolo que ha de ser firmado. Por ejemplo, si el valor del atributo ID es "foo", el atributo URI en el elemento <ds:Reference> ha de ser "#foo".

8.4.4.3 Método de canonización

Las implementaciones del SAML deberían emplear canonización exclusiva, con o sin comentarios, tanto en el elemento <ds:CanonicalizationMethod> de <ds:SignedInfo>, como en un algoritmo <ds:Transform>. La utilización de este tipo de canonización asegura que las firmas creadas en mensajes SAML que se incorporan en un contexto XML puedan ser verificadas de modo independiente de ese contexto.

8.4.4.4 Transformadas

Las firmas en los mensajes SAML no deberían contener transformadas que no sean la transformada de firma encapsulada (con el identificador http://www.w3.org/2000/09/xmldsig#enveloped-signature) o las transformadas de canonización exclusiva (con el identificador http://www.w3.org/2001/10/xml-exc-c14n# o http://www.w3.org/2001/10/xml-exc-c14n#WithComments).

Los verificadores de firmas pueden rechazar las que contienen otros algoritmos de transformada, por no ser válidas. Si no lo hacen, los verificadores deben garantizar que la firma no excluye contenido del mensaje del SAML. Esto puede llevarse a cabo estableciendo un acuerdo fuera de banda en el que se indiquen las transformadas que son aceptables o aplicando las transformadas manualmente al contenido y volviendo a verificar que el resultado consta del mismo mensaje del SAML.

8.4.4.5 KeyInfo (información relativa a la clave)

En las reglas de firmas del W3C se define la utilización del elemento <ds:KeyInfo>. El sistema SAML no exige el empleo de dicho elemento, ni impone restricción alguna para su uso. Por lo tanto, <ds:KeyInfo> puede estar ausente.

8.4.4.6 Ejemplo

A continuación se presenta un ejemplo de una respuesta firmada que incluye una aserción firmada. Se añadieron saltos de línea para facilitar la lectura; las firmas no son válidas y no pueden ser verificadas satisfactoriamente.

```
<Response
  IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
  ID="_c7055387-af61-4fce-8b98-e2927324b306"
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
       <saml:Issuer>https://www.opensaml.org/IDP"</saml:Issuer>
       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
              <ds:SignedInfo>
                     <ds:CanonicalizationMethod
                            Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
                     <ds:SignatureMethod
                            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
                     <ds:Reference URI="#_c7055387-af61-4fce-8b98-
e2927324b306">
                            <ds:Transforms>
                                   <ds:Transform

       Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                   <ds:Transform

       Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                          <InclusiveNamespaces
PrefixList="#default saml ds xs xsi"

       xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                   </ds:Transform>
                            </ds:Transforms>
                            <ds:DigestMethod

       Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

       <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
                     </ds:Reference>
              </ds:SignedInfo>
              <ds:SignatureValue>

       x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5

       EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3D
                     w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
              </ds:SignatureValue>
              <ds:KeyInfo>
                     <ds:X509Data>
                            <ds:X509Certificate>

       MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

       MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

       F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

       bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg

       LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx

       CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

       Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

       dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

       CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

       IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+
```

```
      c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

      pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

      hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

      qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

      8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
                        </ds:X509Certificate>
                  </ds:X509Data>
            </ds:KeyInfo>
      </ds:Signature>
      <Status>
            <StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
      </Status>
      <Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
            IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
            xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
            <Issuer>https://www.opensaml.org/IDP</Issuer>
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                  <ds:SignedInfo>
                        <ds:CanonicalizationMethod
                              Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#"/>
                        <ds:SignatureMethod

      Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                        <ds:Reference URI="#_a75adf55-01d7-40cc-929f-
dbd8372ebdfc">
                              <ds:Transforms>
                                    <ds:Transform

      Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                    <ds:Transform

      Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                          <InclusiveNamespaces
                                                PrefixList="#default
saml ds xs xsi"

      xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                    </ds:Transform>
                              </ds:Transforms>
                              <ds:DigestMethod

      Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

      <ds:DigestValue>Kclet6XcaOgOWXM4gty6/UNdviI=</ds:DigestValue>
                        </ds:Reference>
                  </ds:SignedInfo>
                  <ds:SignatureValue>

      hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQIgZpkJN9CMLG8ENR4Nrw+n

      7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtp3TD
                        MwuL/cBUj2OtBZOQMFn7jQ9YB7klIz3RqVL+wNmeWI4=
                  </ds:SignatureValue>
                  <ds:KeyInfo>
                        <ds:X509Data>
                              <ds:X509Certificate>

      MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

      MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

      F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

      bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg
```

```
      LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx

      CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

      Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

      dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

      CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

      IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+

      c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

      pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

      hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

      qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

      8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
                                    </ds:X509Certificate>
                              </ds:X509Data>
                        </ds:KeyInfo>
                  </ds:Signature>
                  <Subject>
                        <NameID
                              Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">
                              scott@example.org
                        </NameID>
                        <SubjectConfirmation
                              Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
                  </Subject>
                  <Conditions NotBefore="2003-04-17T00:46:02Z"
                              NotOnOrAfter="2003-04-17T00:51:02Z">
                        <AudienceRestriction>
                              <Audience>http://www.opensaml.org/SP</Audience>
                        </AudienceRestriction>
                  </Conditions>
                  <AuthnStatement AuthnInstant="2003-04-17T00:46:00Z">
                        <AuthnContext>
                              <AuthnContextClassRef>

      urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                              </AuthnContextClassRef>
                        </AuthnContext>
                  </AuthnStatement>
            </Assertion>
</Response>
```

8.5 Sintaxis y procesamiento de la criptación de XML y SAML

La criptación es útil como un medio para aplicar confidencialidad. Los motivos más comunes para la introducción de confidencialidad son proteger la privacidad de las personas o los secretos de una organización a fin de disponer de una ventaja competitiva o por motivos similares. La confidencialidad también puede ser necesaria para asegurar la eficacia de algún otro mecanismo de seguridad. Por ejemplo, se puede criptar una contraseña o una clave secreta.

A continuación se proponen varias formas de utilizar la protección de la confidencialidad de una aserción SAML o parte de ella.

- La confidencialidad de las comunicaciones puede lograrse mediante mecanismos asociados con una vinculación o perfil particular. Por ejemplo, la vinculación de SOAP soporta el empleo de TLS (véase RFC 2246 del IETF) o los mecanismos de seguridad del mensaje SOAP para efectos de confidencialidad.
- Un secreto `<SubjectConfirmation>` puede ser protegido empleando el elemento `<ds:KeyInfo>` en `<SubjectConfirmationData>`, que permite la criptación de claves u otros secretos.
- Se puede criptar un elemento `<Assertion>` completo conforme a lo descrito en 8.1.3.4.

- Se puede criptar el elemento <BaseID> o <NameID> conforme a los descrito en 8.1.2.4.
- Se puede criptar el elemento <Attribute> conforme a lo descrito en 8.1.7.3.2.

8.5.1 Consideraciones generales

La criptación de los elementos <Assertion>, <BaseID>, <NameID> y <Attribute> se puede realizar aplicando la criptación XML. Los datos criptados y facultativamente una o varias claves criptadas deben reemplazar la información en texto explícito en la misma ubicación en el ejemplar de XML. Se debería utilizar el atributo Type del elemento <EncryptedData>, y si está incluido, ha de tener el valor http://www.w3.org/2001/04/xmlenc#Element.

NOTA (informativa) – En PE30 (véase OASIS PE:2006) se sugiere sustituir una o varias en la segunda línea con cero o más.

Cualquiera de los algoritmos definidos para su empleo con la criptación XML puede ser utilizado para llevar a cabo la criptación. El esquema del SAML se define de tal modo que la inclusión de los datos criptados arroje un ejemplar válido.

8.5.2 Combinación de firmas y criptación

Se puede combinar la utilización de la criptación XML y de las firmas XML. Cuando una aserción tiene que ser firmada y criptada, han de aplicarse las siguientes reglas. Una parte confiante debe ejecutar la validación de la firma y la decriptación en el orden inverso en el que fueron realizadas la firma y la criptación.

- Cuando se cripta un elemento <Assertion> firmado, en primer lugar se debe calcular la firma y luego colocarla en el elemento <Assertion> antes de criptar el elemento.
- Cuando se cripta un elemento <BaseID>, <NameID> o <Attribute>, en primer lugar se debe ejecutar la criptación y a continuación calcular la firma en la aserción o mensaje que contiene el elemento criptado.

8.6 Capacidad de extensión del SAML

El SAML soporta la extensión de diferentes maneras, incluyendo la extensión de los esquemas de la aserción y del protocolo. Véase la cláusula relativa a perfiles en esta Recomendación para obtener información relativa a la forma en la que se definen nuevos perfiles, los cuales pueden ser combinados con las extensiones para preparar el marco del SAML para nuevos modos de utilización.

8.6.1 Extensión del esquema

Los elementos en los esquemas del SAML están bloqueados a fin de que no puedan ser sustituidos, lo que significa que ningún elemento del SAML puede servir como elemento cabecera de un grupo de sustitución. No obstante, los tipos del SAML no están definidos como finales, de manera que todos ellos pueden ser extendidos y restringidos. Desde el punto de vista práctico, esto significa que, por lo general, las extensiones se definen sólo como tipos y no como elementos, y que se incluyen en los ejemplares del SAML mediante un atributo xsi:type.

En las siguientes subcláusulas se examinan únicamente elementos y tipos que han sido concebidos específicamente para soportar extensiones.

8.6.1.1 Extensión de esquema de aserción

El esquema de aserción del SAML está concebido de tal modo que sea posible separar el procesamiento del lote de aserciones de los enunciados incluidos, si se emplea el mecanismo de extensión para cada una de las partes.

Los siguientes elementos están previstos específicamente para que sean utilizados como puntos de extensión en un esquema de extensión; sus tipos están fijados a abstracto, por lo que sólo pueden emplearse como base de un tipo derivado:

- <BaseID> y **BaseIDAbstractType**
- <Condition> y **ConditionAbstractType**
- <Statement> y **StatementAbstractType**

Las siguientes construcciones que pueden aplicarse directamente como parte del SAML son objetivos de extensión particularmente interesantes:

- <AuthnStatement> y **AuthnStatementType**
- <AttributeStatement> y **AttributeStatementType**
- <AuthzDecisionStatement> y **AuthzDecisionStatementType**
- <AudienceRestriction> y **AudienceRestrictionType**

- `<ProxyRestriction>` y **ProxyRestrictionType**
- `<OneTimeUse>` y **OneTimeUseType**

8.6.1.2 Extensión de esquema de protocolo

Los siguientes elementos del protocolo del SAML están previstos específicamente para que sean utilizados como puntos de extensión en un esquema de extensión; sus tipos están fijados a abstracto, por lo que sólo pueden emplearse como base de un tipo derivado:

- `<Request>` y **RequestAbstractType**
- `<SubjectQuery>` y **SubjectQueryAbstractType**

Las siguientes construcciones que pueden aplicarse directamente como parte del SAML son objetivos de extensión particularmente interesantes:

- `<AuthnQuery>` y **AuthnQueryType**
- `<AuthzDecisionQuery>` y **AuthzDecisionQueryType**
- `<AttributeQuery>` y **AttributeQueryType**
- **StatusResponseType**

8.6.2 Puntos de extensión comodín de esquema

Los esquemas del SAML emplean construcciones tipo comodín en algunas ubicaciones para facilitar la utilización de elementos y atributos de espacios de nombre arbitrarios, los cuales son útiles como puntos de extensión incorporados que no requieren un esquema de extensión.

8.6.2.1 Puntos de extensión de aserción

Las siguientes construcciones en el esquema de aserción permiten construcciones de espacios de nombre arbitrarios en ellos:

- `<SubjectConfirmationData>`: Emplea **xs:anyType**, que permite cualesquiera subelementos y atributos.
- `<AuthnContextDecl>`: Emplea **xs:anyType**, que permite cualesquiera subelementos y atributos.
- `<AttributeValue>`: Emplea **xs:anyType**, que permite cualesquiera subelementos y atributos.
- `<Advice>` y **AdviceType**: Además de los elementos nativos del SAML, permite elementos de otros espacios de nombre con el procesamiento de validación del esquema lax.

La siguiente construcción en el esquema de aserción permite atributos globales arbitrarios:

- `<Attribute>` y **AttributeType**

8.6.2.2 Puntos de extensión de protocolo

Las siguientes construcciones en el esquema de protocolo permiten construcciones de espacios de nombre arbitrarios en ellos:

- `<Extensions>` y **ExtensionsType**: Permiten elementos de otros espacios de nombre con procesamiento de validación del esquema lax.
- `<StatusDetail>` y **StatusDetailType**: Permiten elementos de otros espacios de nombre con procesamiento de validación del esquema lax.
- `<ArtifactResponse>` y **ArtifactResponseType**: Permiten elementos de cualquier espacio de nombre con procesamiento de validación del esquema lax. (Sin embargo, están previstos específicamente para el transporte de un elemento de mensaje de petición o respuesta del SAML.)

8.6.3 Extensión de identificador

El SAML emplea identificadores basados en URI para diversos propósitos, tales como los formatos de los códigos de situación y de identificador de nombre, y define algunos identificadores que pueden ser utilizados para dichas finalidades; la mayoría están enumerados en 8.7. No obstante, siempre es posible definir identificadores adicionales basados en URI para esos fines. Se recomienda que esos identificadores adicionales se definan en un perfil de uso formal. En ningún caso debería cambiar apreciablemente el significado de un URI determinado que se emplea como un identificador de ese tipo, o ser utilizado con dos significados diferentes.

8.7 Identificadores definidos en el SAML

En las siguientes subcláusulas se definen identificadores basados en URI para las acciones de acceso a los recursos comunes, para los formatos del identificador de nombre del sujeto y para los formatos del nombre del atributo.

Cuando es posible, se emplea un nombre de recurso uniforme (URN, *uniform resource name*) para especificar un protocolo. En el caso de los protocolos del IETF, se emplea el URN de la norma RFC más reciente que especifique el protocolo. Las referencias de URI que se crean específicamente para el SAML tienen uno de los siguientes enunciados, de conformidad con la versión del conjunto de especificaciones en el que se introdujeron inicialmente:

```
urn:oasis:names:tc:SAML:1.0:
urn:oasis:names:tc:SAML:1.1:
urn:oasis:names:tc:SAML:2.0:
```

En esta Recomendación se introduce el último enunciado.

8.7.1 Identificadores de espacio de nombre de acción

Los siguientes identificadores pueden emplearse en el atributo espacio de nombre del elemento `<Action>` para hacer referencia a conjuntos comunes de acciones que se pueden aplicar a los recursos.

8.7.1.1 Leer/Escribir/Ejecutar/Suprimir/Control

URI: `urn:oasis:names:tc:SAML:1.0:action:rwedc`

Acciones definidas: `Read Write Execute Delete Control` (Leer Escribir Ejecutar Suprimir Control)

Estas acciones se interpretan de la siguiente manera:

`Leer`: El sujeto puede leer el recurso.

`Escribir`: El sujeto puede modificar el recurso.

`Ejecutar`: El sujeto puede ejecutar el recurso.

`Suprimir`: El sujeto puede suprimir el recurso.

`Control`: El sujeto puede especificar la política de control de acceso al recurso.

8.7.1.2 Leer/Escribir/Ejecutar/Suprimir/Control con negación

URI: `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation`

Acciones definidas: `Read Write Execute Delete Control ~Read ~Write ~Execute ~Delete ~Control`

Las acciones especificadas en 8.7.1.1 se deben interpretar de la misma manera como se describen en esta subcláusula. Las acciones que tienen como prefijo una tilde (~) representan permisos negados y se emplean para especificar afirmativamente que se niega el permiso estipulado. Por consiguiente, un sujeto que se describe como autorizado para ejecutar la acción ~Read tiene negado afirmativamente el permiso para leer.

Una autoridad SAML no debe autorizar tanto la acción como su forma negada.

8.7.1.3 Get/Head/Put/Post

URI: `urn:oasis:names:tc:SAML:1.0:action:ghpp`

Acciones definidas: `GET HEAD PUT POST`

Estas acciones vinculan con las operaciones HTTP correspondientes. Por ejemplo, un sujeto autorizado a ejecutar la acción `GET` en un recurso, está autorizado a recuperarlo.

Las acciones `GET` y `HEAD` corresponden en términos generales al permiso read convencional y las acciones `PUT` y `POST` al permiso write. Sin embargo, la correspondencia no es exacta debido a que una operación GET HTTP puede provocar la modificación de datos y una operación POST puede causar la modificación de un recurso distinto al especificado en la petición. Por este motivo se proporciona un especificador de referencia de URI de Action.

8.7.1.4 Permisos de fichero UNIX

URI: `urn:oasis:names:tc:SAML:1.0:action:unix`

Las acciones definidas constituyen el conjunto de permisos de acceso al fichero UNIX expresado en la notación numérica (octal).

La cadena de la acción es un código numérico de cuatro dígitos:

extended user group world

Donde el permiso de acceso *extended* tiene el valor

+2 si sgid está fijado

+4 si suid está fijado

Los permisos de acceso *user group* y *world* tienen el valor

+1 si se concede el permiso de ejecución

+2 si se concede el permiso de escritura

+4 si se concede el permiso de lectura

Por ejemplo, 0754 indica el permiso de acceso al fichero UNIX: user read, write y execute; group read y execute; y world read.

8.7.2 Identificadores del formato del nombre del atributo

Se pueden utilizar los siguientes identificadores en el atributo `NameFormat` que se define en el tipo complejo **AttributeType** para hacer referencia a la clasificación del nombre del atributo para fines de interpretación del nombre.

8.7.2.1 No especificado

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`

La interpretación del nombre del atributo queda a cargo de las aplicaciones individuales.

8.7.2.2 Referencia de URI

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

El nombre del atributo sigue el convenio establecido para las referencias de URI, por ejemplo, como se utiliza en los identificadores de atributo XACML. La interpretación del contenido del URI o el esquema de denominación es específico de la aplicación. En la cláusula 11 figuran perfiles de atributos que utilizan este identificador.

8.7.2.3 Básico

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`

La clase de cadenas aceptable como nombre de atributo debe obtenerse del conjunto de valores que pertenecen al tipo primitivo **xs:Name**, conforme a W3C XML Datatypes, 3.3.6. En la cláusula 13 figuran perfiles de atributos que utilizan este identificador.

8.7.3 Identificadores del formato del identificador de nombre

Se pueden emplear los siguientes identificadores en el atributo `Format` de los elementos `<NameID>`, `<NameIDPolicy>` o `<Issuer>` (véase 8.1.2) para hacer referencia a formatos comunes del contenido de los elementos y de las reglas de procesamiento asociadas, si las hubiere.

NOTA – Varios identificadores que fueron desaconsejados en SAML V1.1 han sido suprimidos del SAML V2.0.

8.7.3.1 No especificado

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

La interpretación del contenido del elemento queda a cargo de las aplicaciones individuales.

8.7.3.2 Dirección de correo electrónico

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`

Indica que el contenido del elemento tiene el formato de una dirección de correo electrónico, específicamente "addr-spec" como se define en RFC 2822, 3.4.1 del IETF. Una addr-spec tiene el formato local-part@domain. Obsérvese que addr-spec no tiene una frase (como en el caso de un nombre común) antes de ella, no tiene un comentario (texto entre paréntesis) después de ella y no está rodeada por "<" and ">".

8.7.3.3 Nombre de sujeto X.509

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`

Indica que el contenido del elemento tiene el formato especificado para el contenido del elemento `<ds:X509SubjectName>` en el documento W3C Signature. Los implementadores deberían observar que en ese documento se especifican las reglas de codificación de los nombres de sujeto X.509 que difieren de las reglas correspondientes en RFC 2253 del IETF.

8.7.3.4 Nombre calificado en el dominio de Windows

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName`

Indica que el contenido del elemento es un nombre calificado en el dominio de Windows. Un nombre de usuario en este dominio es una cadena con el formato "DomainName\UserName". El nombre del domino y el separador "\" pueden omitirse.

8.7.3.5 Nombre principal de Kerberos

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos`

Indica que el contenido del elemento tiene el formato de un nombre principal de Kerberos con el formato `name [/instance]@REALM`. La sintaxis, el formato y los caracteres válidos para el nombre, el ejemplar y el dominio se describen en RFC 1510 del IETF.

8.7.3.6 Identificador de entidad

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`

Indica que el contenido del elemento es el identificador de una entidad que proporciona servicios basados en el SAML (por ejemplo; autoridad, peticionario o respondedor del SAML) o que es un participante en los perfiles del SAML (por ejemplo, proveedor de servicio que soporta el perfil SSO del explorador). El elemento `<Issuer>` puede emplear un identificador de ese tipo para identificar el expedidor de una petición, respuesta o aserción, o en el elemento `<NameID>` a fin de crear aserciones acerca de entidades del sistema que pueden expedir peticiones, respuestas y aserciones del SAML. Asimismo, puede ser utilizado en otros elementos y atributos cuyo propósito es identificar una entidad del sistema en varios intercambios de protocolo.

La sintaxis de dicho identificador es un URI con un máximo de 1024 caracteres de longitud. Es recomendable que una entidad del sistema emplee un URL que incluya su propio nombre de dominio para autoidentificarse .

Los atributos `NameQualifier`, `SPNameQualifier` y `SPProvidedID` deben omitirse.

8.7.3.7 Identificador persistente

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`

Indica que el contenido del elemento es un identificador opaco persistente de un principal que es específico para un proveedor de identidad y un proveedor de servicio o afiliación de proveedores de servicio. Los identificadores de nombre persistentes generados por proveedores de identidad deben construirse con valores seudoaleatorios que no tengan una correspondencia perceptible con el identificador real del sujeto (por ejemplo, el nombre de usuario). La intención es crear un seudónimo por pares, privado, para impedir que se descubra la identidad o las actividades del sujeto. Los valores del identificador de nombre persistente no deben sobrepasar una longitud de 256 caracteres.

Si está presente el atributo `NameQualifier` del elemento, debe contener el identificador único del proveedor de identidad que generó el identificador (véase 8.7.3.6). Éste podrá omitirse si el valor puede deducirse del contexto del mensaje que contiene el elemento, tal como el emisor de un mensaje de protocolo o una aserción que contiene el identificador en su sujeto. Posteriormente, una entidad del sistema distinta podrá emitir su propio mensaje de protocolo o aserción incorporando el identificador; el atributo `NameQualifier` no cambia en este caso, pero debe seguir identificando la entidad que creó originalmente el identificador (y no debe omitirse en tal caso).

Si el atributo `SPNameQualifier` del elemento está presente, ha de incluir el identificador único del proveedor de servicio o de la afiliación de proveedores para el que se generó el identificador (véase 8.7.3.6). Podrá omitirse si el elemento está contenido en un mensaje previsto sólo para que lo reciba directamente el proveedor de servicio, y el valor sería el identificador único de éste.

El atributo `SPProvidedID` del elemento ha de contener el identificador alternativo del principal que haya sido establecido más recientemente por el proveedor de servicio o por la afiliación, si la hubiere (véase 8.2.6). Si el identificador no ha sido establecido el atributo debe omitirse.

Los identificadores persistentes están planificados como un mecanismo de protección de la privacidad, por lo que no deben compartirse en texto explícito con proveedores que no sean los que establecieron el identificador compartido. Además, no deben aparecer en ficheros de registro histórico o en ubicaciones similares sin controles y protecciones adecuadas. Las aplicaciones carentes de tales requisitos pueden emplear libremente otras clases de identificadores para sus intercambios de SAML, pero no deben sobrecargar este formato con valores persistentes pero no opacos.

Aunque los identificadores persistentes se emplean normalmente para reflejar una relación de vinculación de cuentas entre dos proveedores, un proveedor de servicio no está obligado a reconocer o aplicar la naturaleza de largo plazo del identificador persistente, o a establecer dicha vinculación. Este tipo de relación de "un solo extremo" (one-sided) no es discerniblemente diferente y no afecta ni el comportamiento del proveedor de identidad ni las reglas de procesamiento específicas a los identificadores persistentes en los protocolos definidos en esta Recomendación.

Los atributos `NameQualifier` y `SPNameQualifier` indican el sentido de la creación, pero no su utilización. Si un proveedor de identidad particular crea un identificador persistente, el valor del atributo `NameQualifier` se establece permanentemente en ese momento. Si un proveedor de servicio que recibe dicho identificador adopta el cometido de un proveedor de identidad y expide su propia aserción que incluye ese identificador, el valor del atributo `NameQualifier` no cambia (y por supuesto no se omitirá). Alternativamente, podría tomar la decisión de crear su propio identificador persistente para representar el principal y enlazar los dos valores. Se trata de una decisión de la instalación.

8.7.3.8 Identificador transitorio

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`

Indica que el contenido del elemento es un identificador con semánticas transitorias y que la parte confiante debería tratarlo como un valor opaco y temporal. Los valores de este identificador deben ser generados de conformidad con las reglas correspondientes a los identificadores del SAML (véase 7.4), y no deben tener una longitud que sobrepase 256 caracteres.

Los atributos `NameQualifier` y `SPNameQualifier` pueden ser aprovechados para señalar que el identificador representa un identificador por pares transitorio y temporal. En este caso, podrán omitirse de conformidad con las reglas especificadas en 8.7.3.7.

8.7.4 Identificadores de consentimiento (consent)

Los siguientes identificadores pueden ser empleados en el atributo Consent que se define en los tipos complejos **RequestAbstractType** y **StatusResponseType** para notificar si un principal otorgó su consentimiento para el mensaje, y en que condiciones.

8.7.4.1 No especificado (unspecified)

URI: `urn:oasis:names:tc:SAML:2.0:consent:unspecified`

No se efectúa ninguna reclamación en cuanto al consentimiento del principal.

8.7.4.2 Obtenido (obtained)

URI: `urn:oasis:names:tc:SAML:2.0:consent:obtained`

Indica que el emisor del mensaje obtuvo el consentimiento del principal.

8.7.4.3 Previo (prior)

URI: `urn:oasis:names:tc:SAML:2.0:consent:prior`

Indica que el emisor del mensaje obtuvo el consentimiento del principal en algún punto previo a la acción que inició el mensaje.

8.7.4.4 Implícito (implicit)

URI: `urn:oasis:names:tc:SAML:2.0:consent:current-implicit`

Indica que el emisor del mensaje obtuvo implícitamente el consentimiento del principal durante la acción que inició el mensaje, como parte de una indicación más amplia de consentimiento. El consentimiento implícito, por lo general, está más próximo a la acción en cuanto a tiempo y presentación que el consentimiento previo (*prior*), como parte de una sesión de actividades.

8.7.4.5 Explícito (explicit)

URI: `urn:oasis:names:tc:SAML:2.0:consent:current-explicit`

Indica que el emisor del mensaje obtuvo explícitamente el consentimiento del principal durante la acción que inició el mensaje.

8.7.4.6 No disponible (unavailable)

URI: `urn:oasis:names:tc:SAML:2.0:consent:unavailable`

Indica que el emisor del mensaje no obtuvo el consentimiento.

8.7.4.7 No aplicable (inapplicable)

URI: `urn:oasis:names:tc:SAML:2.0:consent:inapplicable`

Indica que el emisor del mensaje no considera necesario obtener o comunicar el consentimiento.

9 Metadatos del SAML

Los perfiles del SAML requieren acuerdos entre las entidades del sistema en lo que concierne a identificadores, soporte de vinculación y puntos extremo, certificados y claves, etc. En esta cláusula se define un formato de metadatos extensivo para las entidades del sistema SAML, organizado por cometidos que reflejan los perfiles del SAML. Esos cometidos incluyen los correspondientes a: proveedor de identidad SSO, proveedor de servicio SSO, afiliación, autoridad de atributo, peticionario de atributo y punto de decisión de política.

9.1 Metadatos

Los metadatos del SAML se organizan en torno a una colección de cometidos extensiva que representa combinaciones comunes de protocolos y perfiles del SAML soportadas por entidades del sistema. Cada cometido se describe mediante un elemento derivado del tipo básico extensivo `RoleDescriptor`. Estos descriptores se recogen a su vez en el elemento contenedor `<EntityDescriptor>`, es decir, la unidad primaria de los metadatos del SAML. Alternativamente, una entidad podría representar una afiliación de otras entidades, como por ejemplo una afiliación de proveedores de servicio. El descriptor `<AffiliationDescriptor>` es útil para esta finalidad.

Estos descriptores pueden a su vez agregarse en grupos anidados mediante el elemento `<EntitiesDescriptor>`.

Se puede soportar una diversidad de mecanismos de seguridad necesarios para establecer la veracidad de los metadatos, particularmente con capacidad para firmar individualmente la mayoría de los elementos que se definen en esta Recomendación.

Cuando los elementos con una relación progenitor/vástago contienen atributos comunes, como información oculta o relativa al fin de la validez, el elemento progenitor tendrá precedencia.

NOTA – Como regla general, los metadatos del SAML no deben considerarse como un enunciado autorizado acerca de las capacidades u opciones de una entidad del sistema determinada. Es decir, aunque deberían ser precisos, no es necesario que sean exhaustivos. La omisión de una opción particular no implica que no puede ser soportada o que sí lo puede ser, sino simplemente que no se alega dicha condición. Como ejemplo, una autoridad de atributo del SAML podría soportar cualquier número de atributos que no estén nombrados en un `<AttributeAuthorityDescriptor>`. Las omisiones podrían reflejar privacidad o muchas otras consideraciones. Inversamente, la indicación del soporte de un atributo dado no implica que cierto peticionario puede recibirlo o que lo recibirá.

9.1.1 Espacios de nombre *(*namespaces)

Los metadatos del SAML emplean el siguiente espacio de nombre:

```
urn:oasis:names:tc:SAML:2.0:metadata
```

En esta Recomendación se emplea el prefijo de espacio de nombre `md`: para referirse al espacio de nombre anterior.

En el siguiente fragmento de esquema se ilustra la utilización de espacios de nombre en los documentos con metadatos del SAML:

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
20021210/xenc-schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for SAML metadata, first published in SAML 2.0.
        </documentation>
    </annotation>
…
</schema>
```

9.1.2 Tipos comunes

En esta cláusula se definen varios tipos de metadatos que se emplearán para definir elementos y atributos.

9.1.2.1 Tipo único entityIDType (Tipo identificador de entidad)

El tipo único **entityIDType** restringe el tipo de datos del esquema XML **anyURI** a una longitud máxima de 1024 caracteres. El **entityIDType** se usa como un identificador único para las entidades del SAML. Véase también 8.7.3.6. Este tipo de identificador debe ser único entre todas las entidades que interactúan en una instalación determinada. La utilización de un URI y el respeto de la regla que dice que un URI único no debe hacer referencia a diferentes entidades, permite satisfacer este requisito.

En el siguiente fragmento de esquema se define el elemento **entityIDType** simple type:

```
<simpleType name="entityIDType">
      <restriction base="anyURI">
            <maxLength value="1024"/>
      </restriction>
</simpleType>
```

9.1.2.2 Tipo complejo EndpointType (Tipo punto extremo)

El tipo complejo **EndpointType** describe un punto extremo vinculado con un protocolo de SAML en el cual una entidad del SAML puede recibir mensajes de protocolo. Varios elementos de metadatos específicos de protocolo o de perfil están unidos a este tipo. Consta de los siguientes atributos:

– `Binding` (Vinculación) [Obligatorio]

  Atributo obligatorio que especifica la vinculación del SAML soportada por el punto extremo. A cada vinculación se le asigna un URI para identificarla.

– `Location` (Emplazamiento) [Obligatorio]

  Atributo de URI obligatorio que especifica el emplazamiento del punto extremo. La sintaxis autorizada para este tipo de URI depende de la vinculación del protocolo.

– `ResponseLocation` (Emplazamiento de respuesta) [Facultativo]

Especifica, de manera facultativa, un emplazamiento distinto al que deberían enviarse los mensajes de respuesta como parte del protocolo o perfil. La sintaxis autorizada para este tipo de URI depende de la vinculación del protocolo.

El atributo `ResponseLocation` es útil para posibilitar la especificación de diferentes puntos extremo a fin de que puedan recibir mensajes de petición y respuesta asociados con un protocolo o perfil, pero no como un medio de equilibrio de cargas o de redundancia (para esta finalidad pueden incluirse múltiples elementos de este tipo). Cuando un cometido contiene un elemento de este tipo que pertenece a un protocolo o perfil al que sólo se puede aplicar un tipo único de mensaje (de petición o respuesta), no se utiliza el atributo `ResponseLocation`.

NOTA (informativa) – En PE41 (véase OASIS PE:2006) se aclara el párrafo anterior agregando la siguiente oración:

> Si se omite el atributo `ResponseLocation`, se puede suponer que los mensajes de respuesta asociados con un protocolo o perfil son tratados en el URI indicado por el atributo `Location`.

Los elementos de este tipo, en diversos contextos, aparecen en el esquema en secuencias no limitadas. Esto facilita que una entidad ofrezca un protocolo o un perfil en múltiples puntos extremo, por lo general con diferentes vinculaciones de protocolo, permitiendo que el consumidor de los metadatos elija un punto extremo apropiado para sus necesidades. Asimismo, múltiples puntos extremo podrían ofrecer equilibrio de carga o cambio en caso de fallo en el "lado del cliente", particularmente en el caso de una vinculación de protocolo síncrono.

Además, este elemento permite el empleo de elementos y atributos arbitrarios que se definen en un espacio de nombre ajeno al sistema SAML. Cualquier contenido de este tipo debe ser calificado en el espacio de nombre.

En el siguiente fragmento de esquema se define el tipo complejo **EndpointType**:

```
<complexType name="EndpointType">
      <sequence>
            <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Binding" type="anyURI" use="required"/>
      <attribute name="Location" type="anyURI" use="required"/>
      <attribute name="ResponseLocation" type="anyURI" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

9.1.2.3 Tipo complejo IndexedEndpointType (Tipo punto extremo indexado)

El tipo complejo **IndexedEndpointType** extiende el tipo **EndpointType** con un par de atributos para facilitar la indexación de puntos extremo que no son idénticos, de manera que puedan ser referenciados por mensajes de protocolo. Consta de los siguientes atributos adicionales:

– `index` (Índice) [Obligatorio]

Atributo obligatorio que asigna un valor entero único al punto extremo de tal manera que pueda ser referenciado en un mensaje de protocolo. El valor del índice debe ser único sólo dentro de una colección de elementos similares contenidos en el mismo elemento progenitor (es decir, no tiene que ser único en todo el ejemplar).

– `isDefault` (Punto extremo por defecto) [Facultativo]

Atributo booleano opcional que se emplea para designar el punto extremo por defecto entre un conjunto indexado. Si se omite, se supone que el valor es falso.

En tal secuencia de puntos extremos similares basados en este tipo, el punto extremo por defecto es el primer punto extremo con el atributo `isDefault` fijado a `verdadero`. Si no existen puntos extremo de ese tipo, el punto extremo por defecto será el primero sin el atributo `isDefault` fijado a `falso`. Si no existen puntos extremo de ese tipo, el punto extremo por defecto será el primer elemento en la secuencia.

NOTA (informativa) – En PE37 (véase OASIS PE:2006) se sugiere aclarar el párrafo anterior con:

> En tal secuencia de puntos extremo indexados que comparten un nombre de elemento común y un espacio de nombre (es decir, todos los ejemplares de `<md:AssertionConsumerService>` dentro de un cometido), el punto extremo por defecto es el primer punto extremo con el atributo `isDefault` fijado a `verdadero`. Si no existen puntos extremo de ese tipo, el punto extremo por defecto será el primero sin el atributo `isDefault` fijado a `falso`. Si no existen puntos extremo de ese tipo, el punto extremo por defecto será el primer elemento en la secuencia

En el siguiente fragmento de esquema se define el tipo complejo **IndexedEndpointType**:

```
<complexType name="IndexedEndpointType">
      <complexContent>
            <extension base="md:EndpointType">
                  <attribute name="index" type="unsignedShort"
use="required"/>
                  <attribute name="isDefault" type="boolean"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

9.1.2.4 Tipo complejo localizedNameType (Tipo nombre localizado)

El tipo complejo **localizedNameType** extiende un elemento con un valor de cadena con un atributo del lenguaje XML normalizado. En el siguiente fragmento de esquema se define el tipo complejo **localizedNameType**:

```
<complexType name="localizedNameType">
      <simpleContent>
            <extension base="string">
                  <attribute ref="xml:lang" use="required"/>
            </extension>
      </simpleContent>
</complexType>
```

9.1.2.5 Tipo complejo localizedURIType (Tipo URI localizado)

El tipo complejo **localizedURIType** extiende un elemento con valor de URI con un atributo del lenguaje XML normalizado.

En el siguiente fragmento de esquema se define el tipo complejo **localizedURIType**:

```
<complexType name="localizedURIType">
   <simpleContent>
            <extension base="anyURI">
                  <attribute ref="xml:lang" use="required"/>
            </extension>
      </simpleContent>
</complexType>
```

9.1.3 Elementos raíz

Un ejemplar de metadatos del SAML describe una entidad única o múltiples entidades. En el primer caso, el elemento raíz debe ser `<EntityDescriptor>`. En el segundo, el elemento raíz debe ser `<EntitiesDescriptor>`.

9.1.3.1 Elemento <EntitiesDescriptor> (Descriptor de entidades)

El elemento `<EntitiesDescriptor>` contiene los metadatos de un grupo de entidades del SAML nombrado facultativamente. Su tipo complejo **EntitiesDescriptorType** incluye una secuencia de elementos `<EntityDescriptor>`, de elementos `<EntitiesDescriptor>`, o de ambos:

– `ID` (Identificador) [Facultativo]

  Identificador único de documento para el elemento, empleado normalmente con un punto de referencia durante la firma.

– `validUntil` (Válido hasta) [Facultativo]

  Atributo facultativo que indica el instante de expiración de los metadatos contenidos en el elemento y de cualquier elemento contenido.

– `cacheDuration` (Duración del almacenamiento) [Facultativo]

  Atributo facultativo que indica la longitud máxima de tiempo que un consumidor debería guardar los metadatos contenidos en el elemento y en cualquiera de los elementos contenidos.

– `Name` (Nombre) [Facultativo]

  Nombre de cadena que identifica un grupo de entidades del SAML en el contexto de alguna instalación.

– `<ds:Signature>` [Facultativo]

  Firma XML que autentica el elemento contenedor y su contenido, de conformidad con la cláusula 8.

– `<Extensions>` (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas acordadas entre un editor de metadatos y el consumidor. Los elementos de la extensión deben ser calificados por espacio de nombre mediante un espacio de nombre definido por un sistema ajeno al SAML.

– `<EntitiesDescriptor>` (Descriptor de entidades) o `<EntityDescriptor>` (Descriptor de entidad) [Uno o varios]

  Contiene los metadatos de una o varias entidades del SAML, o un grupo anidado de metadatos adicionales.

Cuando se emplea como elemento raíz de un ejemplar de metadatos, este elemento debe contener un atributo `validUntil` o `cacheDuration`. Se recomienda que sólo el elemento raíz de un ejemplar de metadatos contenga cualquiera de esos atributos.

En el siguiente fragmento de esquema se define el elemento `<EntitiesDescriptor>` y su tipo complejo **EntitiesDescriptorType**:

```
<element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
<complexType name="EntitiesDescriptorType">
      <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice minOccurs="1" maxOccurs="unbounded">
                  <element ref="md:EntityDescriptor"/>
                  <element ref="md:EntitiesDescriptor"/>
            </choice>
      </sequence>
      <attribute name="validUntil" type="dateTime" use="optional"/>
      <attribute name="cacheDuration" type="duration" use="optional"/>
      <attribute name="ID" type="ID" use="optional"/>
      <attribute name="Name" type="string" use="optional"/>
</complexType>
<element name="Extensions" type="md:ExtensionsType"/>
<complexType final="#all" name="ExtensionsType">
      <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
      </sequence>
</complexType>
```

9.1.3.2 Elemento <EntityDescriptor> (Descriptor de entidad)

El elemento `<EntityDescriptor>` especifica los metadatos de una entidad del SAML única. Ésta puede actuar en muchos cometidos diferentes para el soporte de múltiples perfiles. Esta Recomendación soporta directamente los siguientes cometidos concretos así como el elemento de extensibilidad abstracto `<RoleDescriptor>`:

- Proveedor de identidad SSO.
- Proveedor de servicio SSO.
- Autoridad de autenticación.
- Autoridad de atributo.
- Punto de decisión de política.
- Afiliación.

Su tipo complejo **EntityDescriptorType** consiste en los siguientes elementos y atributos:

– `entityID` (Identificador de entidad) [Obligatorio]

  Especifica el identificador único de la entidad del SAML cuyos metadatos se describen mediante el contenido del elemento.

– `ID` (Identificador) [Facultativo]

  Identificador único de documento para el elemento, que se emplea normalmente con un punto de referencia durante la firma.

– `validUntil` (Válido hasta) [Facultativo]

  Atributo facultativo que indica el instante de expiración de los metadatos contenidos en el elemento y de cualquier elemento contenido.

– `cacheDuration` (Duración del almacenamiento) [Facultativo]

  Atributo facultativo que indica la longitud máxima de tiempo que un consumidor debería guardar los metadatos contenidos en el elemento y en cualquiera de los elementos contenidos.

– `<ds:Signature>` [Facultativo]

  Firma XML que autentica el elemento contenedor y su contenido.

– `<Extensions>` (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas acordadas entre un editor de metadatos y el consumidor. Los elementos de la extensión deben ser calificados por espacio de nombre mediante un espacio de nombre definido por un sistema ajeno al SAML.

– `<RoleDescriptor>`, `<IDPSSODescriptor>`, `<SPSSODescriptor>`, `<AuthnAuthorityDescriptor>`, `<AttributeAuthorityDescriptor>`, `<PDPDescriptor>` [Uno o varios]; o

– `<AffiliationDescriptor>` [Obligatorio]

  El contenido primario del elemento es una secuencia de uno o varios elementos del descriptor de cometido o un descriptor especializado que define una afiliación.

– `<Organization>` (Organización) [Facultativo]

  Elemento facultativo que identifica la organización responsable de la entidad del SAML descrita por el elemento.

– `<ContactPerson>` (Persona encargada) [Cero o más]

  Secuencia de elementos facultativa que identifica varias clases de personal encargado con el que se puede entrar en contacto.

– `<AdditionalMetadataLocation>` (Localización de los metadatos adicionales) [Cero o más]

  Secuencia facultativa de emplazamientos calificados por espacio de nombre donde existen metadatos adicionales para la entidad SAML. Puede incluir metadatos en formatos alternativos o que describen la observación estricta de otras Recomendaciones ajenas al SAML.

Asimismo, es posible incluir atributos calificados por espacios de nombre arbitrarios a partir de espacios de nombre definidos en un sistema ajeno al SAML.

Cuando se emplea como elemento raíz de un ejemplar de metadatos, este elemento debe contener un atributo `validUntil` o `cacheDuration`. Se recomienda que sólo el elemento raíz de un ejemplar de metadatos contenga cualquiera de esos atributos.

Si aparecen múltiples elementos del descriptor de cometido del mismo tipo, se recomienda que no compartan valores `protocolSupportEnumeration` que se traslapen. En esta Recomendación no está definida la selección entre múltiples elementos del descriptor de cometido del mismo tipo que no comparten un valor `protocolSupportEnumeration`, pero se puede definir mediante perfiles de metadatos, posiblemente con la utilización de otros atributos de extensión distinguidos.

En el siguiente fragmento de esquema se define el elemento `<EntityDescriptor>` y su tipo complejo **EntityDescriptorType**:

```
<element name="EntityDescriptor" type="md:EntityDescriptorType"/>
<complexType name="EntityDescriptorType">
      <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice>
                  <choice maxOccurs="unbounded">
                        <element ref="md:RoleDescriptor"/>
                        <element ref="md:IDPSSODescriptor"/>
                        <element ref="md:SPSSODescriptor"/>
                        <element ref="md:AuthnAuthorityDescriptor"/>
                        <element ref="md:AttributeAuthorityDescriptor"/>
                        <element ref="md:PDPDescriptor"/>
                  </choice>
```

```
                    <element ref="md:AffiliationDescriptor"/>
             </choice>
             <element ref="md:Organization" minOccurs="0"/>
             <element ref="md:ContactPerson" minOccurs="0"
maxOccurs="unbounded"/>
             <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="entityID" type="md:entityIDType" use="required"/>
      <attribute name="validUntil" type="dateTime" use="optional"/>
      <attribute name="cacheDuration" type="duration" use="optional"/>
      <attribute name="ID" type="ID" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

9.1.3.2.1 Elemento <Organization> (Organización)

El elemento `<Organization>` especifica la información básica relativa a una organización responsable de una entidad o cometido del SAML. La utilización de este elemento siempre es facultativa. Su contenido es de naturaleza informativa y no tiene correspondencia directa con ningún elemento o atributo central del SAML. Su tipo complejo **OrganizationType** consta de los siguientes elementos:

– `<Extensions>` (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas según acuerdos entre un editor de metadatos y el consumidor. Las extensiones no deben incluir elementos globales (que no se califican por espacios de nombre) o elementos calificados por un espacio de nombre definido por el SAML en este elemento.

– `<OrganizationName>` (Nombre de la organización) [Uno o varios]

  Uno o más nombres calificados por lenguaje que pueden ser adecuados para que las personas los aprovechen, o pueden no serlo.

– `<OrganizationDisplayName>` (Nombre de visualización de organización) [Uno o varios]

  Uno o más nombres calificados por lenguaje que son adecuados para que los aprovechen las personas.

– `<OrganizationURL>` (URL de organización) [Uno o varios]

  Uno o más URI calificados por lenguaje que especifican un emplazamiento al que habrá que dirigir un usuario para que obtenga información adicional. El calificador por lenguaje se refiere al contenido del material en el emplazamiento especificado.

Asimismo, es posible incluir atributos calificados por espacios de nombre arbitrarios a partir de espacios de nombre definidos en un sistema ajeno al SAML.

En el siguiente fragmento de esquema se define el elemento `<Organization>` y su tipo complejo **OrganizationType**:

```
<element name="Organization" type="md:OrganizationType"/>
<complexType name="OrganizationType">
      <sequence>
             <element ref="md:Extensions" minOccurs="0"/>
             <element ref="md:OrganizationName" maxOccurs="unbounded"/>
             <element ref="md:OrganizationDisplayName"
maxOccurs="unbounded"/>
             <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
      </sequence>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="OrganizationName" type="md:localizedNameType"/>
<element name="OrganizationDisplayName" type="md:localizedNameType"/>
<element name="OrganizationURL" type="md:localizedURIType"/>
```

9.1.3.2.2 Elemento <ContactPerson> (Persona encargada con la que se debe entrar en contacto)

El elemento `<ContactPerson>` especifica la información básica que permite ponerse en contacto con la persona encargada en cierta medida de una entidad o cometido del SAML. La utilización de este elemento es siempre facultativa. Su contenido es de naturaleza informativa y no tiene correspondencia directa con ningún elemento o atributo central del SAML. Su tipo complejo **ContactType** consta de los siguientes elementos y atributos:

– `contactType` (Tipo de contacto) [Obligatorio]

  Especifica el tipo de contacto usando la enumeración **ContactTypeType**. Los valores posibles son: técnico, soporte, administrativo, facturación y otros.

– `<Extensions>` (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas según acuerdos entre un editor de metadatos y el consumidor. Los elementos de la extensión deben ser calificados por un espacio de nombre que se define en un sistema ajeno al SAML.

– `<Company>` (Empresa) [Facultativo]

  Elemento facultativo de una cadena que especifica el nombre de la empresa donde se puede entrar en contacto con la persona encargada.

– `<GivenName>` (Nombre) [Facultativo]

  Elemento facultativo de una cadena que especifica el nombre de pila (el primero) de la persona encargada.

– `<SurName>` (Apellido) [Facultativo]

  Elemento facultativo de una cadena que especifica el apellido de la persona encargada.

– `<EmailAddress>` (Dirección de correo electrónico) [Cero o varios]

  Cero o más elementos que contienen los URI mailto: que representan las direcciones de correo-e que pertenecen a la persona encargada.

– `<TelephoneNumber>` (Número telefónico) [Cero o varios]

  Cero o más elementos de una cadena que especifican un número telefónico de la persona encargada.

Asimismo, es posible incluir atributos calificados por espacios de nombre arbitrarios a partir de espacios de nombre definidos en un sistema ajeno al SAML.

En el siguiente fragmento de esquema se define el elemento `<ContactPerson>` y su tipo complejo **ContactType**:

```
<element name="ContactPerson" type="md:ContactType"/>
<complexType name="ContactType">
       <sequence>
              <element ref="md:Extensions" minOccurs="0"/>
              <element ref="md:Company" minOccurs="0"/>
              <element ref="md:GivenName" minOccurs="0"/>
              <element ref="md:SurName" minOccurs="0"/>
              <element ref="md:EmailAddress" minOccurs="0"
maxOccurs="unbounded"/>
              <element ref="md:TelephoneNumber" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="contactType" type="md:ContactTypeType"
use="required"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="Company" type="string"/>
<element name="GivenName" type="string"/>
<element name="SurName" type="string"/>
<element name="EmailAddress" type="anyURI"/>
<element name="TelephoneNumber" type="string"/>
<simpleType name="ContactTypeType">
       <restriction base="string">
              <enumeration value="technical"/>
              <enumeration value="support"/>
              <enumeration value="administrative"/>
              <enumeration value="billing"/>
              <enumeration value="other"/>
       </restriction>
</simpleType>
```

9.1.3.2.3 Elemento <AdditionalMetadataLocation> (Emplazamiento de los metadatos adicionales)

El elemento `<AdditionalMetadataLocation>` constituye un URI calificado por espacio de nombre que especifica donde pueden existir metadatos adicionales basados en XML para una entidad del SAML. Su tipo complejo **AdditionalMetadataLocationType** extiende el tipo **anyURI** con un atributo de espacio de nombre (también de tipo **anyURI**). Este atributo obligatorio debe contener el espacio de nombre XML del elemento raíz del documento del ejemplar encontrado en el emplazamiento especificado.

En el siguiente fragmento de esquema se define el elemento `<AdditionalMetadataLocation>` y su tipo complejo **AdditionalMetadataLocationType**:

```
<element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
<complexType name="AdditionalMetadataLocationType">
      <simpleContent>
            <extension base="anyURI">
                  <attribute name="namespace" type="anyURI"
use="required"/>
            </extension>
      </simpleContent>
</complexType>
```

9.1.4 Elementos del descriptor de cometido

Los elementos en esta cláusula constituyen la mayor parte del componente de soporte operacional de los metadatos. Cada elemento (salvo el abstracto) define una colección específica de comportamientos operacionales para soportar los perfiles del SAML.

9.1.4.1 Elemento <RoleDescriptor> (Descriptor de cometido)

El elemento `<RoleDescriptor>` es un punto de extensión abstracto que contiene información descriptiva común prevista para realizar el procesamiento común de los diferentes cometidos. Es posible definir nuevos cometidos extendiendo su tipo complejo abstracto **RoleDescriptorType**, que contiene los siguientes elementos y atributos:

– `ID` (Identificador) [Facultativo]

  Identificador único en el documento para el elemento, que se emplea normalmente con un punto de referencia durante la firma.

– `validUntil` (Válido hasta) [Facultativo]

  Atributo facultativo que indica la hora de expiración de los metadatos incluidos en el elemento y en cualquier elemento contenido.

– `cacheDuration` (Duración del almacenamiento) [Facultativo]

  Atributo facultativo que indica la máxima longitud de tiempo que un consumidor debería guardar los metadatos contenidos en el elemento y en cualquiera de los elementos contenidos.

– `protocolSupportEnumeration` (Enumeración del soporte del protocolo) [Obligatorio]

  Conjunto de URI delimitado por espacios en blanco que identifican el conjunto de especificaciones del protocolo soportado por el elemento role (cometido). Este conjunto, en el caso de las entidades SAML V2.0, debe incluir el URI de espacio de nombre del protocolo del SAML `urn:oasis:names:tc:SAML:2.0:protocol`. Las Recomendaciones SAML subsiguientes podrían compartir el mismo URI de espacio de nombre, pero deberían proporcionar identificadores "protocol support" (soporte de protocolo) para garantizar la discriminación cuando sea necesaria.

– `errorURL` (URL de error) [Facultativo]

  Atributo de URI facultativo que especifica un emplazamiento a donde se debe dirigir un usuario para la resolución de un problema y para el soporte adicional relacionado con este cometido.

– `<ds:Signature>` [Facultativo]

  Firma XML que autentica el elemento contenedor y su contenido.

– `<Extensions>` (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas según acuerdos entre un editor de metadatos y el consumidor. Los elementos de la extensión deben ser calificados por el espacio de nombre que se define en un sistema ajeno al SAML.

– `<KeyDescriptor>` (Descriptor de clave) [Cero o varios]

  Secuencia de elementos facultativa que proporciona información acerca de las claves criptográficas que emplea la entidad cuando adopta este cometido.

– `<Organization>` (Organización) [Facultativo]

  Elemento facultativo que especifica la organización asociada con este cometido. Idéntico al elemento empleado en el elemento `<EntityDescriptor>`.

– `<ContactPerson>` (Persona encargada) [Cero o varios]

  Secuencia de elementos facultativa que especifica las personas encargadas con las que se puede entrar en contacto y que están asociadas con este cometido. Idéntico al elemento empleado en el elemento `<EntityDescriptor>`.

Asimismo, es posible incluir atributos calificados por espacios de nombre arbitrarios a partir de espacios de nombre definidos en un sistema ajeno al SAML.

En el siguiente fragmento de esquema se define el elemento `<RoleDescriptor>` y su tipo complejo **RoleDescriptorType**:

```
<element name="RoleDescriptor" type="md:RoleDescriptorType"/>
<complexType name="RoleDescriptorType" abstract="true">
      <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:KeyDescriptor" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="ID" type="ID" use="optional"/>
      <attribute name="validUntil" type="dateTime" use="optional"/>
      <attribute name="cacheDuration" type="duration" use="optional"/>
      <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
      <attribute name="errorURL" type="anyURI" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<simpleType name="anyURIListType">
      <list itemType="anyURI"/>
</simpleType>
```

9.1.4.1.1 Elemento <KeyDescriptor> (Descriptor de clave)

El elemento `<KeyDescriptor>` proporciona información acerca de la clave o claves criptográficas que emplea una entidad para firmar datos o recibir claves criptadas, así como los detalles criptográficos adicionales. Su tipo complejo **KeyDescriptorType** consta de los siguientes elementos y atributos:

– `use` (Uso) [Facultativo]

  Atributo facultativo que especifica la finalidad de la clave que se está describiendo. Los valores se extraen de la enumeración **KeyTypes**, y consiste en la criptación y firma de los valores.

– `<ds:KeyInfo>` [Obligatorio]

  Elemento facultativo que identifica una clave directa o indirectamente. Véanse las reglas de firmas XML del W3C para obtener detalles adicionales de la aplicación de este elemento.

– `<EncryptionMethod>` (Método de criptación) [Cero o varios]

  Elemento facultativo que especifica un algoritmo y las configuraciones específicas de algoritmo soportadas por la entidad. El contenido exacto varía en función del algoritmo soportado. Véanse las reglas de criptación del W3C en cuanto a la definición de este tipo complejo **xenc:EncryptionMethodType** del elemento.

En el siguiente fragmento de esquema se define el elemento <KeyDescriptor> y su tipo complejo **KeyDescriptorType**:

```
<element name="KeyDescriptor" type="md:KeyDescriptorType"/>
<complexType name="KeyDescriptorType">
    <sequence>
        <element ref="ds:KeyInfo"/>
        <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="use" type="md:KeyTypes" use="optional"/>
</complexType>
<simpleType name="KeyTypes">
    <restriction base="string">
        <enumeration value="encryption"/>
        <enumeration value="signing"/>
    </restriction>
</simpleType>
<element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>
```

9.1.4.2 Tipo complejo SSODescriptorType (Tipo descriptor de SSO)

El tipo abstracto **SSODescriptorType** es un tipo básico común para los tipos concretos **SPSSODescriptorType** e **IDPSSODescriptorType**, que se describen en las cláusulas subsiguientes. Extiende el tipo **RoleDescriptorType** con elementos que reflejan perfiles comunes para los proveedores de identidad y los proveedores de servicio que soportan SSO, y contiene los siguientes elementos adicionales:

– <ArtifactResolutionService> (Servicio de resolución de artefacto) [Cero o varios]

  Cero o más elementos del tipo **IndexedEndpointType** que describe los puntos extremo indexados que soportan el perfil de resolución de artefacto que se define en la cláusula 12. El atributo ResponseLocation debe omitirse.

– <SingleLogoutService> (Servicio de fin de sesión único) [Cero o varios]

  Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan los perfiles del fin de sesión único que se definen en la cláusula 12.

– <ManageNameIDService> (Servicio de gestión de identificador de nombre) [Cero o varios]

  Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan los perfiles de gestión del identificador de nombre que se definen en la cláusula 12.

– <NameIDFormat> (Formato del identificador de nombre) [Cero o varios]

  Cero o más elementos del tipo **anyURI** que enumera los formatos del identificador de nombre soportados por esta entidad del sistema que adopta este cometido. En el siguiente fragmento de esquema se define el tipo complejo **SSODescriptorType**:

```
<complexType name="SSODescriptorType" abstract="true">
      <complexContent>
            <extension base="md:RoleDescriptorType">
                  <sequence>
                        <element ref="md:ArtifactResolutionService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:SingleLogoutService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:ManageNameIDService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
<element name="SingleLogoutService" type="md:EndpointType"/>
<element name="ManageNameIDService" type="md:EndpointType"/>
<element name="NameIDFormat" type="anyURI"/>
```

9.1.4.3 Elemento <IDPSSODescriptor> (Descriptor de IDPSSO)

El elemento `<IDPSSODescriptor>` extiende el tipo **SSODescriptorType** con un contenido que refleja perfiles específicos para los proveedores de identidad que soportan SSO. Su tipo complejo **IDPSSODescriptorType** contiene los siguientes elementos y atributos adicionales:

– `WantAuthnRequestsSigned` [Facultativo]

Atributo facultativo que indica un requisito que exige la firma de los mensajes `<samlp:AuthnRequest>` recibidos por este proveedor de identidad. Si se omite, se supone que el valor es `false (falso)`.

– `<SingleSignOnService>` (Servicio de inicio de sesión único) [Uno o varios]

Uno o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan los perfiles del protocolo de petición de autenticación que se define en la cláusula 12. Por definición, todos los proveedores de identidad aceptan al menos uno de esos puntos extremo. El atributo `ResponseLocation` debe omitirse.

– `<NameIDMappingService>` (Servicio de correspondencia del identificador de nombre) [Cero o varios]

Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan el perfil de correspondencia del identificador de nombre que se define en la cláusula 12. El atributo `ResponseLocation` debe omitirse.

– `<AssertionIDRequestService>` (Servicio de petición de identificador de aserción) [Cero o varios]

Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan el perfil del protocolo de petición de aserción o la vinculación del URI especial para las peticiones de aserción que se definen en la cláusula 10.

NOTA 1 (informativa) – En PE33 (véase OASIS PE:2006) se sugiere sustituir el protocolo de petición de aserción con la consulta/petición de aserción.

– `<AttributeProfile>` (Perfil de atributo) [Cero o varios]

Cero o más elementos del tipo **anyURI** que enumera los perfiles de atributo soportados por este proveedor de identidad.

– `<saml:Attribute>` [Cero o varios]

Cero o más elementos que identifican los atributos del SAML soportados por el proveedor de identidad. Es posible incluir facultativamente valores específicos que indican que sólo se pueden soportar ciertos valores autorizados por la definición del atributo. En este contexto, "el soporte" de un atributo significa que el proveedor de identidad dispone de la capacidad para incluirlo durante la entrega de aserciones relacionadas con un inicio de sesión único.

NOTA 2 (informativa) – En PE7 (véase OASIS PE:2006) se sugiere añadir el siguiente texto al final del párrafo anterior:

El atributo `WantAuthnRequestsSigned` está previsto para indicar a los proveedores de servicio si pueden esperar o no que el proveedor de identidad acepte un mensaje `<AuthnRequest>` no firmado. El proveedor de identidad no está obligado a rechazar peticiones no firmadas ni el proveedor de servicio está obligado a firmar sus peticiones, aunque se podría esperar de manera razonable que una petición no firmada sea rechazada. En algunos casos, es posible que un proveedor de servicio no sepa ni siquiera qué proveedor de identidad recibirá finalmente sus peticiones y dará respuesta a las mismas, de manera que en tal caso no se puede definir con exactitud el uso de este atributo. Además, obsérvese que el método específico de firma que podría esperarse depende de la vinculación. La vinculación Redirect (redireccionar) de HTTP en 10.2.4 exige que la firma se aplique al valor codificado en el URL en lugar de colocarla en el mensaje XML, mientras que en otras vinculaciones se permite, por lo general, que la firma esté en el mensaje en la forma convencional.

En el siguiente fragmento de esquema se define el elemento `<IDPSSODescriptor>` y su tipo complejo **IDPSSODescriptorType**:

```
<element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
<complexType name="IDPSSODescriptorType">
      <complexContent>
            <extension base="md:SSODescriptorType">
                  <sequence>
                        <element ref="md:SingleSignOnService"
maxOccurs="unbounded"/>
                        <element ref="md:NameIDMappingService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                        <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
                  <attribute name="WantAuthnRequestsSigned"
type="boolean" use="optional"/>
            </extension>
      </complexContent>
</complexType>
<element name="SingleSignOnService" type="md:EndpointType"/>
<element name="NameIDMappingService" type="md:EndpointType"/>
<element name="AssertionIDRequestService" type="md:EndpointType"/>
<element name="AttributeProfile" type="anyURI"/>
```

9.1.4.4 Elemento <SPSSODescriptor> (Descriptor de SPSSO)

El elemento `<SPSSODescriptor>` extiende el tipo **SSODescriptorType** con un contenido que refleja perfiles específicos para los proveedores de servicio. Su tipo complejo **SPSSODescriptorType** contiene los siguientes elementos y atributos adicionales:

– `AuthnRequestsSigned` (Peticiones de autorización firmadas) [Facultativo]

Atributo facultativo que indica si los mensajes `<samlp:AuthnRequest>` enviados por este proveedor de servicio serán firmados. Si se omite, se supone que el valor es `false (falso)`.

NOTA 1 (informativa) – En PE7 (véase OASIS PE:2006) se sugiere añadir el siguiente texto al final del párrafo anterior:

Un valor false (o la omisión de este atributo) no implica que el proveedor de servicio no firmará nunca sus peticiones o que una petición firmada debe considerarse como un error. No obstante, un proveedor de identidad que recibe un mensaje `<samlp:AuthnRequest>` no firmado de un proveedor de servicio cuyos metadatos contienen este atributo con un valor true tiene que devolver una respuesta de error del SAML y no debe satisfacer la petición.

– `WantAssertionsSigned` [Facultativo]

Atributo facultativo que indica un requisito que exige la firma de los elementos `<saml:Assertion>` recibidos por este proveedor de servicio. Si se omite, se supone que el valor es `false`. Este requisito es adicional a cualquier requisito de firma derivado de la utilización de una combinación perfil/vinculación particular.

NOTA 2 (informativa) – En PE7 (véase OASIS PE:2006) se sugiere añadir el siguiente texto al final del párrafo anterior:

Obsérvese que una firma circundante en la capa de vinculación o protocolo del SAML no es suficiente para satisfacer este requisito, por ejemplo, firmando una `<samlp:Response>` que contiene la aserción o aserciones, o una conexión TLS.

– `<AssertionConsumerService>` (Servicio de consumidor de aserciones) [Uno o varios]

Uno o más elementos que describen puntos extremo indexados que soportan los perfiles del protocolo de petición de autenticación que se define en esta Recomendación. Por definición, todos los proveedores de servicio soportan al menos uno de esos puntos extremo.

– `<AttributeConsumingService>` (Servicio de consumo de atributos) [Cero o varios]

Cero o más elementos que describen una aplicación o servicio proporcionado por el proveedor de servicio que necesita o desea utilizar atributos del SAML.

Como máximo sólo un elemento `<AttributeConsumingService>` puede tener el atributo `isDefault` fijado a `true (verdadero)`. Ninguno de los elementos incluidos tiene derecho a contener un atributo `isDefault` fijado a `verdadero`.

En el siguiente fragmento de esquema se define el elemento `<SPSSODescriptor>` y su tipo complejo **SPSSODescriptorType**:

```
<element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
<complexType name="SPSSODescriptorType">
      <complexContent>
            <extension base="md:SSODescriptorType">
                  <sequence>
                        <element ref="md:AssertionConsumerService"
maxOccurs="unbounded"/>
                        <element ref="md:AttributeConsumingService"
minOccurs="0" maxOccurs="unbounded"/>
                  </sequence>
```

```
                    <attribute name="AuthnRequestsSigned" type="boolean"
use="optional"/>
                    <attribute name="WantAssertionsSigned" type="boolean"
use="optional"/>
             </extension>
      </complexContent>
</complexType>
<element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
```

9.1.4.4.1 Elemento <AttributeConsumingService> (Servicio de consumo de atributos)

El elemento `<AttributeConsumingService>` define un servicio particular ofrecido por el proveedor de servicio en cuanto a los atributos que requiere o desea el servicio. Su tipo complejo **AttributeConsumingServiceType** contiene los siguientes elementos y atributos:

– `index` (Índice) [Obligatorio]

  Atributo obligatorio que asigna un valor entero único al elemento para que pueda ser referenciado en un mensaje de protocolo.

– `IsDefault` (Servicio por defecto) [Facultativo]

  Identifica el servicio por defecto que soporta el proveedor de servicio. Es útil si el servicio específico no está indicado de alguna manera por el contexto de aplicación. Si se omite, se supone que el valor es `false (falso)`.

– `<ServiceName>` (Nombre de servicio) [Uno o varios]

  Uno o más nombres calificados por lenguaje para el servicio.

– `<ServiceDescription>` (Descripción del servicio) [Cero o varios]

  Cero o más cadenas calificadas por lenguaje que describen el servicio.

– `<RequestedAttribute>` (Atributo solicitado) [Uno o varios]

  Uno o más elementos que especifican los atributos necesarios o deseados por este servicio.

En el siguiente fragmento de esquema se define el elemento `<AttributeConsumingService>` y su tipo complejo **AttributeConsumingServiceType** complex type:

```
<element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
<complexType name="AttributeConsumingServiceType">
      <sequence>
             <element ref="md:ServiceName" maxOccurs="unbounded"/>
             <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
             <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="index" type="unsignedShort" use="required"/>
      <attribute name="isDefault" type="boolean" use="optional"/>
</complexType>
<element name="ServiceName" type="md:localizedNameType"/>
<element name="ServiceDescription" type="md:localizedNameType"/>
```

9.1.4.4.2 Elemento <RequestedAttribute> (Atributo solicitado)

El elemento `<RequestedAttribute>` especifica el interés de un proveedor de servicio en un atributo del SAML específico, incluyendo facultativamente valores específicos. Su tipo complejo **RequestedAttributeType** extiende el tipo **saml:AttributeType** con el siguiente atributo:

– `IsRequired` (Requerido) [Facultativo]

  Atributo XML facultativo que indica si el servicio requiere el atributo SAML correspondiente para poder funcionar (en oposición a encontrar simplemente un atributo útil o deseable).

  Si se incluyen elementos `<saml:AttributeValue>` específicos, en ese caso, sólo los valores concordantes son pertinentes para el servicio.

En el siguiente fragmento de esquema se define el elemento <RequestedAttribute> y su tipo complejo **RequestedAttributeType**:

```
<element name="RequestedAttribute" type="md:RequestedAttributeType"/>
<complexType name="RequestedAttributeType">
  <complexContent>
        <extension base="saml:AttributeType">
              <attribute name="isRequired" type="boolean"
use="optional"/>
        </extension>
  </complexContent>
</complexType>
```

9.1.4.5 Elemento <AuthnAuthorityDescriptor> (Descriptor de autoridad de autorización)

El elemento <AuthnAuthorityDescriptor> extiende el tipo **RoleDescriptorType** con un contenido que refleja perfiles específicos para las autoridades de autenticación, es decir, las autoridades del SAML que responden a los mensajes <samlp:AuthnQuery>. Su tipo complejo **AuthnAuthorityDescriptorType** contiene el siguiente elemento adicional:

– <AuthnQueryService> (Servicio de consulta de autorización) [Uno o varios]

  Uno o más elementos del tipo **EndpointType** que describe puntos extremo que soportan el perfil del protocolo de consulta de autenticación que se define en la cláusula 12. Por definición, todas las autoridades de autenticación soportan al menos uno de esos puntos extremo.

– <AssertionIDRequestService> (Servicio de petición de identificador de aserción) [Cero o varios]

  Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan el perfil del protocolo de petición de aserción que se define en la cláusula 12 o la vinculación del URI especial para las peticiones de aserción que se definen en la cláusula 10.

– <NameIDFormat> (Formato del identificador de nombre) [Cero o varios]

  Cero o más elementos del tipo **anyURI** que enumera los formatos del identificador de nombre soportados por esta autoridad (en 8.7.3 figuran algunos valores posibles para este elemento).

En el siguiente fragmento de esquema se define el elemento <AuthnAuthorityDescriptor> y su tipo complejo **AuthnAuthorityDescriptorType**:

```
<element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
<complexType name="AuthnAuthorityDescriptorType">
      <complexContent>
            <extension base="md:RoleDescriptorType">
                  <sequence>
                        <element ref="md:AuthnQueryService"
maxOccurs="unbounded"/>
                        <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="AuthnQueryService" type="md:EndpointType"/>
```

9.1.4.6 Elemento <PDPDescriptor> (Descriptor de PDP)

El elemento <PDPDescriptor> extiende el tipo **RoleDescriptorType** con un contenido que refleja perfiles específicos para los puntos de decisión de política, es decir, las autoridades del SAML que dan respuesta a los mensajes <samlp:AuthzDecisionQuery>. Su tipo complejo **PDPDescriptorType** contiene el siguiente elemento adicional:

– <AuthzService> (Servicio de autorización) [Uno o varios]

  Uno o más elementos del tipo **EndpointType** que describe puntos extremo que soportan el perfil del protocolo de consulta de decisión de autorización que se define en la cláusula 12. Por definición, todos los puntos de decisión de política soportan al menos un punto extremo de ese tipo.

– `<AssertionIDRequestService>` [Cero o varios]

Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan el perfil del protocolo de petición de aserción que se define en la cláusula 12 o la vinculación del URI especial para las peticiones de aserción que se definen en la cláusula 10.

NOTA (informativa) – En PE33 (véase OASIS PE:2006) se sugiere reemplazar protocolo de petición de aserción con consulta/petición de aserción.

– `<NameIDFormat>` (Formato del identificador de nombre) [Cero o varios]

Cero o más elementos del tipo **anyURI** que enumera los formatos del identificador de nombre soportados por esta autoridad (en 8.7.3 figuran algunos valores posibles para este elemento).

En el siguiente fragmento de esquema se define el elemento `<PDPDescriptor>` y su tipo complejo **PDPDescriptorType**:

```
<element name="PDPDescriptor" type="md:PDPDescriptorType"/>
<complexType name="PDPDescriptorType">
      <complexContent>
            <extension base="md:RoleDescriptorType">
                  <sequence>
                        <element ref="md:AuthzService"
maxOccurs="unbounded"/>
                        <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="AuthzService" type="md:EndpointType"/>
```

9.1.4.7 Elemento <AttributeAuthorityDescriptor> (Descriptor de la autoridad de atributo)

El elemento `<AttributeAuthorityDescriptor>` extiende el tipo **RoleDescriptorType** con un contenido que refleja los perfiles específicos para las autoridades de atributo, es decir, las autoridades del SAML que dan respuesta a los mensajes `<samlp:AttributeQuery>`. Su tipo complejo **AttributeAuthorityDescriptorType** contiene los siguientes elementos adicionales:

– `<AttributeService>` (Servicio de atributo) [Uno o varios]

Uno o más elementos del tipo **EndpointType** que describe puntos extremo que soportan el perfil del protocolo de consulta de atributo que se define en la cláusula 12. Por definición, todas las autoridades de atributo soportan al menos un punto extremo de ese tipo.

– `<AssertionIDRequestService>` (Servicio de petición de identificador de aserción) [Cero o varios]

Cero o más elementos del tipo **EndpointType** que describe los puntos extremo que soportan el perfil del protocolo de petición de aserción que se define en la cláusula 12 o la vinculación del URI especial para las peticiones de aserción que se definen en la cláusula 10.

NOTA (informativa) – En PE33 (véase OASIS PE:2006) se sugiere sustituir protocolo de petición de aserción por consulta/petición de aserción.

– `<NameIDFormat>` (Formato del identificador de nombre) [Cero o varios]

Cero o más elementos del tipo **anyURI** que enumera los formatos del identificador de nombre soportados por esta autoridad (en 8.7.3 figuran algunos valores posibles para este elemento).

– `<AttributeProfile>` (Perfil de atributo) [Cero o varios]

Cero o más elementos del tipo **anyURI** que enumera los perfiles de atributo soportados por esta autoridad (en 8.7.3 figuran algunos valores posibles para este elemento).

– `<saml:Attribute>` [Cero o varios]

Cero o más elementos que identifican los atributos del SAML soportados por la autoridad. Es posible incluir facultativamente valores específicos, que indican que sólo se soportan algunos valores autorizados por la definición del atributo.

En el siguiente fragmento de esquema se define el elemento <AttributeAuthorityDescriptor> y su tipo complejo **AttributeAuthorityDescriptorType**:

```
<element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
<complexType name="AttributeAuthorityDescriptorType">
      <complexContent>
            <extension base="md:RoleDescriptorType">
                  <sequence>
                        <element ref="md:AttributeService"
maxOccurs="unbounded"/>
                        <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                        <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                        <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="AttributeService" type="md:EndpointType"/>
```

9.1.5 Elemento <AffiliationDescriptor> (Descriptor de afiliación)

El elemento <AffiliationDescriptor> constituye una alternativa a la secuencia de los descriptores de cometido que se emplean cuando un <EntityDescriptor> describe una afiliación de entidades del SAML (por lo general proveedores de servicio) en lugar de una entidad única. El elemento <AffiliationDescriptor> proporciona un resumen de las entidades individuales que forman la afiliación y la información general acerca de la propia afiliación. Su tipo complejo **AffiliationDescriptorType** contiene los siguientes elementos y atributos:

– `affiliationOwnerID` (Identificador del propietario de la afiliación) [Obligatorio]

  Especifica el identificador único de la entidad que se encarga de la afiliación. No se pretende que el propietario sea miembro de la afiliación, pero si lo es, su identificador ha de aparecer en el elemento <AffiliateMember>.

– `ID` (Identificador) [Facultativo]

  Identificador único de documento para el elemento, que por lo general se emplea como punto de referencia durante la firma.

– `ValidUntil` (Válido hasta) [Facultativo]

  Atributo facultativo que indica la hora de expiración de los metadatos contenidos en el elemento y en cualquiera de los elementos contenidos.

– `cacheDuration` (Duración del almacenamiento) [Facultativo]

  Atributo facultativo que indica la longitud máxima de tiempo que un consumidor debería guardar los metadatos contenidos en el elemento y en cualquiera de los elementos contenidos.

– <ds:Signature> [Facultativo]

  Firma XML que autentica el elemento circundante y su contenido (véase la cláusula 8).

– <Extensions> (Extensiones) [Facultativo]

  Contiene extensiones de metadatos facultativas acordadas entre un editor de metadatos y el consumidor. Los elementos de la extensión deben ser calificados por espacios de nombre definidos por un sistema ajeno al SAML.

– <AffiliateMember> (Miembro afiliado) [Uno o varios]

  Uno o más elementos que enumeran los miembros de la afiliación mediante la especificación de cada identificador único de miembro (véase también 8.7.3.6).

– <KeyDescriptor> (Descriptor de clave) [Cero o varios]

  Secuencia de elementos facultativa que proporciona información acerca de las claves criptográficas que emplea la afiliación de manera global, que difieren de las claves utilizadas por los miembros individuales de la afiliación, y que se publican en los metadatos para esas entidades.

Asimismo, es posible incluir atributos calificados por espacios de nombre arbitrarios a partir de espacios de nombre definidos en un sistema ajeno al SAML.

En el siguiente fragmento de esquema se define el elemento `<AffiliationDescriptor>` y su tipo complejo **AffiliationDescriptorType**:

```
<element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
<complexType name="AffiliationDescriptorType">
      <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
            <element ref="md:KeyDescriptor" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
      <attribute name="validUntil" type="dateTime" use="optional"/>
      <attribute name="cacheDuration" type="duration" use="optional"/>
      <attribute name="ID" type="ID" use="optional"/>
      <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="AffiliateMember" type="md:entityIDType"/>
```

9.1.6 Ejemplos

A continuación se presenta un ejemplo de metadatos para una entidad del sistema SAML que actúa como un proveedor de identidad y como una autoridad de atributo. Se muestra una firma como un guardador de puesto, sin el contenido real.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      entityID="https://IdentityProvider.com/SAML">
      <ds:Signature>...</ds:Signature>
    <IDPSSODescriptor WantAuthnRequestsSigned="true"

      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <KeyDescriptor use="signing">
           <ds:KeyInfo>
               <ds:KeyName>IdentityProvider.com SSO Key</ds:KeyName>
           </ds:KeyInfo>
       </KeyDescriptor>
       <ArtifactResolutionService isDefault="true" index="0"
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://IdentityProvider.com/SAML/Artifact"/>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://IdentityProvider.com/SAML/SLO/SOAP"/>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"

      Location="https://IdentityProvider.com/SAML/SLO/Browser"

      ResponseLocation="https://IdentityProvider.com/SAML/SLO/Response"/>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
       </NameIDFormat>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
       </NameIDFormat>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:transient
       </NameIDFormat>
       <SingleSignOnService
```

```
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"

       Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
        <SingleSignOnService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"

       Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                    FriendlyName="eduPersonPrincipalName">
        </saml:Attribute>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                    FriendlyName="eduPersonAffiliation">
            <saml:AttributeValue>member</saml:AttributeValue>
            <saml:AttributeValue>student</saml:AttributeValue>
            <saml:AttributeValue>faculty</saml:AttributeValue>
            <saml:AttributeValue>employee</saml:AttributeValue>
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </IDPSSODescriptor>
    <AttributeAuthorityDescriptor

       protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:KeyName>IdentityProvider.com AA Key</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <AttributeService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                    Location="https://IdentityProvider.com/SAML/AA/SOAP"/>
        <AssertionIDRequestService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:URI"
                    Location="https://IdentityProvider.com/SAML/AA/URI"/>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
        </NameIDFormat>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
        </NameIDFormat>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </NameIDFormat>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                    FriendlyName="eduPersonPrincipalName">
        </saml:Attribute>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                    FriendlyName="eduPersonAffiliation">
            <saml:AttributeValue>member</saml:AttributeValue>
            <saml:AttributeValue>student</saml:AttributeValue>
            <saml:AttributeValue>faculty</saml:AttributeValue>
            <saml:AttributeValue>employee</saml:AttributeValue>
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </AttributeAuthorityDescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Identity Providers R
US</OrganizationName>
```

```
        <OrganizationDisplayName xml:lang="en">
             Identity Providers R US, a Division of Lerxst Corp.
        </OrganizationDisplayName>
        <OrganizationURL
xml:lang="en">https://IdentityProvider.com</OrganizationURL>
    </Organization>
</EntityDescriptor>
```

A continuación se presenta un ejemplo de metadatos para una entidad del sistema SAML que actúa como un proveedor de servicio. Se muestra una firma como un guardador de lugar, sin el contenido real. Para fines ilustrativos, el servicio es del tipo que no exige que los usuarios se autoidentifiquen de manera inequívoca, sino que por el contrario autoriza el acceso basándose en el atributo de tipo cometido.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      entityID="https://ServiceProvider.com/SAML">
      <ds:Signature>...</ds:Signature>
    <SPSSODescriptor AuthnRequestsSigned="true"

      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <KeyDescriptor use="signing">
           <ds:KeyInfo>
               <ds:KeyName>ServiceProvider.com SSO Key</ds:KeyName>
           </ds:KeyInfo>
       </KeyDescriptor>
       <KeyDescriptor use="encryption">
           <ds:KeyInfo>
               <ds:KeyName>ServiceProvider.com Encrypt Key</ds:KeyName>
           </ds:KeyInfo>
           <EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
       </KeyDescriptor>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://ServiceProvider.com/SAML/SLO/SOAP"/>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"
                   Location="https://ServiceProvider.com/SAML/SLO/Browser"

      ResponseLocation="https://ServiceProvider.com/SAML/SLO/Response"/>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:transient
       </NameIDFormat>
       <AssertionConsumerService isDefault="true" index="0"
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact"

      Location="https://ServiceProvider.com/SAML/SSO/Artifact"/>
       <AssertionConsumerService index="1"
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"
                   Location="https://ServiceProvider.com/SAML/SSO/POST"/>
       <AttributeConsumingService index="0">
           <ServiceName xml:lang="en">Academic Journals R US</ServiceName>
           <RequestedAttribute
                        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
                        FriendlyName="eduPersonEntitlement">
               <saml:AttributeValue>
                   https://ServiceProvider.com/entitlements/123456789
               </saml:AttributeValue>
           </RequestedAttribute>
       </AttributeConsumingService>
    </SPSSODescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Academic Journals R
US</OrganizationName>
        <OrganizationDisplayName xml:lang="en">
```

```
                Academic Journals R US, a Division of Dirk Corp.
        </OrganizationDisplayName>
        <OrganizationURL
xml:lang="en">https://ServiceProvider.com</OrganizationURL>
    </Organization>
</EntityDescriptor>
```

9.2 Procesamiento de firmas

Diversos elementos en un ejemplar de metadatos pueden ser firmados digitalmente (indicado por la inclusión de un elemento `<ds:Signature>`), con las siguientes ventajas:

9.2.1 Integridad de los metadatos

Autenticación de los metadatos mediante un firmador de confianza.

No siempre se exige una firma digital, por ejemplo, si la parte confiante obtiene la información directamente de la entidad editora (sin intermediarios) a través de un canal seguro, y la parte confiante ha autenticado la entidad por algún medio distinto de una firma digital.

Se dispone de muchas técnicas diferentes para la autenticación "directa" y el establecimiento de un canal seguro entre dos partes. La lista incluye TLS, HMAC, los mecanismos basados en contraseña, etc. Asimismo, los requisitos de seguridad que se pueden aplicar dependen de las aplicaciones de comunicación.

Adicionalmente, los elementos pueden heredar firmas de elementos progenitores circundantes que están firmados.

En ausencia de un contexto de este tipo, se recomienda que al menos esté firmado el elemento raíz de un ejemplar de metadatos.

9.2.2 Perfil de la firma XML

La especificación de las firmas XML del W3C hace intervenir una sintaxis XML general para la firma de datos con flexibilidad y múltiples opciones. En esta cláusula se establecen los pormenores de las restricciones relativas a esos recursos, de modo que los procesadores de metadatos no tengan que tratar con toda la generalidad del procesamiento de las firmas XML. Este tipo de utilización aprovecha específicamente los atributos **xs:ID**-typed que están presentes, facultativamente, en los elementos a los que se pueden aplicar las firmas. En esta cláusula, estos atributos se denominan colectivamente atributos de identificador.

1) Formatos y algoritmos de la firma

La firma XML considera tres formas para establecer una relación entre una firma y un documento: encapsulando, encapsulado y separado.

Los metadatos del SAML deben aplicar firmas encapsuladas cuando firmen los elementos que se definen en esta Recomendación. Los procesadores del SAML deberían ser capaces de utilizar las firmas y la verificación del RSA para las operaciones de clave pública de conformidad con el algoritmo identificado por http://www.w3.org/2000/09/xmldsig#rsa-sha1.

2) Referencias

Los elementos de los metadatos firmados deben proporcionar un valor para el atributo del identificador en el elemento firmado. El elemento puede ser el elemento raíz, o puede no serlo, del documento XML real que contiene el elemento de los metadatos firmado.

Las firmas deben incluir una sola `<ds:Reference>` que contiene una referencia de URI al valor de atributo del identificador del elemento de metadatos que va a ser firmado. Por ejemplo, si el valor del atributo del identificador es "foo", el atributo del URI en el elemento `<ds:Reference>` debe ser "#foo".

Por consiguiente, la firma de un elemento de metadatos se debe aplicar al contenido de elemento firmado y a cualquier vástago incluido en el mismo.

3) Método de canonización

Las implementaciones del SAML deberían emplear canonización exclusiva, con o sin comentarios, tanto en el elemento `<ds:CanonicalizationMethod>` de `<ds:SignedInfo>`, como en un algoritmo `<ds:Transform>`. La utilización de este tipo de canonización asegura que las firmas creadas en metadatos del SAML incorporados en un contexto XML puedan ser verificadas de modo independiente a ese contexto.

4) **Transformadas**

Las firmas en los metadatos del SAML no deberían contener transformadas que no sean la transformada de firma encapsulada (con el identificador http://www.w3.org/2000/09/xmldsig#enveloped-signature) o las transformadas de canonización exclusiva (con el identificador http://www.w3.org/2001/10/xml-exc-c14n# o http://www.w3.org/2001/10/xml-exc-c14n#WithComments).

Los verificadores de firmas pueden rechazar, como no válidas, las que contienen otros algoritmos de transformada. Si no lo hacen, los verificadores deben garantizar que no se excluya de la firma contenido del elemento de metadatos firmado. Esto puede llevarse a cabo estableciendo un acuerdo fuera de banda en el que se indiquen las transformadas que son aceptables o aplicando las transformadas manualmente al contenido y volviendo a verificar que el resultado consta de los mismos metadatos del SAML.

5) **KeyInfo (Información relativa a las claves)**

En las reglas de firmas XML del W3C se define la utilización del elemento `<ds:KeyInfo>`. El sistema SAML no exige el empleo de dicho elemento, ni impone restricción alguna para su uso. Por lo tanto, `<ds:KeyInfo>` puede estar ausente.

9.3 Edición y resolución de los metadatos

En esta Recomendación se proponen dos mecanismos para que una entidad pueda publicar (y para que un consumidor determine el emplazamiento de) documentos con metadatos: a través de "un emplazamiento conocido" mediante la anulación directa de la referencia del identificador único de la entidad (un URI al que se hace referencia, de forma muy diversa, como un *entityID* o *providerID*), o indirectamente publicando el emplazamiento de los metadatos en el DNS. Por supuesto que se pueden autorizar otros mecanismos fuera de banda. Si un consumidor soporta ambas alternativas, ha de intentar la resolución a través del DNS antes de emplear el mecanismo basado en "un emplazamiento conocido".

Cuando la recuperación exige el transporte del documento por la red, éste debe ser protegido con mecanismos que proporcionen la autenticación del servidor y la protección de la integridad. Por ejemplo, la resolución basada en HTTP se protegerá con TLS como se define en RFC 2246 del IETF enmendada a través de RFC 3546 del IETF.

En esta cláusula se describen varios mecanismos que ayudan a establecer la confianza en la precisión y legitimación de los metadatos, incluyendo la utilización de las firmas XML, la autenticación del servidor TLS y las firmas DNS. Independientemente del mecanismo o mecanismos seleccionados, las partes confiantes deberían contar con algún medio para establecer la confianza en la información de los metadatos antes de utilizarla.

9.3.1 Publicación y resolución a través de un emplazamiento conocido

En las siguientes subcláusulas se describe la publicación y resolución de los metadatos a través de un emplazamiento conocido.

9.3.1.1 Publicación

Las entidades pueden publicar sus documentos con metadatos en un emplazamiento conocido, colocando el documento en el emplazamiento indicado por su identificador único, el cual debe tener el formato de un URL (en lugar de un URN). Se recomienda firmemente que se utilicen los URL `https` para esta finalidad. Si el documento no se coloca directamente en el emplazamiento puede aplicarse un mecanismo de indirección soportado por el esquema de URL (tal como un HTTP 1.1 302 redirect). Si el protocolo de publicación autoriza la identificación de los tipos de contenido basada en MIME, el tipo de contenido del ejemplar de metadatos debe ser `application/samlmetadata+xml`.

El documento XML proporcionado en el emplazamiento conocido debe describir los metadatos sólo de la entidad representada por el identificador único (es decir, el elemento raíz debe ser un `<EntityDescriptor>` con un `entityID` que concuerde con el emplazamiento). Si es necesario describir otras entidades, se ha de emplear el elemento `<AdditionalMetadataLocation>`. Por consiguiente, no se debe utilizar el elemento `<EntitiesDescriptor>` en documentos publicados que apliquen este mecanismo ya que un identificador de ese tipo no puede definir un grupo de entidades.

9.3.1.2 Resolución (determinación)

Si un identificador único de entidad es un URL, los consumidores de metadatos pueden tratar de resolver directamente un identificador único de entidad, de una manera específica de esquema, anulando la referencia al identificador.

9.3.2 Publicación y resolución a través del DNS

Para tratar de mejorar la accesibilidad a los documentos con metadatos y ofrecer indirección adicional entre un identificador único de entidad y el emplazamiento de los metadatos, las entidades pueden publicar sus emplazamientos de documentos metadatos en una zona de su DNS correspondiente como se define en RFC 1034 del IETF. El identificador único de entidad (un URI) se emplea como la entrada al proceso. Ya que los URI son identificadores

flexibles, los métodos de publicación y el proceso de resolución del emplazamiento se determinan mediante el esquema de URI y el nombre calificado plenamente. Los emplazamientos de URI para los metadatos pueden ser derivados ulteriormente a través de consultas del registro de recursos (RR, *resource record*) NAPTR como se define en RFC 2914 RFC 3403 del IETF.

Se recomienda que las entidades publiquen sus registros de recursos en ficheros de zona firmados aplicando RFC 2535 del IETF de manera que las partes confiantes puedan establecer la validez del emplazamiento publicado y de la autoridad de la zona, así como la integridad de la respuesta del DNS. Si las firmas de zona del DNS están presentes, las partes confiantes deben validarlas adecuadamente.

9.3.2.1 Publicación

Esta Recomendación aprovecha el registro del recurso NAPTR que se describe en RFC 2915 y RFC 3403 del IETF. Se alienta a familiarizarse con estos documentos.

El sistema de descubrimiento de delegación dinámico (DDDS, *dynamic delegation discovery system*) es un sistema de propósito general para la recuperación de información basada en una cadena de entradas específicas de la aplicación y en la utilización de reglas conocidas para transformar esa cadena hasta que se alcanza una situación terminal, lo que requiere una búsqueda en una base de datos definida específica de la aplicación o la resolución de un URL basado en las reglas definidas por la aplicación. El DDDS define un tipo específico de registro de recurso DNS, de registros NAPTR, para el almacenamiento de información en el DNS, que resulta necesario para aplicar las reglas del DDDS.

Las entidades pueden publicar URL separados cuando se necesita distribuir múltiples documentos con metadatos, o cuando se requieren documentos con metadatos diferentes debido a múltiples relaciones de confianza que exigen material de claves independiente, o cuando las interfaces de servicio necesitan declaraciones de metadatos separadas. Esto puede lograrse mediante la aplicación del elemento `<AdditionalMetadataLocation>`, o del recurso regexp y múltiples campos de definición de servicio en el propio registro del recurso NAPTR.

Si el protocolo de publicación autoriza la identificación de los tipos de contenido basada en MIME, el tipo de contenido del ejemplar de metadatos debe ser `application/samlmetadata+xml`.

Si el identificador único de la entidad es un URN, la publicación del emplazamiento de metadatos correspondiente se lleva a cabo conforme a RFC 3404 del IETF. De lo contrario, la resolución de dicho emplazamiento se realizará como se especifica más adelante.

A continuación se presenta el perfil específico de aplicación del sistema DDDS para la resolución de los metadatos del SAML:

1) **Primera regla conocida**

   La "primera regla conocida" para procesar la resolución de los metadatos del SAML es analizar sintácticamente el identificador único de la entidad y extraer el nombre de dominio plenamente calificado (subexpresión 3).

2) **El campo order (orden)**

   Indica el orden en que se ha de procesar cada registro de recurso NAPTR devuelto. Los editores pueden proporcionar múltiples registros de recurso NAPTR que deben ser procesados por la aplicación interpretadora (resolver) en el orden indicado por este campo.

3) **El campo preference (preferencia)**

   Para los registros de recursos NAPTR, el editor expresa el orden de utilización preferido a la aplicación interpretadora, la cual puede ignorar esta orden cuando el valor del campo servicio no satisface los requisitos del interpretador (por ejemplo, el registro del recurso devuelve un protocolo que la aplicación no puede soportar).

4) **El campo flag (bandera)**

   La resolución de los metadatos del SAML emplea dos veces la bandera "U", como terminal y como valor null (*nulo*) (implica la necesidad de procesar registros de recursos adicionales). La bandera "U" indica que la regla arroja un URI.

5) El campo service (servicio)

El campo service específico del SAML, descrito en la siguiente BNF, declara los modos mediante los cuales se podrá disponer del documento o documentos del ejemplar:

```
servicefield = 1("PID2U" / "NID2U") "+" proto [*(":" class) *(":"
servicetype)]
proto = 1("https" / "uddi")
class = 1[ "entity" / "entitygroup" )
servicetype = 1(si / "spsso" / "idpsso" / "authn" / "authnauth" / "pdp" /
"attrauth" / alphanum )
si = "si" [":" alphanum] [":endpoint"]
alphanum = 1*32(ALPHA / DIGIT)
```

donde:

– `servicefield` PID2U interpreta un identificador único de entidad al URL de los metadatos.

– `servicefield` NID2U interpreta un `<NameID>` del principal al URL de los metadatos.

– `proto` describe el protocolo de recuperación (`https` o `uddi`). En el caso de UDDI, el URL será un URL http(s) que hace referencia a un documento WSDL.

– `class` identifica si el documento con metadatos referenciado describe una sola entidad o múltiples entidades. En el segundo caso, el documento referenciado ha de contener la entidad definida por el identificador único original como un miembro de un grupo de entidades dentro del propio documento tal como un `<AffiliationDescriptor>` o `<EntitiesDescriptor>`.

– `servicetype` permite que una entidad pueda publicar los metadatos de distintos cometidos y servicios como documentos separados. Los interpretadores que encuentran múltiples declaraciones `servicetype` anularán la referencia al URI pertinente, dependiendo del servicio necesario para una operación (por ejemplo, una entidad que funciona como un proveedor de identidad y también como un proveedor de servicio puede publicar metadatos de cada cometido en diferentes emplazamientos). El tipo `authn` service representa un punto extremo `<SingleSignOnService>`.

– `si` (con un componente de punto extremo facultativo) permite que el editor publique directamente los metadatos de un ejemplar de servicio, o bien mediante la expresión de un punto extremo SOAP (utilizando `endpoint`).

Por ejemplo:

– `PID2U+https:entity` – representa el documento con metadatos completo de la entidad que está disponible a través del protocolo https.

– `PID2U+uddi:entity:si:foo` – representa el emplazamiento del documento WSDL que describe un ejemplar de servicio "foo".

– `PID2U+https:entitygroup:idpsso` – representa los metadatos de un grupo de entidades que actúan como proveedores de identidad SSO, del cual es miembro la entidad original.

– `NID2U+https:idp` – representa los metadatos del proveedor de identidad SSO de un principal.

6) Los campos regex y replacement (reemplazo)

El resultado previsto tras el procesamiento de la cadena de entrada a través de regex debe ser un URL https o una dirección de nodo UDDI (documento WSDL).

9.3.2.2 Ejemplos de NAPTR

En esta subcláusula se presentan algunos ejemplos de URL y correos electrónicos que pueden ser utilizados por las entidades que soportan NAPTR (véase RFC 2915 del IETF).

a) Ejemplos de NAPTR de metadatos de entidad

Las entidades publican los URL de metadatos de la siguiente manera:

```
$ORIGIN provider.biz

;; order pref f service regexp or replacement

IN NAPTR 100 10 "U" PID2U+https:entity
 "!^.*$!https://host.provider.biz/some/directory/trust.xml!" ""
IN NAPTR 110 10 "U" PID2U+https: entity:trust
 "!^.*!https://foo.provider.biz:1443/mdtrust.xml!" ""
IN NAPTR 125 10 "U" PID2U+https:"
IN NAPTR 110 10 "U" PID2U+uddi:entity
 "!^.*$!https://this.uddi.node.provider.biz/libmd.wsdl" ""
```

b) Ejemplos de identificador de nombre

Un gerente `example.int` del principal se encarga de la operación de un proveedor de identidad que puede ser utilizado por una empresa de suministro de material de oficina para autenticar a los compradores autorizados. El proveedor escoge una dirección de correo-e de usuario `buyer@example.int` como entrada al proceso de interpretación, y examina las direcciones de correo-e para extraer el nombre de dominio totalmente cualificado (FQDN, *fully qualified domaine name*) (`example.int`). El gerente publica los siguientes registros NAPTR en el DNS `example.int`:

```
$ORIGIN example.int

IN NAPTR 100 10 "U" NID2U+https:authn
 "!^([^@]+)@(.*)$!https://serv.example.int:8000/cgi-bin/getmd?\1!" ""
IN NAPTR 100 10 "U" NID2U+https:idp
 "!^([^@]+)@(.*)$!https://auth.example.int/app/auth?\1" ""
```

9.3.2.3 Resolución (interpretación)

Cuando se interpretan los metadatos de una entidad a través del DNS, se emplea el identificador único de esa entidad como la entrada inicial al proceso de interpretación, en lugar de un emplazamiento real y se debe proseguir de la siguiente manera:

- Si el identificador único es un URN, se debe continuar con los pasos de la resolución como se define en RFC 3403 del IETF.
- En cualquier otro caso, se debe analizar el identificador para obtener el nombre de dominio plenamente calificado.
- Consultar el DNS en cuanto a los registros del recurso NAPTR relacionados con el dominio, de manera iterativa, hasta que se devuelva un registro de recurso terminal.
- Identificar qué registro de recurso se debe utilizar basándose en los campos de servicio, después ordenar los campos, y a continuación dar preferencia a los campos del conjunto resultante.
- Obtener el documento o documentos en el emplazamiento o emplazamientos proporcionados según lo exija la aplicación.

En algunos casos, para iniciar la resolución del emplazamiento de la información de los metadatos, será necesario fragmentar el identificador único de la entidad (expresado como un URI) en uno o más elementos atómicos.

La siguiente expresión regular debería aplicarse cuando se inicia el proceso de fragmentación:

```
^([^:/?#]+:)?/*([^:/?#]*@)?(([^/?:#]*\.)*(([^/?#:\.]+)\.([^/?#:\.]+)))(:\d+)?([^?
#]*)(\?[^#]*)?(#.*)?$
     1           2          34           56            7            8      9
10       11
```

La subexpresión 3 debe arrojar un FQDN que constituirá la base para recuperar emplazamientos de metadatos de esta zona.

Cuando se completa el análisis sintáctico del identificador, la aplicación realiza una consulta DNS relativa al dominio resultante (subexpresión 5) de los registros de recursos NAPTR; debería esperar una o más respuestas. Las aplicaciones pueden excluir del conjunto de resultados las definiciones de servicio que no tengan nada que ver con las operaciones de petición en curso.

A continuación, las aplicaciones orientadas a resolución deben ordenar el conjunto de resultados de conformidad con el campo order (orden) y pueden llevarlo a cabo basándose en el conjunto de preferencias. Los interpretadores no están obligados a seguir el orden de las preferencias. El registro o registros de recursos NAPTR resultantes se utilizan iterativamente (basándose en la bandera de orden) hasta alcanzar un registro de recurso NAPTR terminal.

El resultado será un URL absoluto y bien formado, que se aprovecha a continuación para recuperar el documento con metadatos.

9.3.2.4 Almacenamiento de los emplazamientos de metadatos

Este tipo de almacenamiento no debe sobrepasar el TTL de la zona DNS de donde se deduce el emplazamiento. Los interpretadores han de obtener una copia reciente del emplazamiento de los metadatos cuando se alcanza el fin de la validez del TTL de la zona.

Los editores de documentos con metadatos deberían tener en cuenta el TTL de la zona, con precaución, cuando cambian los emplazamientos del documento con metadatos. Si el editor lleva a cabo un cambio de ese tipo, debe conservar el documento en ambos emplazamientos hasta que los interpretadores conformes estén seguros de que ya tienen el emplazamiento actualizado (por ejemplo, el cambio del huso horario + TTL), o envíen una respuesta Redirect HTTP al antiguo emplazamiento especificando el nuevo.

9.3.3 Postprocesamiento de los metadatos

En las siguientes subcláusulas se describe el postprocesamiento de los metadatos.

9.3.3.1 Almacenamiento del ejemplar de los metadatos

El almacenamiento del documento no debe sobrepasar el atributo `validUntil` o `cacheDuration` del elemento o elementos del sujeto. Si los elementos de los metadatos tienen elementos progenitores que incluyen políticas de almacenamiento, el elemento progenitor tendrá la precedencia.

Para que los consumidores puedan procesar adecuadamente el atributo `cacheDuration`, deben conservar la fecha y la hora en las que fue recuperado el documento.

Cuando se termina la validez de un documento o elemento, el consumidor debe recuperar una copia reciente, lo cual puede exigir una restauración del emplazamiento o emplazamientos del documento. Los consumidores deberían procesar el documento almacenado conforme a RFC 2616 13 del IETF, y pueden solicitar la fecha y hora de la última modificación del servidor HTTP. Los editores deberían asegurar un procesamiento aceptable del documento almacenado conforme a RFC 2616, 10.3.5 del IETF (304 Not Modified).

9.3.3.2 Tratamiento de los redireccionamientos (*redirects*) de HTTPS

Los editores pueden emitir una solicitud HTTP Redirect (301 Moved Permanently, 302 ó 307 Temporary Redirect) como se define en RFC 2616, del IETF, y los agentes de usuario deben seguir el URL especificado en la respuesta Redirect. Los Redirects deberían pertenecer al mismo protocolo de la petición inicial.

9.3.3.3 Procesamiento de firmas XML y de confianza general

El procesamiento de los metadatos dispone de varios mecanismos para negociar la confianza de los propios metadatos y la confianza atribuida a la entidad descrita mediante los metadatos:

- confianza derivada de la firma de la zona DNS a partir de la cual se interpretó el URL del emplazamiento de los metadatos, garantizando la precisión del emplazamiento o emplazamientos del documento con los metadatos;
- confianza derivada del procesamiento de la firma del propio documento con los metadatos, garantizando la integridad del documento XML;
- confianza derivada de la autenticación del servidor TLS del URL del emplazamiento de los metadatos, garantizando la identidad del editor de los metadatos.

El posprocesamiento del documento con los metadatos debe incluir el procesamiento de la firma en el nivel del documento XML y puede incluir uno de los otros dos procesos. Específicamente, la parte confiante puede decidir confiar en una cualquiera de las autoridades citadas en la resolución y en el proceso de análisis sintáctico. Los editores de los metadatos deben emplear un mecanismo de integridad del documento y pueden aplicar cualquiera de los otros

dos perfiles de procesamiento a fin de establecer la confianza en el documento con los metadatos, regido por políticas de la implementación. Las siguientes consideraciones habrán de tenerse en cuenta:

1) Procesamiento de zonas DNS firmadas

Si está presente la verificación de la firma de la zona DNS debería ser procesada conforme a RFC 2535 del IETF.

2) Procesamiento de documentos y fragmentos firmados

Los documentos con metadatos publicados deberían estar firmados como se describe en esta Recomendación, mediante un certificado expedido al sujeto del documento o bien a través de otra parte de confianza. Los editores deben tener en cuenta las firmas de otras partes como un medio de transporte de confianza.

Los consumidores de los metadatos han de validar las firmas, cuando las hubiere, en el documento con los metadatos como se describe en esta Recomendación.

3) Procesamiento de la autenticación del servidor durante la recuperación de los metadatos a través de TLS

Se alienta firmemente a los editores a que implementen los URL de TLS; por consiguiente, los consumidores deberían tener en cuenta la confianza heredada del emisor del certificado TLS. Es posible que los URL de publicación no estén ubicados siempre en el dominio del sujeto del documento con los metadatos; por lo tanto, los consumidores no deberían confiar en certificados cuyo sujeto es la entidad en cuestión, ya que pueden estar acogidos por otra parte de confianza.

Como la base de esta confianza no puede estar disponible con referencia a un documento almacenado, en estas circunstancias se deberían aplicar otros mecanismos.

10 Vinculaciones del SAML

En esta cláusula se especifican las vinculaciones del protocolo del SAML necesarias para la utilización de las aserciones del SAML y los mensajes de petición-respuesta en los protocolos y marcos de comunicaciones.

Las traducciones de los intercambios de mensajes de petición-respuesta del SAML en mensajes normales o protocolos de comunicación se denominan *vinculaciones del protocolo* del SAML (o simplemente *vinculaciones*). Un ejemplar de traducción de intercambios de mensajes de petición-respuesta del SAML en un protocolo de comunicación específico `<FOO>` se denomina una *vinculación <FOO> para el SAML* o una *vinculación <FOO> del SAML*.

Por ejemplo, una vinculación SOAP del SAML describe como se traducen los intercambios de mensajes de petición y respuesta del SAML en intercambios de mensajes SOAP.

La finalidad de esta Recomendación es especificar un conjunto de vinculaciones seleccionadas con el detalle suficiente para garantizar que el software conforme con el SAML que se implemente de modo independiente sea capaz de interfuncionar cuando se empleen mensajes normales o protocolos de comunicaciones.

A menos que se indique otra cosa, una vinculación se debería interpretar como concebida para el soporte de la transmisión de cualquier mensaje de protocolo del SAML deducido de los tipos **samlp:RequestAbstractType** y **samlp:StatusResponseType**. Además, cuando una vinculación se refiere a "peticiones y respuestas del SAML", debería entenderse que significa cualesquiera mensajes de protocolo derivados de esos tipos.

En esta Recomendación se utilizan los siguientes convenios tipográficos en el texto: `<ns:Element>`, XMLAttribute, **Datatype**, OtherKeyword. En algunos casos, se emplean paréntesis angulares para señalar no terminales en lugar de elementos XML; la intención se aclara a partir del contexto.

10.1 Directrices para especificar vinculaciones de protocolo adicionales

En esta Recomendación se define un conjunto seleccionado de vinculaciones de protocolo, pero en el futuro se podrán desarrollar otros. En esta cláusula se ofrecen directrices para terceras partes que deseen especificar vinculaciones adicionales. A continuación se presenta una lista de control de los puntos que deben abordarse para cada vinculación de protocolo:

– Especificar tres piezas de información de identificación: un URI que identifica de manera inequívoca la vinculación de protocolo, información de dirección postal o electrónica que permita entrar en contacto con el autor y una referencia a vinculaciones o perfiles que se hayan definido antes y que estén siendo actualizados o suprimidos por la nueva vinculación.

– Describir el conjunto de interacciones entre las partes que intervienen en la vinculación. Se deben nombrar explícitamente las restricciones o aplicaciones utilizadas por cada parte, así como los protocolos que intervienen en cada interacción.
– Identificar la partes que participan en cada interacción, incluyendo cuántas partes participan y si intervienen intermediarios.
– Especificar el método de autenticación de las partes que intervienen en cada interacción, incluyendo si se requiere la autenticación y si los tipos de autenticación son aceptables.
– Identificar el nivel de soporte necesario para la integridad del mensaje, incluyendo los mecanismos que se emplean para garantizarla.
– Identificar el nivel de soporte necesario para la confidencialidad, incluyendo si una tercera parte puede ver el contenido de los mensajes y aserciones del SAML, si la vinculación requiere confidencialidad y los mecanismos recomendados para lograrla.
– Identificar los estados de error, incluyendo los que corresponden a cada participante, especialmente los que reciben y procesan aserciones o mensajes del SAML.
– Identificar las consideraciones de seguridad, incluyendo el análisis de las amenazas y la descripción de las medidas preventivas.
– Identificar las consideraciones relativas a los metadatos, de manera que el soporte de una vinculación en la que interviene un protocolo de comunicaciones particular o que se emplea en un perfil particular pueda ser difundido de una manera eficaz y con capacidad de interfuncionamiento.

10.2 Vinculaciones de protocolo

En las siguientes subcláusulas se definen las vinculaciones de protocolo que se especifican en la norma del SAML.

10.2.1 Consideraciones generales

En las siguientes subcláusulas se describen las características de todas las vinculaciones de protocolo definidas para el SAML.

10.2.1.1 Utilización de RelayState (estado de retransmisión)

Algunas vinculaciones definen un mecanismo denominado "RelayState" que es útil para preservar y transportar información de estado. Cuando se emplea este mecanismo para transportar un mensaje de petición como paso inicial de un protocolo del SAML, se establecen requisitos relativos a la selección y uso de la vinculación que se emplea ulteriormente para transportar la respuesta. Concretamente, si un mensaje de petición del SAML incluye datos RelayState, el respondedor del SAML tiene que devolver su respuesta de protocolo del SAML aplicando una vinculación que también soporta el mismo mecanismo, y debe colocar los mismos datos RelayState que recibió con la petición en el parámetro RelayState correspondiente en la respuesta.

10.2.1.2 Seguridad

A menos que se indique otra cosa, estos enunciados de seguridad se aplican a todas las vinculaciones. Las vinculaciones también pueden emitir enunciados adicionales acerca de estas características de seguridad.

1) Utilización de TLS 1.0

Si no se especifica otra cosa, en cualquier utilización del TLS 1.0 (RFC 2246 del IETF) de la vinculación del SAML, los servidores deben autenticar a los clientes aplicando un certificado X.509 v3. El cliente establecerá la identidad del servidor basándose en el contenido del certificado (por lo general, examinando el campo DN del sujeto del certificado, el atributo `subjectAltName`, etc.).

2) Autenticación del origen de los datos

La autenticación tanto del peticionario del SAML como del respondedor del SAML asociados con un mensaje es facultativa y depende del contexto de utilización. Los mecanismos de autenticación disponibles en la capa de intercambio de mensajes SOAP o a partir del protocolo sustrato subyacente (por ejemplo, en muchas vinculaciones se trata del protocolo TLS o HTTP) pueden ser aprovechados para ofrecer la autenticación del origen de los datos.

En las vinculaciones donde el mensaje de protocolo del SAML pasa a través de un intermediario, la autenticación del transporte no podrá satisfacer los requisitos de autenticación del origen de extremo a extremo. En este caso se recomienda la autenticación del mensaje.

El SAML ofrece por sí mismo mecanismos que permiten que las partes se autentiquen entre ellas, pero adicionalmente puede emplear otros mecanismos de autenticación para proporcionarse seguridad a sí mismo.

3) Integridad del mensaje

La integridad del mensaje tanto de las peticiones como de las respuestas del SAML es facultativa y depende del contexto de uso. Para asegurar la integridad del mensaje puede emplearse la capa de seguridad en el protocolo sustrato subyacente o un mecanismo en la capa de intercambio de mensajes SOAP.

En las vinculaciones donde el mensaje de protocolo del SAML pasa a través de un intermediario, la integridad del transporte no podrá satisfacer los requisitos de integridad de extremo a extremo. En este caso se recomienda la integridad del mensaje.

4) Confidencialidad del mensaje

La confidencialidad del mensaje tanto de las peticiones como de las respuestas del SAML es facultativa y depende del contexto de uso. Para asegurar la confidencialidad del mensaje puede emplearse la capa de seguridad del protocolo del sustrato subyacente o un mecanismo de la capa de intercambio de mensajes SOAP.

En las vinculaciones donde el mensaje de protocolo del SAML pasa a través de un intermediario, la confidencialidad del transporte no podrá satisfacer los requisitos de confidencialidad de extremo a extremo.

5) Otras consideraciones de seguridad

Antes de la instalación debería analizarse la vulnerabilidad de cada combinación de mecanismos de autenticación, integridad del mensaje y confidencialidad, en el contexto del intercambio del protocolo específico y el entorno de despliegue (en el apéndice I figuran los pormenores correspondientes). En RFC 2617 del IETF se describen los posibles ataques al entorno de HTTP cuando se emplean esquemas de autenticación básicos o de compendio de mensajes. Habrá que conceder una precaución especial a la repercusión del posible almacenamiento sobre la seguridad.

10.2.2 Vinculación de SOAP del SAML

El SOAP representa un protocolo ligero previsto para el intercambio de información estructurada en un entorno distribuido y descentralizado. Emplea tecnologías XML para definir un marco de mensajes que se puede extender y proporciona una construcción de mensaje que se puede intercambiar mediante una diversidad de protocolos subyacentes. El marco se concibió de manera que sea independiente de cualquier modelo de programación particular y otras semánticas específicas de la implementación. La simplicidad y la capacidad de extensión son dos de los principales objetivos de diseño de SOAP. Este protocolo trata de cumplir con esos objetivos al omitir, del marco de mensajes, características que a menudo están disponibles en los sistemas distribuidos. Esas características incluyen, aunque no son las únicas, "fiabilidad" (*reliability*), "seguridad" (*security*), "correlación" (*correlation*), "encaminamiento" (*routing*) y "patrones de intercambio de mensajes" (*MEP, message exchange patterns*).

Un mensaje SOAP constituye fundamentalmente una transmisión unidireccional entre nodos SOAP desde un emisor SOAP a un receptor SOAP, encaminada probablemente a través de uno o varios intermediarios SOAP. Se prevé que los mensajes SOAP serán combinados por las aplicaciones para implementar patrones de interacción más complejos que van de petición/respuesta a múltiples intercambios "conversacionales" de ida y vuelta.

El SOAP define un sobre de mensaje XML que incluye las secciones de encabezamiento y cuerpo, que permiten la transmisión de información de datos y de control. Asimismo, define las reglas de procesamiento asociadas con este sobre y una vinculación HTTP para la transmisión de mensajes SOAP.

La vinculación SOAP del SAML define la forma de utilización de SOAP para enviar y recibir peticiones y respuestas del SAML.

El SOAP, de manera similar al SAML puede ser aprovechado por múltiples transportes subyacentes. Esta vinculación posee aspectos que son independientes del protocolo, pero también solicita la utilización de SOAP por HTTP cuando resulta necesario (implementación obligatoria).

10.2.2.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Se proporciona más adelante.

Actualizaciones: `urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding`.

10.2.2.2 Aspectos de la vinculación de SOAP del SAML independientes del protocolo

En las siguientes subcláusulas se definen los aspectos de esta vinculación que son independientes del protocolo subyacente, como el http en el que se transportan los mensajes SOAP. Esta vinculación soporta únicamente el empleo de SOAP 1.1.

10.2.2.2.1 Operación básica

Los mensajes SOAP 1.1 constan de tres elementos: un sobre, datos del encabezamiento y el cuerpo del mensaje. Los elementos del protocolo de petición-respuesta del SAML deben incorporarse en el cuerpo del mensaje SOAP.

El SOAP 1.1 define también un sistema de codificación de datos facultativo, el cual no se utiliza en la vinculación de SOAP del SAML. Esto significa que los mensajes del SAML pueden ser transportados mediante SOAP sin necesidad de volver a codificar del esquema SAML "normal" a otro basado en la codificación SOAP.

El modelo del sistema empleado para las conversaciones del SAML por SOAP es un modelo simple de petición-respuesta.

- Una entidad del sistema que actúa como un peticionario del SAML transmite un elemento de petición del SAML dentro del cuerpo de un mensaje SOAP a una entidad del sistema que actúa como un respondedor del SAML. El peticionario del SAML no debe incluir más de una petición SAML por cada mensaje SOAP ni incluir elementos XML adicionales en el cuerpo de SOAP.
- El respondedor del SAML tiene que devolver un elemento de respuesta del SAML en el cuerpo de otro mensaje SOAP o bien generar un fallo de SOAP. El respondedor del SAML no debe incluir más de una respuesta SAML por cada mensaje SOAP ni incluir elementos XML adicionales en el cuerpo de SOAP. Si un respondedor del SAML no puede, por cualquier motivo, procesar una petición del SAML, habrá de generar un fallo de SOAP. No se deben enviar códigos de fallo de SOAP correspondientes a errores dentro del dominio de un problema del SAML, por ejemplo, incapacidad para encontrar un esquema de extensión o una señal indicando que el sujeto no tiene autorización para acceder a un recurso durante una consulta de autorización.

  NOTA (informativa) – En PE19 (véase OASIS PE:2006) se sugiere sustituir el párrafo anterior por:

  El respondedor del SAML debería devolver un mensaje SOAP que contenga un elemento de respuesta del SAML en el cuerpo o bien un fallo de SOAP. El respondedor del SAML no debe incluir más de una respuesta SAML por cada mensaje SOAP ni incluir elementos XML adicionales en el cuerpo de SOAP. No se deben enviar códigos de fallo de SOAP correspondientes a errores dentro del dominio de un problema del SAML, por ejemplo, incapacidad para encontrar un esquema de extensión o una señal indicando que el sujeto no tiene autorización para acceder a un recurso durante una consulta de autorización.

Cuando se recibe una respuesta SAML en un mensaje SOAP, el peticionario del SAML no debe enviar un código de fallo u otros mensajes de error al respondedor del SAML. Ya que el formato del intercambio de mensajes es un patrón simple de petición-respuesta, si se añaden elementos adicionales como las condiciones de error, éstas podrían complicar el protocolo innecesariamente.

En el documento acerca de SOAP del W3C se hace referencia a un proyecto anterior de la especificación del esquema XML en el que se incluye un espacio de nombre obsoleto. Los peticionarios del SAML deberían generar documentos de SOAP que hagan referencia sólo al espacio de nombre del esquema XML final. Los respondedores del SAML deben ser capaces de procesar ambos espacios de nombre, el del esquema XML utilizado en SOAP 1.1 (véase SOAP W3C) y el del esquema XML final.

10.2.2.2.2 Encabezamiento de SOAP

Un peticionario del SAML en una conversación SAML por SOAP puede añadir encabezamientos arbitrarios al mensaje SOAP. Esta vinculación no define ningún encabezamiento SOAP adicional.

NOTA 1 – El motivo por el que es necesario autorizar otros encabezamientos, es que algunos programas informáticos y bibliotecas de SOAP podrían añadir encabezamientos a un mensaje SOAP que están fuera del control del proceso basado en el SAML. Asimismo, se podrían necesitar algunos encabezamientos para los protocolos subyacentes que requieren encaminamiento de los mensajes o para los mecanismos de seguridad del mensaje.

Un respondedor del SAML no debe solicitar encabezamiento alguno en el mensaje SOAP para procesar correctamente el mensaje del SAML, pero puede exigir encabezamientos adicionales que aborden requisitos de encaminamiento subyacente o de seguridad del mensaje.

NOTA 2 – La razón es que la utilización de encabezamientos suplementarios provocará la fragmentación de la norma del SAML y afectará el interfuncionamiento.

10.2.2.3 Utilización de SOAP por HTTP

Un procesador del SAML que alegue conformidad con la vinculación SOAP del SAML debe implementar SAML por SOAP por HTTP. En esta cláusula se describen algunos datos específicos de utilización de SOAP por HTTP, incluyendo encabezamientos, almacenamiento y notificación de errores de HTTP.

La vinculación de HTTP para SOAP se describe en SOAP W3C, 6.0. En dicho documento se exige la aplicación de un encabezamiento `SOAPAction` como parte de una petición HTTP SOAP. Un respondedor del SAML no debe depender del valor de este encabezamiento. Un peticionario del SAML puede fijar el valor del encabezamiento `SOAPAction` como sigue:

```
http://www.oasis-open.org/committees/security
```

10.2.2.3.1 Encabezamientos de HTTP

Un peticionario del SAML en una conversación SAML por SOAP por HTTP puede añadir encabezamientos arbitrarios a la petición HTTP. Esta vinculación no define ningún encabezamiento HTTP adicional.

NOTA 1 – El motivo por el que es necesario autorizar otros encabezamientos, es que algunos programas informáticos y bibliotecas de HTTP podrían añadir encabezamientos a un mensaje HTTP que están fuera del control del proceso basado en el SAML. Asimismo, se podrían necesitar algunos encabezamientos para los protocolos subyacentes que requieren encaminamiento de los mensajes o para los mecanismos de seguridad del mensaje.

Un respondedor del SAML no debe solicitar encabezamiento alguno en la petición HTTP para procesar correctamente el mensaje del SAML, pero puede exigir encabezamientos adicionales que aborden requisitos de encaminamiento subyacente o de seguridad del mensaje.

NOTA 2 – La razón es que la utilización de encabezamientos suplementarios provocará la fragmentación de la norma del SAML y afectará el interfuncionamiento.

10.2.2.3.2 Almacenamiento

Los mandatarios HTTP no deben almacenar mensajes de protocolo del SAML. Para garantizarlo se deben seguir las reglas siguientes.

Cuando los peticionarios empleen HTTP 1.1, deberían:

- Incluir un campo de encabezamiento `Cache-Control` (de control de almacenamiento) fijado a "`no-cache, no-store`" (sin almacenamiento en memoria intermedia, sin almacenamiento).
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".

Cuando los respondedores empleen HTTP 1.1, deberían:

- Incluir un campo de encabezamiento `Cache-Control` fijado a "`no-cache, no-store, must-revalidate, private`" (sin almacenamiento en memoria intermedia, sin almacenamiento, debe revalidar, privado).
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".
- Impedir la inclusión de un Validator, tal como un encabezamiento `Last-Modified` o ETag.

10.2.2.3.3 Notificación de error

Un respondedor del SAML que se niega a llevar a cabo un intercambio de mensajes con el peticionario del SAML tiene que devolver una respuesta "`403 Forbidden`" (prohibido 403). En este caso, el contenido del cuerpo HTTP no es significativo.

Como se describe en el documento SOAP W3C, 6.2, en caso de un error de SOAP durante el procesamiento de una petición SOAP, el servidor HTTP SOAP tiene que devolver una respuesta "`500 Internal Server Error`" (error de servidor interno 500) e incluir un mensaje SOAP con un elemento SOAP `<SOAP-ENV:fault>`. Este tipo de error debería ser devuelto en los casos de errores relacionados con SOAP que sean detectados antes de pasar el control al procesador del SAML, o cuando el procesador de SOAP comunique un error interno (por ejemplo, el espacio de nombre XML SOAP es incorrecto, el esquema del SAML no puede ser localizado, el procesador del SAML emite una excepción, etc.).

NOTA (informativa) – En PE19 (véase [OASIS Document Errata]) se sugiere sustituir la primera oración del párrafo anterior por:

> Como se describe en el documento SOAP W3C, 6.2, en caso de un error de SOAP durante el procesamiento de una petición SOAP, el servidor HTTP SOAP debería devolver una respuesta "`500 Internal Server Error`" (error de servidor interno 500) e incluir un mensaje SOAP con un elemento SOAP `<SOAP-ENV:fault>`.

En caso de un error de procesamiento del SAML, el servidor HTTP SOAP debe responder con "200 OK" e incluir un elemento <samlp:Status> especificado por el SAML en la respuesta del SAML dentro del cuerpo de SOAP. El elemento <samlp:Status> no aparece por sí mismo en el cuerpo de SOAP, sino únicamente dentro de una respuesta SAML.

En la cláusula relativa a aserciones y protocolos del SAML de esta Recomendación figura información adicional acerca de la utilización de los códigos de situación del SAML.

10.2.2.3.4 Consideraciones acerca de los metadatos

El soporte de la vinculación de SOAP debería reflejarse indicando un punto extremo de URL al que se han de enviar las peticiones contenidas en los mensajes SOAP para un protocolo o perfil particular, o alternativamente con una definición de puerto/punto extremo del lenguaje de descripción de servicios disponibles en la web (*WSDL, web services description language*).

10.2.2.3.5 Ejemplo de un intercambio de mensajes del SAML utilizando SOAP por HTTP

El siguiente es un ejemplo de una consulta que solicita una aserción que contiene un enunciado de atributo de una autoridad de atributo del SAML.

```
POST /SamlService HTTP/1.1
Host: www.example.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:AttributeQuery xmlns:samlp:="…"
xmlns:saml="…" xmlns:ds="…" ID="_6c3a4f8b9c2d" Version="2.0"
IssueInstant="2004-03-27T08:41:00Z"
            <ds:Signature> … </ds:Signature>
            <saml:Subject>
            …
            </saml:Subject>
        </samlp:AttributeQuery>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Following is an example of the corresponding response, which supplies an
assertion containing the attribute statement as requested.
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:Response xmlns:samlp="…" xmlns:saml="…" xmlns:ds="…"
ID="_6c3a4f8b9c2d" Version="2.0" IssueInstant="2004-03-27T08:42:00Z">
            <saml:Issuer>https://www.example.com/SAML</saml:Issuer>
            <ds:Signature> … </ds:Signature>
            <Status>
              <StatusCode Value="…"/>
            </Status>

            <saml:Assertion>
                <saml:Subject>
                …
                </saml:Subject>
                <saml:AttributeStatement>
            …
                </saml:AttributeStatement>
             </saml:Assertion>
        </samlp:Response>
    </SOAP-Env:Body>
</SOAP-ENV:Envelope>
```

10.2.3 Vinculación de SOAP (PAOS) inversa

Esta vinculación soporta la vinculación HTTP inversa para la especificación de SOAP (véase PAOS:2003). Los implementadores están obligados a cumplir con las reglas de procesamiento generales que se especifican en PAOS, además de las especificadas en esta Recomendación. En caso de conflicto, se respetarán las normas POAS:2003 de Liberty Alliance.

10.2.3.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

10.2.3.2 Generalidades

La vinculación SOAP inversa constituye un mecanismo útil para que un peticionario HTTP pueda notificar a un peticionario del SAML su capacidad para actuar como un respondedor de SOAP o un intermediario de SOAP. El peticionario HTTP puede soportar un patrón en el que recibe una petición SAML dentro de un sobre SOAP en una respuesta HTTP del peticionario SAML, y responde con una respuesta SAML dentro de un sobre SOAP en una petición HTTP subsiguiente. Este patrón de intercambio de mensajes soporta el caso de utilización que se define en el perfil ECP SSO, en el cual el peticionario HTTP es un intermediario en un intercambio de autenticación.

10.2.3.3 Intercambio de mensajes

La vinculación PAOS incluye dos patrones de intercambio de mensajes:

1) El peticionario HTTP envía una petición HTTP a un peticionario SAML. Este último responde con una respuesta anHTTP que contiene un sobre SOAP que a su vez contiene un mensaje de petición SAML.
2) A continuación, el peticionario HTTP envía una petición HTTP al peticionario SAML original que contiene un sobre SOAP que a su vez contiene un mensaje de respuesta SAML. El peticionario SAML responde con una respuesta HTTP, posiblemente en respuesta a la petición de servicio original del paso 1.

El perfil ECP emplea la vinculación PAOS para proporcionar la autenticación del cliente al proveedor de servicio antes de que se otorgue el servicio. Esto sucede en los siguientes pasos que se ilustran en la figura 10-1.

1) El cliente solicita un servicio mediante una petición HTTP.
2) El proveedor de servicio responde con una petición de autenticación del SAML. Ésta se envía utilizando una petición SOAP que se transporta en la respuesta HTTP.
3) El cliente devuelve una respuesta SOAP en la que se transporta una respuesta de autenticación del SAML. Ésta se envía utilizando una nueva petición HTTP.
4) Suponiendo que la autenticación y la autorización del proveedor de servicio son satisfactorias, éste puede responder a la petición de servicio original en la respuesta HTTP.

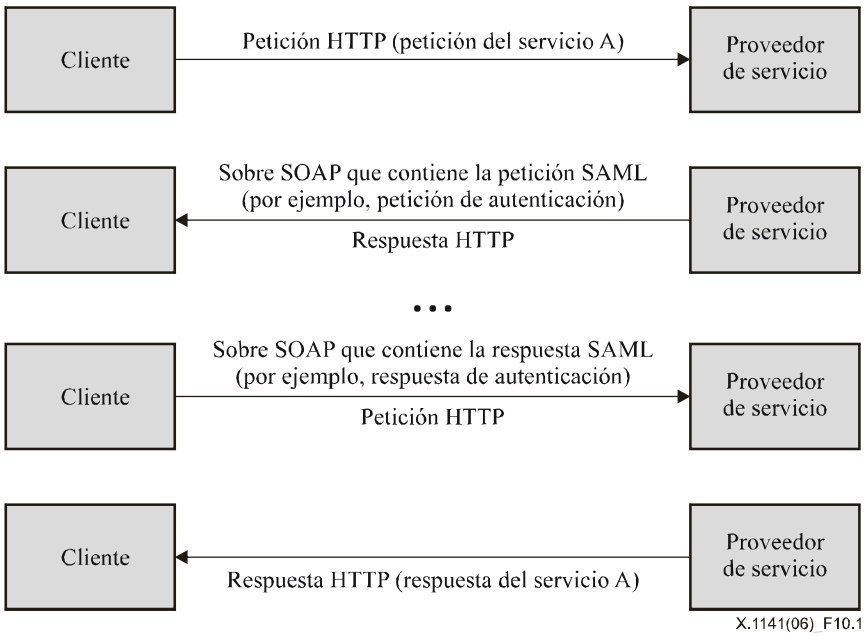

Figura 10-1/X.1141 – Intercambio de mensajes de la vinculación PAOS

El peticionario HTTP notifica su capacidad para manejar la vinculación SOAP inversa en sus peticiones HTTP empleando los encabezamientos HTTP que se definen en la especificación PAOS:2003. Concretamente:

- El campo de encabezamiento `Accept` HTTP debe indicar la capacidad para aceptar el tipo de contenido "`application/vnd.paos+xml`".
- El campo de encabezamiento `PAOS` HTTP debe estar presente y especificar la versión de PAOS con "`urn:liberty:paos:2003-08`" como mínimo.

NOTA 1 (informativa) – En PE21 (véase OASIS PE:2006) se sugiere suprimir "como mínimo" en el párrafo anterior.

Los encabezamientos PAOS adicionales tales como el valor de servicio podrán ser especificadas mediante perfiles que emplean la vinculación PAOS. El peticionario HTTP puede añadir encabezamientos arbitrarios a la petición HTTP.

NOTA 2 – Esta vinculación no define un mecanismo RelayState. Si es necesario, los perfiles específicos que utilizan esta vinculación tendrán que definir, por consiguiente, ese mecanismo. Para esta finalidad se sugiere aplicar el encabezamiento de SOAP.

En las siguientes subcláusulas se proporcionan más detalles acerca de los dos pasos del intercambio de mensajes.

10.2.3.3.1 Petición HTTP, petición SAML en la respuesta SOAP

El respondedor HTTP, en respuesta a una petición HTTP arbitraria, puede devolver un mensaje de petición SAML utilizando esta vinculación para devolver un sobre SOAP 1.1 en la respuesta HTTP que contiene un solo mensaje de petición SAML en el cuerpo de SOAP, sin contenido de cuerpo adicional. El sobre SOAP puede contener encabezamientos SOAP arbitrarios definidos por PAOS, perfiles del SAML o Recomendaciones adicionales.

Aunque el mensaje de petición del SAML se entrega al peticionario HTTP, el destinatario objetivo real puede ser otra entidad del sistema, y el peticionario HTTP actúa como un intermediario, como se define en los perfiles específicos.

10.2.3.3.2 Respuesta del SAML en la petición SOAP, respuesta HTTP

Cuando el peticionario HTTP entrega un mensaje de respuesta SAML al destinatario objetivo mediante la vinculación PAOS, éste se coloca como el único elemento en el cuerpo SOAP dentro del sobre SOAP en una petición HTTP. El peticionario HTTP puede ser el originador de la respuesta SAML, o puede no serlo. El sobre SOAP puede contener encabezamientos SOAP arbitrarios definidos por PAOS, perfiles del SAML o Recomendaciones adicionales. El intercambio del SAML se considera completo y esta vinculación no especifica la respuesta HTTP.

Los perfiles pueden definir restricciones adicionales relativas al contenido HTTP de las respuestas ajenas al SOAP durante los intercambios cubiertos por esta vinculación.

10.2.3.4 Almacenamiento

Los mandatarios HTTP no deben almacenar los mensajes de protocolo del SAML. Para garantizarlo se deben respetar las siguientes reglas.

Cuando se emplea HTTP 1.1, los peticionarios que envían mensajes de protocolo del SAML deberían :

- Incluir un campo de encabezamiento `Cache-Control` fijado a "`no-cache`, `no-store`".
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".

Cuando se emplee HTTP 1.1, los respondedores que devuelven mensajes de protocolo del SAML deberían:

- Incluir un campo de encabezamiento `Cache-Control` fijado a "`no-cache`, `no-store`, `must-revalidate`, `private`".
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".
- Impedir la inclusión de un Validator, tal como una cabecera `Last-Modified` o ETag.

10.2.3.5 Consideraciones de seguridad

El peticionario HTTP en la vinculación PAOS puede actuar como un intermediario SOAP, y de ser así, es posible que la seguridad de la capa de transporte para la autenticación del origen, la integridad y la confidencialidad no pueda cumplir los requisitos de seguridad extremo a extremo. En este caso se recomienda la seguridad en la capa de mensajes SOAP.

NOTA (informativa) – En PE31 (véase OASIS PE:2006) se sugiere cambiar recomienda por RECOMIENDA.

10.2.3.5.1 Notificación de error

Se deben respetar los convenios de error relativos a HTTP normal y SOAP. Los errores que se producen durante el procesamiento del SAML no deben señalarse a la capa HTTP o SOAP sino que deben tratarse utilizando mensajes de respuesta del SAML con un elemento de error `<samlp:Status>`.

10.2.3.5.2 Consideraciones relativas a los metadatos

El soporte de la vinculación PAOS debería reflejarse indicando un punto extremo de URL al que se deben enviar las peticiones HTTP y/o los mensajes de protocolo del SAML contenidos en sobres SOAP para un protocolo o perfil particular. Se puede proporcionar un punto extremo único o distintos puntos extremo de petición y respuesta.

10.2.4 Vinculación de redirección (*redirect*) HTTP

Esta vinculación define un mecanismo que permite la transmisión de los mensajes de protocolo del SAML en parámetros del URL. La longitud permitida para el URL es teóricamente infinita, pero en la práctica se limita de forma imprevisible. Por consiguiente, se necesitan codificaciones especializadas para transportar los mensajes XML en un URL, y se puede enviar contenido de mensaje más largo o complejo mediante POST HTTP o las vinculaciones Artifact.

Esta vinculación puede formarse con la vinculación POST HTTP (véase 10.2.5) y la vinculación Artifact HTTP (véase 10.2.6) para transmitir mensajes de petición y respuesta en un solo intercambio de protocolo gracias a dos vinculaciones diferentes.

Esta vinculación incluye la utilización de una codificación de mensaje. Aunque en la definición de esta vinculación se incluye la definición de una codificación de mensaje particular, se pueden definir y emplear otras.

10.2.4.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

10.2.4.2 Generalidades

La vinculación HTTP Redirect está prevista para los casos en que el peticionario y el respondedor del SAML necesitan comunicarse a través de un agente del usuario HTTP (como se define en RFC 2616 del IETF) que actúa como intermediario. Esto puede ser necesario, por ejemplo, si las partes en la comunicación no comparten un trayecto un trayecto directo de comunicación, o si el respondedor requiere una interacción con el agente del usuario para satisfacer la petición, como cuando el agente del usuario debe autenticarse ante él.

Algunos agentes de usuario HTTP pueden tener la capacidad de desempeñar un cometido más activo en el intercambio de protocolo y pueden soportar otras vinculaciones que emplean HTTP, tales como las vinculaciones SOAP y SOAP inversa. Esta vinculación no adopta nada aparte de las capacidades de un explorador web común.

10.2.4.3 RelayState (Estado de retransmisión)

Cuando se transmite un mensaje de protocolo del SAML con esta vinculación pueden incluirse datos RelayState. El valor no debe exceder de 80 bytes de longitud y la entidad que crea el mensaje debería proteger su integridad independientemente de otras protecciones que puedan o no existir durante su transmisión. La firma no es una solución realista debido a la limitación de espacio, pero como el valor está expuesto a la manipulación de terceros, la entidad debería garantizar que el valor no ha sido alterado mediante la aplicación de una suma de control, un valor seudoaleatorio o medios similares.

NOTA (informativa) – De acuerdo con PE1 (véase OASIS PE:2006), la última oración del párrafo anterior debería rezar lo siguiente:

> Cuando se transmite un mensaje de protocolo del SAML con esta vinculación pueden incluirse datos RelayState. El valor no debe exceder de 80 bytes de longitud y la entidad que crea el mensaje debería proteger su integridad, bien sea mediante una firma digital (véase la cláusula 10) o por medios independientes.

Si un mensaje de petición del SAML incluye datos RelayState, el respondedor del SAML tiene que devolver su respuesta de protocolo SAML utilizando una vinculación que soporte también un mecanismo de RelayState, y debe colocar los datos exactos que recibió con la petición en el parámetro RelayState correspondiente en la respuesta.

Si en un mensaje de petición del SAML no está incluido dicho valor, o si el mensaje de respuesta del SAML se genera sin una petición correspondiente, el respondedor del SAML puede incluir datos RelayState que han de ser interpretados por el destinatario basándose en la aplicación de un perfil o en un acuerdo previo entre las partes.

10.2.4.4 Codificación del mensaje

Los mensajes se codifican para que puedan ser utilizados con esta vinculación aplicando una técnica de codificación en URL, y transmitidos empleando el método GET HTTP. Hay varios métodos posibles para codificar XML en un URL, dependiendo de las restricciones que estén en vigor. En esta Recomendación se define uno de esos métodos sin excluir los otros. Los puntos extremo de la vinculación deberían, cuando proceda, indicar qué codificaciones soportan mediante los metadatos. Cuando se definen codificaciones particulares, éstas deben identificarse de forma única con un URI. No es obligatorio que todos los mensajes del SAML posibles puedan ser codificados con un conjunto de reglas particular, pero las reglas deben indicar con claridad qué mensajes o contenido pueden o no ser codificados.

Una codificación en URL debe colocar el mensaje completamente dentro de la cadena de consultas URL, y ha de reservar el resto del URL para el punto extremo del destinatario del mensaje.

Se reserva un parámetro de la cadena de consulta denominado `SAMLEncoding` para identificar el mecanismo de codificación empleado. Si se omite este parámetro, se supone que el valor será `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`.

Todos los puntos extremo que soportan esta vinculación han de soportar también la codificación DEFLATE que se describe a continuación.

i) Codificación DEFLATE (desinflar)

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`

Los mensajes de protocolo del SAML pueden ser codificados en un URL gracias al método de compresión DEFLATE (RFC 1951 del IETF). En este tipo de codificación, debería aplicarse el siguiente procedimiento a la serialización original XML del mensaje de protocolo del SAML:

1) Cualquier firma en el mensaje de protocolo del SAML, incluyendo el propio elemento XML `<ds:Signature>`, debe ser suprimida. Si el contenido del mensaje incluye otra firma, tal como una aserción SAML firmada, esta firma incorporada no se suprime. Sin embargo, la longitud de ese mensaje tras la codificación excluye esencialmente la utilización de este mecanismo. Por consiguiente, los mensajes de protocolo SAML que incluyen contenido firmado no deberían ser codificados con este mecanismo.
2) A continuación se aplica el mecanismo de compresión DEFLATE, especificado en RFC 1951 del IETF, a todo el resto del contenido XML del mensaje original de protocolo del SAML.
3) Los datos comprimidos se codifican ulteriormente en base 64 de conformidad con las reglas especificadas en RFC 2045 del IETF. Los cambios de línea o los espacios en blanco tendrán que ser suprimidos del resultado.

4) Los datos codificados en base 64 se codifican enseguida en URL, y se añaden al URL como un parámetro de cadena de consulta que ha de denominarse `SAMLRequest` (si el mensaje es una petición del SAML) o `SAMLResponse` (si el mensaje es una respuesta del SAML).

5) Si los datos RelayState van a acompañar al mensaje de protocolo del SAML, deben codificarse en URL y colocarse en un parámetro de cadena de consulta adicional denominado `RelayState`.

6) Si el mensaje de protocolo del SAML original fue firmado con una firma digital XML, es necesario agregar una nueva firma que abarque los datos codificados especificados arriba mediante las reglas que se establecen a continuación.

Las firmas digitales XML no se codifican en URL directamente de conformidad con las reglas anteriores, debido a cuestiones de espacio. Si el mensaje de protocolo SAML subyacente está firmado con una firma XML, el formato del mensaje codificado en URL debe firmarse de la siguiente manera:

1) El identificador del algoritmo de firma tiene que incluirse como un parámetro de cadena de consulta adicional, denominado `SigAlg`. El valor de este parámetro debe ser un URI que identifique el algoritmo utilizado para firmar el mensaje de protocolo del SAML codificado en URL, y especificado de conformidad con la firma XML o cualquier Recomendación que rija el algoritmo.

2) Para estructurar la firma, se construye una cadena que consiste en la concatenación de parámetros de cadena de consulta `RelayState` (si está presente), `SigAlg` y `SAMLRequest` (o `SAMLResponse`) (cada uno codificado en URL) en una de las siguientes formas (ordenadas como se indica a continuación):

   a) `SAMLRequest=value&RelayState=value&SigAlg=value`
   `SAMLResponse=value&RelayState=value&SigAlg=value`

   b) La cadena de bytes resultante es la cadena de octetos que se va a introducir en el algoritmo de firma. Cualquier otro contenido en la cadena de consulta original no está incluido y no está firmado.

   c) El valor de la firma se codificará en base 64 (véase RFC 2045 del IETF) suprimiendo cualquier espacio en blanco, y se incluirá como un parámetro de cadena de consulta denominado `Signature`. Algunos caracteres en el valor de la firma codificada en base 64 pueden exigir la codificación en URL antes de que se añadan.

   d) Con este mecanismo de codificación es necesario soportar los siguientes algoritmos de firma (véanse las reglas de firma del W3C) y sus representaciones en URI:

      - DSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#dsa-sha1
      - RSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#rsa-sha1

   NOTA – El NIST (National Institute of Standards and Technology) fomenta la utilización de SHA-256 (algoritmo de troceo seguro con claves codificadas de 256 bits) en lugar de SHA-1.

Cuando se verifican las firmas, esta vinculación no implica que se verifique el orden de los parámetros de la cadena de consulta en el URL resultante. Los parámetros pueden aparecer en cualquier orden. Antes de verificar una firma, si la hubiere, la parte confiante debe garantizar que los valores de los parámetros que se van a verificar están ordenados como lo exigen las reglas de firma antes mencionadas.

La codificación URL no es canónica; es decir, hay múltiples codificaciones válidas para un determinado valor. Por lo tanto, la parte confiante ejecutará la etapa de verificación con los valores originales codificados en URL que recibió en la cadena de consulta. No basta con volver a codificar los parámetros tras haber sido procesados por el programa informático, porque es posible que la codificación resultante no concuerde con la codificación del firmante.

Si no existe un valor `RelayState`, habría que omitir todo el parámetro del cómputo de la firma (y no se ha de incluir como un nombre de parámetro vacío).

10.2.4.5 Intercambio de mensajes

El modelo de sistema que se emplea para las conversaciones del SAML a través de esta vinculación es del tipo petición-respuesta, aunque los mensajes se envían al agente usuario en una respuesta HTTP y se entregan al destinatario del mensaje en una petición HTTP. Las interacciones de HTTP antes, entre y después de que se llevan a cabo estos intercambios, no están especificadas. Puede suponerse que tanto el peticionario como el respondedor del SAML son respondedores HTTP. Véase el diagrama de secuencias (figura 10-2) en el que se ilustra el intercambio de mensajes.

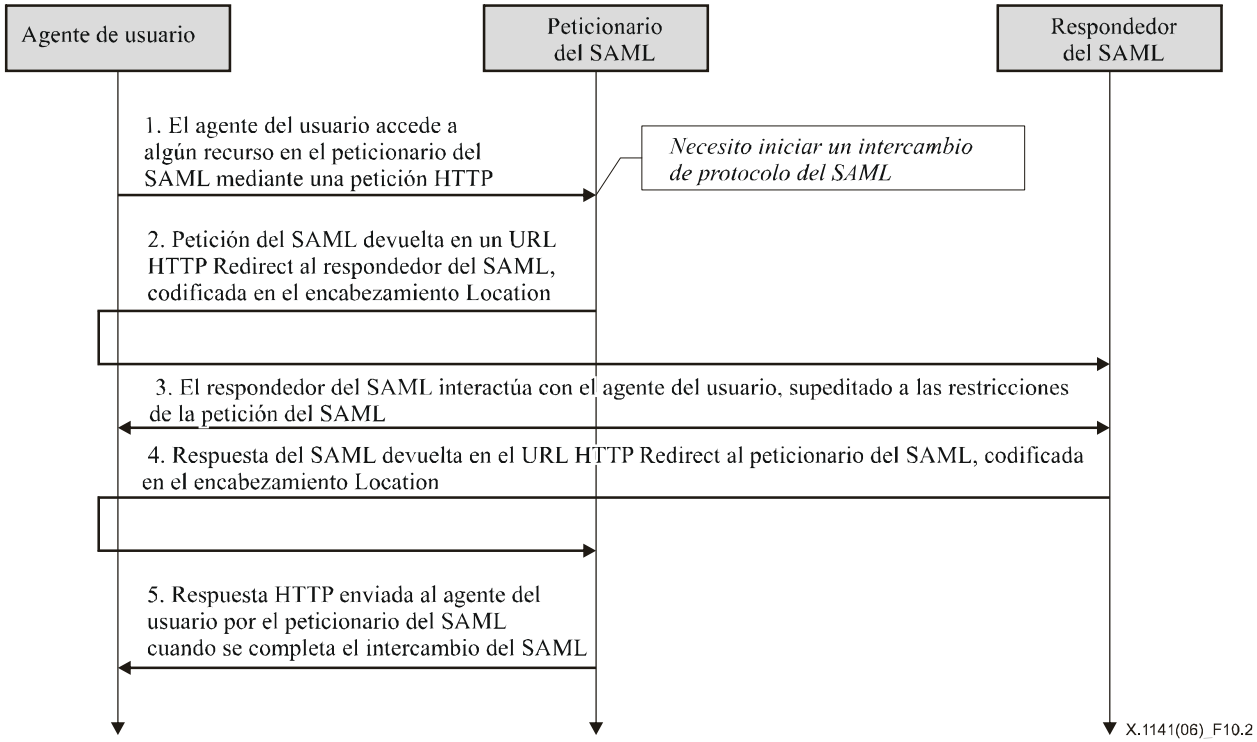

Figura 10-2/X.1141 – Intercambio de mensajes HTTP redirect

1) Inicialmente, el agente del usuario genera una petición HTTP arbitraria a una entidad del sistema. Durante el procesamiento de la petición, la entidad del sistema decide iniciar un intercambio de protocolos del SAML.
2) La entidad del sistema que actúa como un peticionario del SAML da respuesta a la petición HTTP desde el agente del usuario en el paso 1 devolviendo una petición del SAML. Ésta se devuelve codificada en el encabezamiento Location de la respuesta HTTP, y la situación HTTP debe ser 303 ó 302. El peticionario del SAML podrá incluir presentación y contenido adicional en la respuesta HTTP a fin de facilitar la transmisión del mensaje del agente de usuario, como se define en RFC 2616 del IETF. El agente de usuario entrega la petición del SAML generando una petición GET HTTP al respondedor del SAML.
3) En general, el respondedor del SAML puede dar respuesta a la petición del SAML devolviendo inmediatamente una respuesta del SAML o bien puede devolver contenido arbitrario para facilitar la interacción subsiguiente con el agente de usuario que es necesaria para cumplir con la petición. Los protocolos y perfiles específicos pueden incluir mecanismos para señalar el nivel de disposición del peticionario para autorizar esta clase de interacción (por ejemplo, el atributo `IsPassive` en `<samlp:AuthnRequest>`).
4) Finalmente, el respondedor debería devolver una respuesta del SAML al agente del usuario que ha de ser devuelta al peticionario del SAML. Esta respuesta se devuelve de la misma manera descrita para la petición del SAML en el paso 2.
5) Cuando el peticionario del SAML recibe la respuesta del SAML, devuelve una respuesta HTTP arbitraria al agente del usuario.

10.2.4.5.1 HTTP y consideraciones de almacenamiento

Los mandatarios HTTP y el intermediario del agente del usuario no deberían almacenar los mensajes del protocolo del SAML. Para garantizarlo, se deben respetar las siguientes reglas.

Cuando se devuelven mensajes del protocolo del SAML mediante HTTP 1.1, los respondedores deberían:

- Incluir un campo de encabezamiento `Cache-Control` (control de almacenamiento) fijado a "`no-cache`, `no-store`" (sin almacenamiento en memoria intermedia, sin almacenamiento).
- Incluir un campo de encabezamiento Pragma fijado a "`no-cache`".

No se aplica ninguna otra restricción a la utilización de los encabezamientos HTTP.

10.2.4.5.2 Consideraciones de seguridad

La presencia del intermediario del agente del usuario significa que el peticionario y el respondedor no pueden confiar en la capa de transporte para efectos de autenticación, integridad y confidencialidad de extremo a extremo. Los mensajes codificados en URL podrán ser firmados para proporcionar autenticación e integridad de origen si el método de codificación especifica un método de firma.

Si el mensaje está firmado, el atributo XML `Destination` en el elemento SAML raíz del mensaje del protocolo debe contener el URL que recibirá el mensaje del agente del usuario por instrucciones del emisor. A continuación, el destinatario ha de verificar que el valor concuerda con el emplazamiento en el que se recibió el mensaje.

Esta vinculación no debería emplearse si el contenido de la petición o respuesta no debe exponerse al intermediario del agente del usuario. En los demás casos, la confidencialidad de las peticiones y respuestas del SAML son facultativas y depende del entorno de uso. Si la confidencialidad es necesaria, se debería utilizar TLS 1.0 para proteger el mensaje en tránsito entre el agente de usuario y el peticionario y el respondedor del SAML.

Los mensajes codificados en URL pueden quedar expuestos en una variedad de registros HTTP y en el encabezamiento "Referrer" HTTP.

Previo al despliegue, sería conveniente analizar la vulnerabilidad de cada combinación de los mecanismos de autenticación, integridad del mensaje y confidencialidad en el contexto del intercambio de protocolos específico (véase el apéndice I).

En general, esta vinculación se apoya en la protección de autenticación e integridad en el nivel de mensajes a través de firmas y no soporta la confidencialidad de los mensajes proveniente del intermediario del agente del usuario.

10.2.4.6 Notificación de error

Si un respondedor del SAML se niega a realizar un intercambio de mensajes con el peticionario del SAML, debería devolver un mensaje de respuesta del SAML con un valor `<samlp:StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

En las interacciones HTTP durante el intercambio de mensajes no se deben utilizar códigos de situación de error HTTP para indicar fallos de procesamiento del SAML, ya que el agente del usuario no es una parte plena para el intercambio de protocolos del SAML. Véase también la cláusula 9.

10.2.4.7 Consideraciones relativas a los metadatos

El soporte de la vinculación HTTP Redirect debería reflejarse señalando los puntos extremo del URL a los que se deberían enviar las peticiones y respuestas de un protocolo o perfil particular. Es posible proporcionar un solo punto extremo o distintos puntos extremo de petición y respuesta.

NOTA (informativa) – En PE2 (véase OASIS PE:2006) se estipula que el párrafo anterior se debe sustituir por:

El soporte para recibir mensajes utilizando la vinculación Artifact HTTP debería reflejarse señalando los puntos extremo del URL a los que se deberían enviar las peticiones y respuestas de un protocolo o perfil particular. Es posible proporcionar un solo punto extremo o distintos puntos extremo de petición y respuesta. El soporte para enviar mensajes utilizando esta vinculación debería complementarse con uno o varios puntos extremo `<md:ArtifactResolutionService>` indexados para el procesamiento de los mensajes `<samlp:ArtifactResolve>`.

10.2.4.8 Ejemplo de intercambio de mensajes del SAML mediante HTTP Redirect

En este ejemplo, se intercambia un par de mensajes `<LogoutRequest>` y `<LogoutResponse>` gracias a la vinculación HTTP Redirect.

En primer lugar, a continuación se presentan los mensajes del protocolo del SAML reales que se han de intercambiar:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
```

```
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Esta vinculación no define la petición HTTP inicial del agente del usuario en el paso 1. El peticionario del SAML, para iniciar el intercambio de protocolos de fin de sesión, devuelve la siguiente respuesta HTTP, que contiene un mensaje de petición del SAML firmado. El valor del parámetro `SAMLRequest` se deriva realmente del mensaje de petición anterior. La parte de la firma es únicamente ilustrativa y no el resultado de un cómputo real. Los cambios de línea en el encabezamiento `Location` HTTP que se muestra a continuación representan un artefacto del documento, y no existen cambios de línea en el valor del encabezamiento real.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLRequest=fVFdS8MwFH0f7D%2BU
vGdNsq62oSsIQyhMESc%2B%2BJYlmRbWpObeyvz3puv2IMjyFM7HPedyK1DdsZdb%2F%2BEHfLF
fgwVMTt3RgTwzazIEJ72CFqRTnQWJWu7uH7dSLJjsg0ev%2FZFMlttiBWADtt6R%2BSyJr9msiR
H7O70sCm31Mj%2Bo%2BC%2B1KA5GlEWeZaogSQMw2MYBKodrIhjLKONU8FdeSsZkVr6T5M0GiHM
jvWCknqZXZ2OoPxF7kGnaGOuwxZ%2Fn4L9bY8NC%2By4du1XpRXnxPcXizSZ58KFTeHujEWkNPZ
ylsh9bAMYYUjO2Uiy3jCpTCMo5M1StVjmN9SO150sl9lU6RV2Dp0vsLIy7NM7YU82r9B90PrvCf
85W%2FwL8zSVQzAEAAA%3D%3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAl
g=http%3A%2F%2Fwww.w3.org%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

Una vez llevado a cabo un número no especificado de interacciones, el respondedor del SAML devuelve la respuesta HTTP que se presenta más abajo, la cual contiene el mensaje de respuesta del SAML firmado. Una vez más, el valor del parámetro `SAMLResponse` se deriva realmente del mensaje de respuesta anterior. La parte de la firma es únicamente ilustrativa y no el resultado de un cómputo real.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLResponse=fVFNa4QwEL0X%2B
h8k912TaDUGFUp7EbZQ6rKH3mKcbQVNJBOX%2FvxaXQ9tYec0vHlv3nzkqIZ%2BlAf7YSf%2FBj
hagxB8Db1BuZQKMjkjrcIOpVEDoPRa1o8vB8n3VI7OeqttT1bJbbJCBOc7a8j9XTBH9VyQhqYRb
TlrEi4Yo61oUqA0pvShYZHiDQkqs411tAVpeZPqSAgNOkrOas4zzcW55ZlI4liJrTXiBJVBr4wv
CJ877ijbcXZkmaRUxtk7CU7gcB5mLu8pKVddvghd%2Ben9iDIMa3CXTsOrs5euBbfXdgh%2F9sn
DK%2FEqW69Ye%2BUnvGL%2F8CfbQnBS%2FQS3z4QLW9aT1oBIws0j%2FGOyAb9%2FV34Dw5k779
IBAAA%3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAlg=http%3A%2F%2Fww
w.w3.org%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

10.2.5 Vinculación POST HTTP

Esta vinculación define un mecanismo que permite la transmisión de los mensajes del protocolo del SAML dentro del contenido de un control de formato HTML codificado en base 64.

Esta vinculación puede formarse con la vinculación Redirect HTTP (véase 10.2.4) y la vinculación Artifact HTTP (véase 10.2.6) para transmitir mensajes de petición y respuesta en un solo intercambio de protocolos mediante dos vinculaciones diferentes.

10.2.5.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

10.2.5.2 Generalidades

La vinculación POST HTTP está prevista para los casos en que el peticionario y el respondedor del SAML necesitan comunicarse a través de un agente del usuario HTTP (como se define en RFC 2616 del IETF) que actúa como intermediario. Esto puede ser necesario, por ejemplo, si las partes en la comunicación no comparten un trayecto directo

de comunicación, o si el respondedor requiere una interacción con el agente del usuario para satisfacer la petición, como cuando el agente del usuario debe autenticarse ante él.

Algunos agentes de usuario HTTP pueden tener la capacidad para desempeñar un cometido más activo en el intercambio de protocolo y pueden soportar otras vinculaciones que emplean HTTP, tal como las vinculaciones SOAP y SOAP inversa. Esta vinculación no adopta nada aparte de las capacidades de un explorador web común.

10.2.5.3 RelayState

Cuando se transmite un mensaje de protocolo del SAML con esta vinculación pueden incluirse datos RelayState. El valor no debe exceder de 80 bytes de longitud y la entidad que crea el mensaje debería proteger su integridad independientemente de otras protecciones que puedan o no existir durante su transmisión. La firma no es una solución realista debido a la limitación de espacio, pero como el valor está expuesto a la manipulación de terceros, la entidad debería garantizar que el valor no ha sido alterado mediante la aplicación de una suma de control, un valor seudoaleatorio o medios similares.

Si un mensaje de petición del SAML va acompañado por datos RelayState, el respondedor del SAML tiene que devolver su respuesta de protocolo SAML utilizando una vinculación que soporte también un mecanismo de RelayState, y debe colocar los datos exactos que recibió con la petición en el parámetro RelayState correspondiente en la respuesta.

Si en un mensaje de petición del SAML no está incluido dicho valor, o si el mensaje de respuesta del SAML se genera sin una petición correspondiente, el respondedor del SAML puede incluir datos RelayState que han de ser interpretados por el destinatario basándose en la aplicación de un perfil o en un acuerdo previo entre las partes.

NOTA (informativa) – En PE31 (véase OASIS PE:2006) se sugiere aclarar el párrafo anterior como se indica a continuación:

Si en un mensaje de petición del SAML no está incluido el parámetro RelayState, o si el mensaje de respuesta del SAML se genera sin una petición correspondiente, el respondedor del SAML puede incluir datos RelayState que han de ser interpretados por el destinatario basándose en la aplicación de un perfil o en un acuerdo previo entre las partes.

10.2.5.4 Codificación del mensaje

Para que los mensajes puedan emplear esta vinculación es necesario que codifiquen el XML en un control de formato HTML y se transmitan mediante el método POST HTTP. Un mensaje de protocolo del SAML se codifica en formato aplicando las reglas de codificación en base 64 para su representación en XML y colocando el resultado en un control de formato oculto dentro de un formato que se define en las reglas del HTML del W3C, cláusula 17. El documento HTML debe ser conforme con las reglas XHTML del W3C de acuerdo con las prácticas normales.

Si el mensaje es una petición del SAML, el control del formato debe denominarse `SAMLRequest`. Si el mensaje es una respuesta del SAML, el control del formato debe denominarse `SAMLResponse`. Pueden incluirse controles o presentaciones de formato adicionales pero no deben ser obligatorios para que el destinatario pueda procesar el mensaje.

Si el mensaje de protocolo del SAML tiene que estar acompañado por un valor "RelayState", éste debe colocarse en un control de formato oculto denominado `RelayState` dentro del mismo formato con el mensaje del SAML.

El atributo action del formato debe ser el punto extremo HTTP del destinatario para el protocolo o perfil que utiliza esta vinculación, al que se ha de entregar el mensaje del SAML. El atributo method ha de ser "POST".

Para provocar el envío del formato puede emplearse cualquier técnica soportada por el agente del usuario, y puede incluirse cualquier contenido del formato necesario para dicho soporte, tal como controles de envío e instrucciones scripting (redacción de guiones) en el lado del cliente. Sin embargo, el destinatario debe disponer de la capacidad para procesar el mensaje sin tener en cuenta el mecanismo empleado para enviar el formato.

Es necesario transformar cualquier valor de control del formato para asegurar su inclusión en el documento XHTML. Esto incluye los caracteres de transformación tales como las comillas en las entidades HTML, etc.

10.2.5.5 Intercambio de mensajes

El modelo del sistema que se emplea para las conversaciones del SAML a través de esta vinculación es del tipo petición-respuesta, pero los mensajes son enviados al agente de usuario en una respuesta HTTP y entregados al destinatario del mensaje en una petición HTTP. No se especifican las interacciones que suceden antes, entre y después de estos intercambios. Se supone que tanto el peticionario como el respondedor del SAML son respondedores HTTP. Véase la figura 10-3 en la que se ilustra el intercambio de los mensajes.

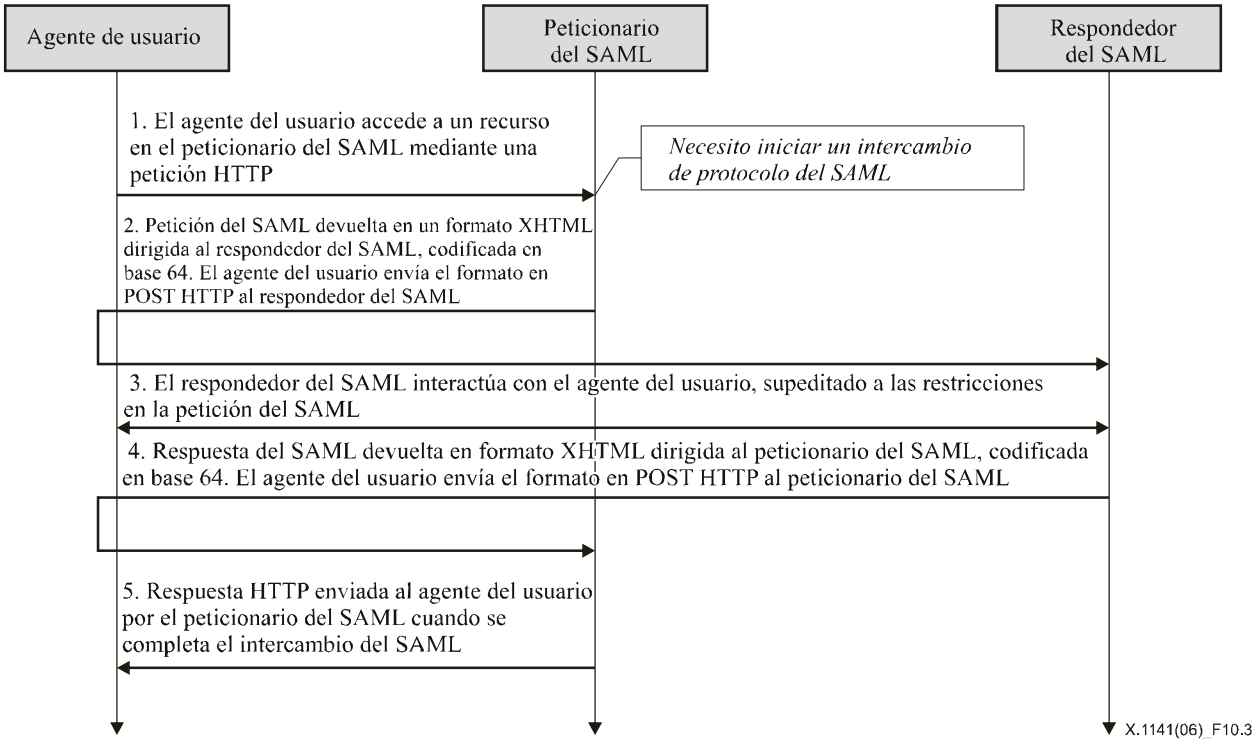

Figura 10-3/X.1141 – Intercambio de mensajes POST HTTP

1) Inicialmente, el agente del usuario genera una petición HTTP arbitraria a una entidad del sistema. Durante el procesamiento de la petición, la entidad del sistema decide iniciar un intercambio de protocolos del SAML.
2) La entidad del sistema que actúa como un peticionario del SAML da respuesta a la petición HTTP desde el agente del usuario devolviendo una petición del SAML. La petición se devuelve en un documento XHTML que contiene el formato y el contenido que se definen en 10.2.5.4. El agente del usuario entrega la petición del SAML enviando una petición POST HTTP al respondedor del SAML.
3) En general, el respondedor del SAML puede dar respuesta a la petición del SAML devolviendo inmediatamente una respuesta del SAML o bien puede devolver contenido arbitrario para facilitar la interacción subsiguiente con el agente de usuario que es necesaria para cumplir con la petición. Los protocolos y perfiles específicos pueden incluir mecanismos para señalar el nivel de disposición del peticionario para autorizar esta clase de interacción (por ejemplo, el atributo `IsPassive` en `<samlp:AuthnRequest>`).
4) Finalmente, el respondedor debería devolver una respuesta del SAML al agente del usuario que ha de ser devuelta al peticionario del SAML. Esta respuesta se devuelve de la misma manera descrita para la petición del SAML en el paso 2.
5) Cuando el peticionario del SAML recibe la respuesta del SAML, devuelve una respuesta HTTP arbitraria al agente del usuario.

10.2.5.5.1 HTTP y consideraciones de almacenamiento

Los mandatarios HTTP y el intermediario del agente del usuario no deberían almacenar los mensajes del protocolo del SAML. Para garantizarlo, se deben respetar las siguientes reglas.

Cuando se devuelven mensajes del protocolo del SAML mediante HTTP 1.1, los respondedores HTTP deberían:

- Incluir un campo de encabezamiento `Cache-Control` (control de almacenamiento) fijado a "`no-cache`, `no-store`" (sin almacenamiento en memoria intermedia, sin almacenamiento).
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".

No se aplica ninguna otra restricción a la utilización de los encabezamientos HTTP.

10.2.5.5.2 Consideraciones de seguridad

La presencia del intermediario del agente del usuario significa que el peticionario y el respondedor no pueden confiar en la capa de transporte en cuanto a la protección de la autenticación, integridad o confidencialidad y por consecuencia

tienen que autenticar los mensajes recibidos. En esos casos, el SAML prevé una firma en los mensajes de protocolo para la autenticación y la integridad. Los mensajes codificados en formato pueden ser firmados antes de aplicar la codificación base 64.

Si el mensaje está firmado, el atributo XML `Destination` en el elemento SAML raíz del mensaje del protocolo debe contener el URL que recibirá el mensaje del agente del usuario por instrucciones del emisor. A continuación, el destinatario ha de verificar que el valor concuerda con el emplazamiento en el que se recibió el mensaje.

Esta vinculación no debería emplearse si el contenido de la petición o respuesta no debe exponerse al intermediario del agente del usuario. En los demás casos, la confidencialidad de las peticiones y respuestas del SAML son facultativas y depende del entorno de uso. Si la confidencialidad es necesaria, se debería utilizar TLS 1.0 para proteger el mensaje en tránsito entre el agente de usuario y el peticionario y el respondedor del SAML.

En general, esta vinculación se apoya en la protección de autenticación e integridad en el nivel de mensajes a través de firmas y no soporta la confidencialidad de los mensajes proveniente del intermediario del agente del usuario.

No se ha definido un mecanismo para proteger la integridad de la relación entre el mensaje del protocolo del SAML y el valor "RelayState", si lo hubiere. Es decir, un atacante, puede, potencialmente, recombinar un par de respuestas HTTP válidas cambiando los valores "RelayState" asociados con cada mensaje de protocolo del SAML. Existe la posibilidad de proteger la integridad de los valores individuales de "RelayState" y del mensaje SAML, pero no de la combinación. Como resultado, el productor y el consumidor de la información "RelayState" deben tener la precaución de no asociar información de situación sensible con el valor de "RelayState" sin tomar medidas adicionales (como las basadas en la información en el mensaje SAML).

10.2.5.6 Notificación de error

Si un respondedor del SAML se niega a realizar un intercambio de mensajes con el peticionario del SAML, debería devolver un mensaje de respuesta con un valor `<samlp:StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

En las interacciones HTTP durante el intercambio de mensajes no se deben utilizar códigos de situación de error HTTP para indicar fallos de procesamiento del SAML, ya que el agente del usuario no es una parte plena para el intercambio de protocolos del SAML.

En la cláusula 8.2 figura más información acerca de los códigos de situación del SAML.

10.2.5.7 Consideraciones relativas a los metadatos

El soporte de la vinculación POST HTTP debería reflejarse señalando los puntos extremo del URL a los que se deberían enviar las peticiones y respuestas de un protocolo o perfil particular. Es posible proporcionar un solo punto extremo o distintos puntos extremo de petición y respuesta.

10.2.5.8 Ejemplo de intercambio de mensajes del SAML mediante HTTP POST

En este ejemplo, se intercambia un par de mensajes `<LogoutRequest>` y `<LogoutResponse>` gracias a la vinculación POST HTTP.

En primer lugar, a continuación se presentan los mensajes del protocolo del SAML reales que se han de intercambiar:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Esta vinculación no define la petición HTTP inicial del agente del usuario en el paso 1. El peticionario del SAML, para iniciar el intercambio de protocolos de fin de sesión, devuelve la siguiente respuesta HTTP, que contiene un mensaje de petición del SAML. El valor del parámetro `SAMLRequest` se deriva del mensaje de petición anterior.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://ServiceProvider.com/SAML/SLO/Browser" method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkxvZ291dFJlcXVlc3QgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1l
czp0YzpTQU1MOjIuMDpwcm90b2NvbCIgeG1sbnM9InVybjpvYXNpczpuYW1lczp0
YzpTQU1MOjIuMDphc3NlcnRpb24iDQogICAgSUQ9ImQyYjdjMzg4Y2VjMzZmYTdj
MzljMjhmZDI5ODY0NGE4IiBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFUMTk6MDA6
NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9JZGVudGl0
eVByb3ZpZGVyLmNvbS9TQU1MPC9Jc3N1ZXI+DQogICAgPE5hbWVJRCBGb3JtYXQ9
InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpuYW1laWQtZm9ybWF0OnBlcnNp
c3RlbnQiPjAwNWEwNmUwLWFkODItMTEwZC1hNTU2LTAwNDAwNWIxM2EyYjwvTmFt
ZUlEPg0KICAgIDxzYW1scDpTZXNzaW9uSW5kZXg+MTwvc2FtbHA6U2Vzc2lvbklu
ZGV4Pg0KPC9zYW1scDpMb2dvdXRSZXF1ZXN0Pg=="/>
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

Una vez llevado a cabo un número no especificado de interacciones, el respondedor del SAML devuelve la respuesta HTTP siguiente, que contiene el mensaje de respuesta del SAML firmado. Una vez más, el valor del parámetro SAMLResponse se deriva del mensaje de respuesta anterior.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://IdentityProvider.com/SAML/SLO/Response"
method="post">
<div>
```

```
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLResponse"
value="PHNhbWxwOkxvZ291dFJlc3BvbnNlIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFt
ZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiIHhtbG5zPSJ1cm46b2FzaXM6bmFtZXM6
dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIg0KICAgIElEPSJiMDczMGQyMWI2MjgxMTBk
OGI3ZTAwNDAwNWIxM2EyYiIgSW5SZXNwb25zZVRvPSJkMmI3YzM4OGNlYzM2ZmE3
YzM5YzI4ZmQyOTg2NDRhOCINCiAgICBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFU
MTk6MDA6NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9T
ZXJ2aWNlUHJvdmlkZXIuY29tL1NBTUw8L0lzc3Vlcj4NCiAgICA8c2FtbHA6U3Rh
dHVzPg0KICAgICAgICA8c2FtbHA6U3RhdHVzQ29kZSBWYWx1ZT0idXJuOm9hc2lz
Om5hbWVzOnRjOlNBTUw6Mi4wOnN0YXR1czpTdWNjZXNzIi8+DQogICAgPC9zYW1s
cDpTdGF0dXM+DQo8L3NhbWxwOkxvZ291dFJlc3BvbnNlPg=="/>
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

10.2.6 Vinculación Artifact HTTP

En la vinculación Artifact HTTP, la petición SAML, la respuesta SAML o ambas se transmiten por referencia mediante un objeto de datos estructurado de tamaño fijo (stand-in) denominado artefacto. Se emplea una vinculación síncrona independiente, tal como la vinculación SOAP SAML, para intercambiar el artefacto por el mensaje de protocolo real utilizando el protocolo de resolución de artefacto que se define en la cláusula 8.

Esta vinculación puede formarse con la vinculación Redirect HTTP (véase 10.2.4) y la vinculación POST HTTP (véase 10.2.5) para transmitir mensajes de petición y respuesta en un solo intercambio de protocolos mediante dos vinculaciones diferentes.

10.2.6.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

10.2.6.2 Generalidades

La vinculación Artifact HTTP está prevista para los casos en que el peticionario y el respondedor necesitan comunicarse a través de un agente de usuario como intermediario, pero las limitaciones del intermediario impiden o desalientan la transmisión de un mensaje completo (o intercambio de mensajes) a través del mismo. Esto puede deberse a motivos técnicos o a la renuencia a exponer el contenido del mensaje al intermediario (y cuando la aplicación de criptación no es práctica).

Debido a la necesidad de resolver ulteriormente el artefacto mediante otra vinculación síncrona, como SOAP, se debe disponer de un trayecto de comunicación directo entre el emisor y el destinatario del mensaje del SAML en la dirección inversa de la transmisión del artefacto (el receptor del mensaje y el artefacto ha de tener la capacidad para devolver una petición `<samlp:ArtifactResolve>` al emisor del artefacto). Asimismo, el emisor del artefacto tiene que mantener el estado mientras el artefacto está pendiente, lo cual presenta implicaciones para los entornos balanceados en carga.

10.2.6.3 Codificación del mensaje

Existen dos métodos de codificación de un artefacto que se han de emplear con esta vinculación. El primero consiste en codificar el artefacto en un parámetro URL y el segundo en colocarlo en un control de formato HTML. Cuando se emplea la codificación en URL, se utiliza el método GET HTTP para entregar el mensaje, mientras que POST se usa con la codificación en formato. Todos los puntos extremo que soportan esta vinculación deben soportar ambas técnicas.

10.2.6.3.1 RelayState

Cuando se transmite un mensaje de protocolo del SAML con esta vinculación pueden incluirse datos RelayState. El valor no debe exceder de 80 bytes de longitud y la entidad que crea el mensaje debería proteger su integridad independientemente de otras protecciones que puedan o no existir durante su transmisión. La firma no es una solución

realista debido a la limitación de espacio, pero como el valor está expuesto a la manipulación de terceros, la entidad debería garantizar que el valor no ha sido alterado mediante la aplicación de una suma de control, un valor seudoaleatorio o medios similares.

Si un artefacto que representa una petición del SAML va acompañado por datos RelayState, el respondedor del SAML tiene que devolver su respuesta de protocolo SAML utilizando una vinculación que soporte también un mecanismo de RelayState, y debe colocar los datos exactos que recibió con el artefacto en el parámetro RelayState correspondiente en la respuesta.

Si en un artefacto que representa una petición del SAML no está incluido dicho valor, o si el mensaje de respuesta del SAML se genera sin una petición correspondiente, el respondedor del SAML puede incluir datos RelayState que han de ser interpretados por el destinatario basándose en la aplicación de un perfil o en un acuerdo previo entre las partes.

10.2.6.3.2 Codificación en un URL

Para codificar un artefacto en un URL, el valor del artefacto se codifica en el URL y se coloca en un parámetro de cadena de consulta denominado `SAMLart`.

Si es necesario que un valor "RelayState" acompañe al artefacto del SAML, éste debe codificarse en URL y colocarse en un parámetro de cadena de consulta adicional denominado `RelayState`.

10.2.6.3.3 Codificación en un formato

Un artefacto del SAML puede codificarse en un formato colocándolo en un control de formato oculto dentro de un formato como se define en las reglas de HTML del W3C. El documento HTML debe respetar XHTML W3C. El control del formato ha de denominarse `SAMLart`. Es posible incluir controles o presentaciones de formato adicionales pero no deben ser obligatorios para que el destinatario pueda procesar el artefacto.

Si es necesario que un valor "RelayState" acompañe al artefacto, éste debe colocarse en un control de formato oculto adicional denominado `RelayState`, dentro del mismo formato con el mensaje del SAML.

El atributo `action` del formato debe ser el punto extremo HTTP del destinatario para el protocolo o perfil que utiliza esta vinculación, al que se ha de entregar el artefacto. El atributo `method` ha de ser "POST".

Para provocar el envío del formato puede emplearse cualquier técnica soportada por el agente del usuario, y puede incluirse cualquier contenido del formato necesario para dicho soporte, tal como controles de envío e instrucciones scripting (redacción de guiones) en el lado del cliente. Sin embargo, el destinatario debe disponer de la capacidad para procesar el artefacto sin tener en cuenta el mecanismo empleado para enviar el formato.

Es necesario transformar cualquier valor de control del formato para asegurar su inclusión en el documento XHTML. Esto incluye los caracteres de transformación tales como las comillas en las entidades HTML, etc.

10.2.6.4 Formato del artefacto

Con relación a esta vinculación, un artefacto es una cadena opaca y corta. Es posible definir y utilizar diferentes tipos sin afectar la vinculación. Las características importantes son la capacidad que tiene un receptor de artefactos para identificar al emisor del artefacto, la resistencia a la manipulación y la falsificación, la singularidad y la compacticidad.

El formato general de cualquier artefacto incluye un código de tipo de artefacto de dos bytes y un valor de índice de dos bytes que identifica un punto extremo específico del servicio de resolución de artefacto del emisor, de la siguiente manera:

```
SAML_artifact     := B64(TypeCode EndpointIndex RemainingArtifact)
TypeCode          := Byte1Byte2
EndpointIndex     := Byte1Byte2
```

La notación `B64 (TypeCode EndpointIndex RemainingArtifact)` representa la aplicación de la transformación de la base 64 (véase RFC 2045 del IETF) a la concatenación de `TypeCode`, `EndpointIndex` y `RemainingArtifact`.

Para la creación de los artefactos del SAML se recomiendan las siguientes prácticas:

- A cada emisor se le asigna un URI de identificación, conocido también como el ID de la entidad del emisor (o proveedor). Véase la cláusula 8 en cuanto a esta clase de identificador.
- El emisor construye el componente `SourceID` del artefacto extrayendo el trozo SHA-1 del URL de identificación. El valor de troceo no se codifica en hexadecimal.

  NOTA 1 – El NIST (National Institute of Standards and Technology) alienta la utilización de SHA-256 (algoritmo de troceo seguro con claves codificadas de 256 bits) en lugar de SHA-1.

- El valor de `MessageHandle` se construye a partir de una secuencia de números aleatoria o seudoaleatoria estricta criptográficamente (véase la norma RFC 1750 del IETF) que genera el usuario. La secuencia consta de valores de al menos 16 bytes de tamaño. Estos valores han de rellenarse según se requiera hasta una longitud de 20 bytes.

  NOTA 2 (informativa) – En PE4 (véase [OASIS Errata Document]) se sugiere añadir el siguiente texto al final del párrafo anterior:

  Aunque la estructura del artefacto general se parece a la utilizada en versiones anteriores del SAML y el código del tipo del formato único descrito más adelante no entra en conflicto con los formatos definidos previamente, no existe una correspondencia explícita entre los artefactos SAML 2.0 y los que se pueden encontrar en especificaciones anteriores, y los formatos de artefacto que no se definen específicamente para su uso con SAML 2.0 no deben utilizarse con esta vinculación.

A continuación se describe el tipo de artefacto único definido por el SAML V2.0.

10.2.6.4.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:artifact-04`

Persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

10.2.6.4.2 Detalles del formato

El SAML V2.0 define un tipo de artefacto con código de tipo 0x0004. Este tipo se define como sigue:

```
TypeCode          := 0x0004
RemainingArtifact := SourceID MessageHandle
SourceID          := 20-byte_sequence
MessageHandle     := 20-byte_sequence
```

`SourceID` representa una secuencia de 20-bytes que es útil para que el receptor del artefacto pueda determinar la identidad del emisor del artefacto y el conjunto de los posibles puntos extremo de resolución.

Se supone que el sitio de destino dispondrá de un cuadro de valores `SourceID` así como de uno o varios puntos extremo de URL indexados (o direcciones) para el respondedor del SAML correspondiente. Para este fin puede emplearse la cláusula 9. Cuando el receptor obtiene el artefacto del SAML, determina si el `SourceID` pertenece a un emisor de artefactos conocido y obtiene el emplazamiento del respondedor del SAML utilizando el `EndpointIndex` antes de enviarle un mensaje `<samlp:ArtifactResolve>` del SAML.

Dos emisores de artefactos cualesquiera con un receptor común están obligados a emplear valores de `SourceID` distintos. La construcción de valores `MessageHandle` se rige por el principio de que no deberían tener una relación previsible con el contenido del mensaje referenciado en el sitio de emisión y de que no debe ser viable construir o suponer el valor a partir del tratamiento de un mensaje pendiente válido.

10.2.6.5 Intercambio de mensajes

El modelo del sistema empleado para las conversaciones del SAML por medio de esta vinculación es del tipo petición-respuesta donde una referencia de artefacto toma el lugar del contenido del mensaje real, y esta referencia se envía al agente del usuario en una respuesta HTTP y se entrega al destinatario del mensaje en una petición HTTP. No se especifican las interacciones HTTP que puedan suceder antes, entre y después de que se llevan a cabo estos intercambios. Se supone que el peticionario y el respondedor del SAML son respondedores HTTP.

Adicionalmente, se supone que cuando el destinatario recibe un artefacto a través del agente del usuario, el primero solicita un intercambio directo e independiente con el emisor del artefacto aprovechando el protocolo de resolución de artefacto que se define en esta Recomendación. Este intercambio ha de emplear una vinculación que no haga uso del agente del usuario HTTP como intermediario, como la vinculación SOAP. Cuando se recibe satisfactoriamente un mensaje de protocolo del SAML, el artefacto se descarta y se reanuda el procesamiento del intercambio de protocolos del SAML primario (o termina si el mensaje es una respuesta).

La emisión y entrega de un artefacto y la etapa de resolución subsiguiente, constituyen la mitad del intercambio de protocolos del SAML. Esta vinculación puede ser útil para entregar una o ambas mitades del intercambio de protocolos del SAML. Es factible emplear una vinculación formada por ese tipo de intercambio, tal como la vinculación HTTP Redirect (véase 10.2.4) o POST (véase 10.2.5), para transportar la otra mitad del intercambio. En la secuencia que se presenta a continuación se supone que se utiliza la vinculación Artifact para ambas mitades. Véase la figura 10-4 a continuación que ilustra el intercambio de mensajes.

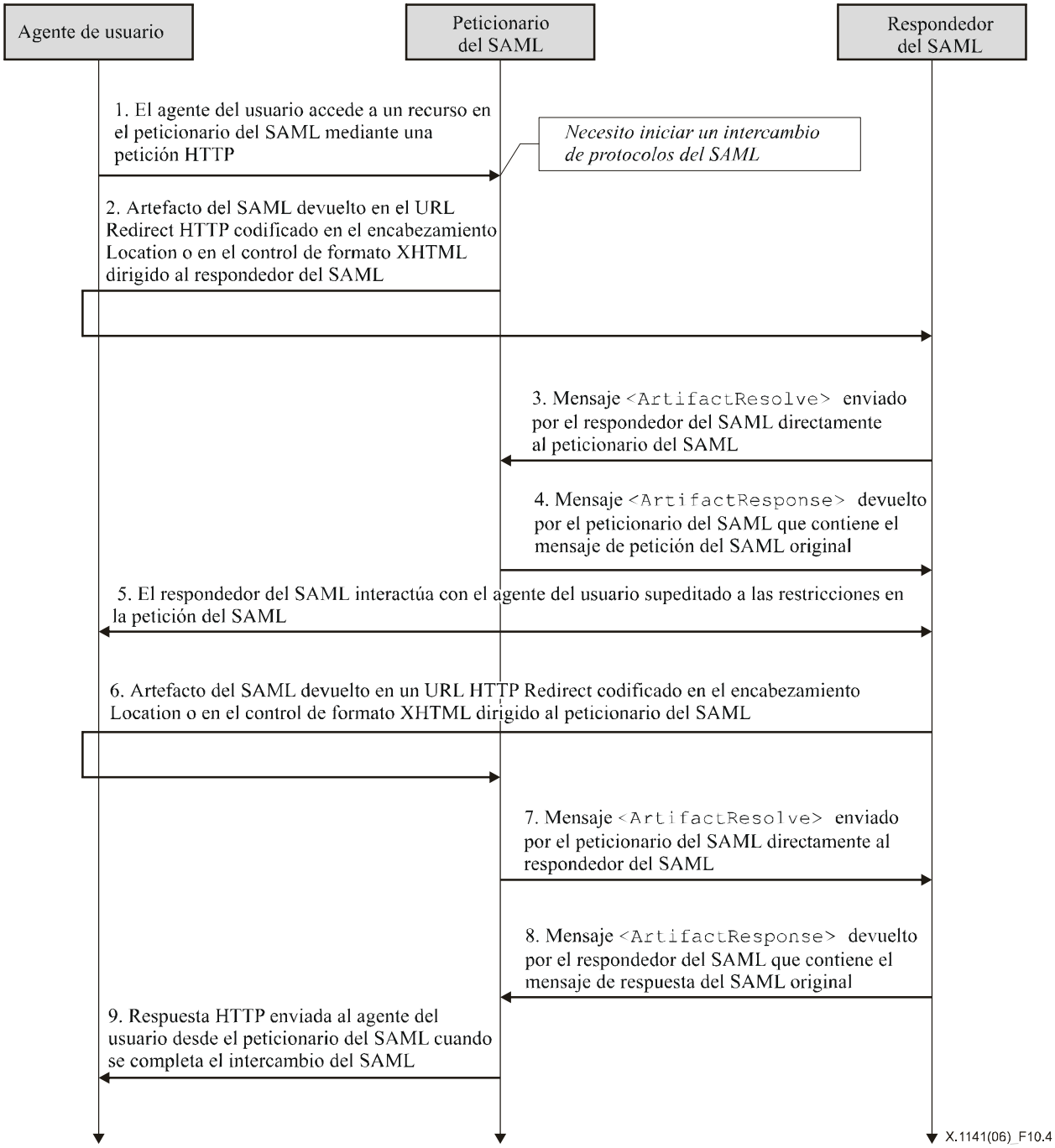

Figura 10-4/X.1141 – Intercambio de mensajes Artifact HTTP

1) Inicialmente, el agente del usuario genera una petición HTTP arbitraria a una entidad del sistema. Durante el procesamiento de la petición, la entidad del sistema decide iniciar un intercambio de protocolos del SAML.

2) La entidad del sistema que actúa como peticionario del SAML responde a una petición HTTP desde el agente del usuario devolviendo un artefacto que representa una petición del SAML.

   - Si el artefacto está codificado en URL, se devolverá codificado en el encabezamiento Location de la respuesta HTTP, y la situación HTTP debe ser 303 ó 302. El peticionario del SAML puede incluir presentación y contenido adicional en la respuesta HTTP a fin de facilitar la transmisión del mensaje del agente del usuario, como se define en RFC 2616 del IETF. El agente del usuario entrega el artefacto enviando una petición GET HTTP al respondedor del SAML.
   - Si el artefacto está codificado en formato, se devuelve en un documento XHTML que contiene el formato y el contenido que se definen en 10.2.6.3.3. El agente del usuario entrega el artefacto enviando una petición POST HTTP al respondedor del SAML.

3) El respondedor del SAML determina el peticionario del SAML examinando el artefacto (el proceso exacto depende del tipo de artefacto), y genera una petición `<samlp:ArtifactResolve>` que contiene el artefacto al peticionario del SAML mediante una vinculación del SAML directa, invirtiendo los papeles temporalmente.

4) Suponiendo que se satisfacen las condiciones necesarias, el peticionario del SAML devuelve una `<samlp:ArtifactResponse>` que contiene el mensaje de petición del SAML original que desea que sea procesado por el respondedor del SAML.

5) En general, el respondedor del SAML puede dar respuesta a la petición del SAML devolviendo inmediatamente un artefacto del SAML o bien puede devolver contenido arbitrario para facilitar la interacción subsiguiente con el agente de usuario que es necesaria para cumplir con la petición. Los protocolos y perfiles específicos pueden incluir mecanismos para señalar el nivel de disposición del peticionario para autorizar esta clase de interacción (por ejemplo, el atributo `IsPassive` en `<samlp:AuthnRequest>`).

6) Finalmente, el respondedor podría devolver un artefacto del SAML al agente del usuario para que sea devuelto al peticionario del SAML. El artefacto de respuesta del SAML se devuelve de la misma manera descrita para el artefacto de petición del SAML en el paso 2.

7) El peticionario del SAML determina el respondedor del SAML examinando el artefacto, y emite una petición `<samlp:ArtifactResolve>` que contiene el artefacto al respondedor del SAML utilizando una vinculación del SAML directa, como en el paso 3.

   NOTA (informativa) – En PE31 (véase OASIS PE:2006) se sugiere sustituir la última oración del paso 6 por:

   El peticionario del SAML determina el respondedor del SAML examinando el artefacto, y emite una petición `<samlp:ArtifactResolve>` que contiene el artefacto al respondedor del SAML utilizando una vinculación del SAML síncrona, como en el paso 3.

8) Suponiendo que se satisfacen las condiciones necesarias, el respondedor del SAML devuelve una `<samlp:ArtifactResponse>` que contiene el mensaje de respuesta del SAML que desea que sea procesado por el peticionario, como en el paso 4.

9) Cuando el peticionario del SAML recibe la respuesta del SAML, devuelve una respuesta HTTP arbitraria al agente del usuario.

10.2.6.5.1 HTTP y consideraciones de almacenamiento

Los mandatarios HTTP y el intermediario del agente del usuario no deberían almacenar los artefactos del SAML. Para garantizarlo, se deberían respetar las siguientes reglas.

Al devolver artefactos del SAML mediante HTTP 1.1, los respondedores HTTP deberían:

- Incluir un campo de encabezamiento `Cache-Control` fijado a "`no-cache, no-store`".
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".

No se aplica ninguna otra restricción a la utilización de los encabezamientos HTTP.

10.2.6.5.2 Consideraciones de seguridad

Para devolver el mensaje en curso, esta vinculación combina una transmisión indirecta de una referencia de mensaje seguida por un intercambio directo. Como resultado, no es necesario autenticar la propia referencia del mensaje (artefacto) ni proteger su identidad, pero puede ser que el intercambio de petición/respuesta de la llamada de retorno que devuelve el mensaje en cuestión sea autenticado mutuamente y tenga protección de integridad, dependiendo del entorno de utilización.

Si el mensaje de protocolo del SAML está dirigido a un destinatario específico, el emisor del artefacto debe autenticar al emisor del mensaje `<samlp:ArtifactResolve>` subsiguiente antes de devolver dicho mensaje.

La transmisión de un artefacto al agente del usuario y desde el mismo debería protegerse mediante confidencialidad, o debería emplearse TLS 1.0. Se puede proteger el intercambio de petición/respuesta de la llamada de retorno que devuelve el mensaje en cuestión, dependiendo del entorno de utilización.

En general, esta vinculación se apoya en el artefacto como una referencia de corto plazo que es difícil falsificar y aplica otras medidas de seguridad al intercambio de petición/respuesta de la llamada de retorno que devuelve el mensaje en curso. Todos los artefactos deben tener una semántica de uso único impuesta por el emisor del artefacto.

Además, es recomendable que los receptores de artefactos impongan también una semántica de uso único a los valores de los artefactos que reciben, a fin de evitar que un atacante pueda interferir en la resolución de un artefacto a través de un agente de usuario y la retransmita al receptor del artefacto. Si un intento de resolución de un artefacto no se completa

satisfactoriamente, el artefacto debería colocarse en una lista de artefactos bloqueados por un periodo de tiempo que exceda un periodo de aceptación razonable mientras el emisor del artefacto resuelve el artefacto.

No se ha definido un mecanismo para proteger la integridad de la relación entre el artefacto y el valor "RelayState", si lo hubiere. Es decir, un atacante puede volver a combinar, potencialmente, un par de respuestas HTTP válidas cambiando los valores "RelayState" asociados con cada uno de los artefactos. Como resultado, el productor/consumidor de la información "RelayState" ha de tener cuidado de no asociar información de estado sensible con el valor "RelayState" sin tomar precauciones adicionales (tales como las basadas en la información del mensaje de protocolo del SAML recuperado a través de un artefacto).

10.2.6.6 Notificación de error

Si un respondedor del SAML se niega a realizar un intercambio de mensajes con el peticionario del SAML, debería devolver un mensaje de respuesta con un valor `<samlp:StatusCode>` de segundo nivel de `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

En las interacciones HTTP durante el intercambio de mensajes no se deben utilizar códigos de situación de error HTTP para indicar fallos de procesamiento del SAML, ya que el agente del usuario no es una parte plena para el intercambio de protocolos del SAML.

Si el emisor de un artefacto recibe un mensaje `<samlp:ArtifactResolve>` que puede comprender, tiene que devolver una `<samlp:ArtifactResponse>` con un valor `<samlp:StatusCode>` de `urn:oasis:names:tc:SAML:2.0:status:Success`, aunque no devuelva el mensaje correspondiente (por ejemplo, porque el peticionario del artefacto no está autorizado a recibir el mensaje o porque el artefacto ya no es válido).

10.2.6.7 Consideraciones relativas a los metadatos

El soporte de la vinculación Artifact HTTP debería reflejarse señalando los puntos extremo del URL a los que se deberían enviar las peticiones y respuestas de un protocolo o perfil particular. Es posible proporcionar un solo punto extremo o distintos puntos extremo de petición y respuesta. También sería conveniente describir uno o varios puntos extremo indexados para el procesamiento de mensajes `<samlp:ArtifactResolve>`.

10.2.6.8 Ejemplo de un intercambio de mensajes utilizando Artifact HTTP

En este ejemplo, se intercambia un par de mensajes `<LogoutRequest>` y `<LogoutResponse>` mediante la vinculación Artifact HTTP, y la resolución del artefacto se lleva a cabo utilizando la vinculación SOAP dirigida a HTTP.

En primer lugar, a continuación se presentan los mensajes del protocolo del SAML reales que se han de intercambiar:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Esta vinculación no define la petición HTTP inicial del agente del usuario en el paso 1. El peticionario del SAML, para iniciar el intercambio de protocolos de fin de sesión, devuelve la siguiente respuesta HTTP, que contiene un artefacto del SAML. Los cambios de línea en el encabezamiento de Location HTTP siguiente son el resultado del formateo del documento, y en el valor de cabecera real no hay cambios de línea.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLart=AAQAADWNEw5VT47wcO4zX%
2FiEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU%3D&RelayState=0043bfc1bc45110dae170
04005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

A continuación, el respondedor del SAML resuelve el artefacto recibido en la petición del SAML mediante el protocolo de resolución de artefacto y la vinculación SOAP en los pasos 3 y 4, como sigue:

Paso 3:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: IdentityProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_6c3a4f8b9c2d" Version="2.0"
             IssueInstant="2004-01-21T19:00:49Z">
             <Issuer>https://ServiceProvider.com/SAML</Issuer>
             <Artifact>
             AAQAADWNEw5VT47wcO4zX/iEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU=
             </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Paso 4:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_FQvGknDfws2Z" Version="2.0"
             InResponseTo="_6c3a4f8b9c2d"
             IssueInstant="2004-01-21T19:00:49Z">
             <Issuer>https://IdentityProvider.com/SAML</Issuer>
             <samlp:Status>
                    <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
             </samlp:Status>
             <samlp:LogoutRequest ID="d2b7c388cec36fa7c39c28fd298644a8"
                    IssueInstant="2004-01-21T19:00:49Z"
                    Version="2.0">
                    <Issuer>https://IdentityProvider.com/SAML</Issuer>
                    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
                    <samlp:SessionIndex>1</samlp:SessionIndex>
             </samlp:LogoutRequest>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Una vez llevado a cabo un número no especificado de interacciones, el respondedor del SAML devuelve un segundo artefacto del SAML en su respuesta HTTP en el paso 6:

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:05:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLart=AAQAAFGIZXv5%2BQaBaE
5qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc%3D&RelayState=0043bfc1bc45110dae1
7004005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

A continuación el respondedor del SAML resuelve el artefacto que recibió en la petición del SAML en curso mediante el protocolo de resolución de artefacto y la vinculación SOAP en los pasos 7 y 8, como sigue:

Paso 7:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: ServiceProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_ec36fa7c39" Version="2.0"
             IssueInstant="2004-01-21T19:05:49Z">
             <Issuer>https://IdentityProvider.com/SAML</Issuer>
             <Artifact>
             AAQAAFGIZXv5+QaBaE5qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc=
             </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Paso 8:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:05:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_FQvGknDfws2Z" Version="2.0"
             InResponseTo="_ec36fa7c39"
             IssueInstant="2004-01-21T19:05:49Z">
             <Issuer>https://ServiceProvider.com/SAML</Issuer>
             <samlp:Status>
                    <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
             </samlp:Status>
             <samlp:LogoutResponse ID="_b0730d21b628110d8b7e004005b13a2b"
                    InResponseTo="_d2b7c388cec36fa7c39c28fd298644a8"
                    IssueInstant="2004-01-21T19:05:49Z"
                    Version="2.0">
                    <Issuer>https://ServiceProvider.com/SAML</Issuer>
                    <samlp:Status>
                           <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
                    </samlp:Status>
             </samlp:LogoutResponse>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.7 Vinculación URI SAML

Los URI constituyen medios independientes del protocolo para hacer referencia a un recurso. Esta vinculación no es del tipo general petición/respuesta del SAML, sino que soporta la encapsulación de un mensaje `<samlp:AssertionIDRequest>` con una sola `<saml:AssertionIDRef>` en la resolución de un URI. El resultado de una petición satisfactoria es un elemento `<saml:Assertion>` del SAML (pero no una respuesta del SAML completa).

Una resolución de URI puede ocurrir, como en el caso de SOAP, a través de múltiples transportes subyacentes. Esta vinculación posee aspectos independientes del transporte, pero también solicita la ayuda de HTTP con TLS 1.0 conforme se requiera (la implementación es obligatoria).

NOTA (informativa) – En PE24 (véase OASIS PE:2006) se sugiere sustituir el párrafo anterior por:

Una resolución de URI puede ocurrir, como en el caso de SOAP, a través de múltiples transportes subyacentes. Esta vinculación posee aspectos independientes del protocolo, pero también solicita la ayuda obligatoria de los URI de HTTP.

10.2.7.1 Información obligatoria

Identificación: `urn:oasis:names:tc:SAML:2.0:bindings:URI`

Información de la persona encargada: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna

10.2.7.2 Aspectos independientes del protocolo de la vinculación URI del SAML

En las siguientes subcláusulas se definen aspectos de la vinculación URI del SAML que son independientes del protocolo de transporte subyacente del proceso de resolución del URI.

Una referencia de URI del SAML identifica un aserción del SAML específica. El resultado de la resolución del URI debe arrojar un mensaje que contiene la aserción o un error específico de transporte. El formato específico del mensaje depende del protocolo de transporte subyacente. Si éste permite que se describa el contenido devuelto, tal como HTTP 1.1, la aserción podrá codificarse en cualquier formato permitido. De no ser así, la aserción tendrá que ser devuelta en un formato que puede interpretarse o transformarse de modo inequívoco en una serialización XML de la aserción.

La situación debe ser tal, que si en el futuro se resuelve la misma referencia de URI, se devolverá la misma aserción del SAML o bien un error. Es decir, la referencia puede ser persistente pero debe hacer referencia consistentemente a la misma aserción, si la hubiere.

10.2.7.3 Consideraciones de seguridad

El empleo indirecto de una aserción del SAML presenta riesgos si la vinculación de la referencia al resultado no es segura. Las amenazas particulares y su gravedad dependen del uso destinado para la aserción. En general, el resultado de la resolución entre una referencia de URI y una aserción del SAML sólo debería ser fiable si el peticionario puede estar seguro de la identidad del respondedor y de que el contenido no ha sido modificado durante su transporte.

A menudo no es suficiente que la propia aserción esté firmada, ya que las referencias de URI son por naturaleza un tanto opacas al peticionario. El peticionario debería disponer de medios independientes para asegurarse de que la aserción devuelta es en realidad la que está representada por el URI; esto se logra autenticando el respondedor y confiando en la integridad de la respuesta.

10.2.7.4 Encapsulación en MIME

En el caso de los protocolos de resolución que aceptan MIME como un mecanismo de descripción de contenido y de empaquetamiento, la aserción resultante debería ser devuelta como una entidad MIME del tipo `application/samlassertion+xml`, tal como se define en el apéndice II.

10.2.7.5 Utilización de URI de HTTP

Una autoridad que alegue conformidad con la vinculación URI del SAML está obligada a implementar el soporte de HTTP. En esta subcláusula se describen ciertos datos específicos acerca de la utilización de los URI de HTTP, incluyendo la sintaxis del URI, los encabezamientos de HTTP y la notificación de errores.

10.2.7.5.1 Sintaxis del URI

En general, no hay restricciones en cuanto a la sintaxis autorizada para una referencia de URI del SAML siempre que la autoridad del SAML responsable de la referencia haya creado el mensaje que la contiene. Sin embargo, las autoridades deben soportar un punto extremo de URL al que pueda ser enviada una petición HTTP con un solo parámetro de cadena de consulta denominado `ID`. En el propio URL de punto extremo no debe haber una cadena de consulta que sea independiente de este parámetro.

Por ejemplo, si el punto extremo documentado en una autoridad es "https://saml.example.edu/assertions", una petición de una aserción con un ID abcde puede enviarse a:

```
https://saml.example.edu/assertions?ID=abcde
```

Para este tipo de consultas de ID está prohibido el uso de comodines (*wildcards*).

NOTA (informativa) – En PE31 (véase OASIS PE:2006) se sugiere sustituir el texto anterior por:

Obsérvese que la sintaxis del URI no soporta el uso de comodines (*wildcards*) en dichas consultas.

10.2.7.5.2 HTTP y consideraciones relativas al almacenamiento

Los mandatarios HTTP no deben almacenar las aserciones del SAML. Para garantizarlo, se deberían seguir las reglas a continuación.

Cuando se devuelven aserciones del SAML del SAML mediante HTTP 1.1, los respondedores HTTP deberían:

- Incluir un campo de encabezamiento `Cache-Control` fijado a "`no-cache`, `no-store`".
- Incluir un campo de encabezamiento `Pragma` fijado a "`no-cache`".

10.2.7.5.3 Consideraciones de seguridad

En RFC 2617 del IETF se describen algunos de los posibles ataques al entorno de HTTP cuando se emplean esquemas de autenticación básicos o de resumen de mensajes.

Se recomienda firmemente la utilización de TLS 1.0 como un medio de autenticación, protección de la integridad y confidencialidad.

10.2.7.5.4 Notificación de error

Como intercambio de protocolos de HTTP, se debería emplear el código de situación HTTP adecuado para indicar el resultado de una petición. Por ejemplo, si un respondedor del SAML se niega a realizar un intercambio de mensajes con un peticionario del SAML, debería devolver una respuesta "`403 Forbidden`" (`403 prohibido`). Si el respondedor no conoce la aserción especificada, debería devolverse una respuesta "`404 Not Found`" (`404 no encontrada`). En estos casos, el contenido del cuerpo HTTP no es significativo.

10.2.7.5.5 Consideraciones relativas a los metadatos

El soporte de la vinculación URI por HTTP debería reflejarse indicando un punto extremo del URL al que se han de enviar las peticiones de aserciones arbitrarias.

10.2.7.5.6 Ejemplo de intercambio de mensajes mediante un URI de HTTP

A continuación se presenta un ejemplo de petición de aserción.

```
GET /SamlService?ID=abcde HTTP/1.1
Host: www.example.com
```

A continuación se presenta un ejemplo de la respuesta correspondiente, la cual suministra la aserción solicitada.

```
HTTP/1.1 200 OK
Content-Type: application/samlassertion+xml
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: nnnn

<saml:Assertion ID="abcde" ...>
...
</saml:Assertion>
```

11 Perfiles para el SAML

En esta cláusula se especifican los perfiles que definen la utilización de aserciones SAML y de mensajes de petición-respuesta en protocolos y marcos de comunicaciones, así como los perfiles que definen la sintaxis y los convenios de denominación de valores de atributos SAML.

11.1 Conceptos relativos al perfil

Un tipo de perfil SAML establece un conjunto de reglas que describen cómo se incluyen aserciones SAML en un marco o protocolo y cómo se extraen de ellos. En dichos perfiles se describe cómo una parte que origina puede incluir aserciones SAML en otros objetos o combinarlas con ellos (por ejemplo, ficheros de varios tipos o unidades de datos de protocolo de protocolos de comunicación), cómo puede comunicarlos a la parte que recibe y tramitarlos en su destino. El conjunto particular de reglas que rige la inserción de aserciones SAML y su extracción de una clase específica de objetos `<FOO>` se denomina *perfil <FOO>de SAML*.

Por ejemplo, un perfil SOAP de SAML describe cómo se pueden añadir aserciones SAML a mensajes SOAP, cómo dichas aserciones pueden afectar los encabezamientos SOAP, y cómo se pueden reflejar en mensajes SOAP los estados de error relacionados con el SAML.

Otro tipo de perfil SAML define un conjunto de restricciones que se imponen a la utilización de un protocolo general SAML o a la capacidad de aserción en determinado entorno de contexto o utilización. Asimismo, es posible que perfiles de este tipo requieran el empleo de funcionalidades específicas SAML (por ejemplo, atributos, condiciones o vínculos) y en otros términos definan las reglas de procesamiento que han de seguir quienes participan en los perfiles.

Un ejemplo de esto último lo constituyen quienes direccionan atributos SAML. El elemento `<Attribute>` SAML proporciona bastante flexibilidad a la hora de denominar el atributo, a su sintaxis de valor y a la inclusión de metadatos en banda, a través de la utilización de atributos XML. Se logra la interoperabilidad restringiendo dicha flexibilidad, siempre que se garantice ésta mediante la adhesión a perfiles que definan la utilización de dichos elementos con mayor precisión que las reglas genéricas definidas en la cláusula 8.

Los perfiles de atributo suministran las definiciones necesarias para restringir la expresión de atributo SAML cuando se trate de tipos particulares de información de atributo o cuando se esté interactuando con sistemas externos o normas abiertas que requieran mayor rigor.

En esta Recomendación se pretende especificar un conjunto escogido de perfiles de varios tipos con el detalle suficiente para garantizar que los productos que se implementen independientemente sean interoperables entre sí.

11.2 Especificación de perfiles adicionales

Si bien en esta Recomendación se define un conjunto seleccionado de perfiles, es posible que en el futuro se desarrollen otros más. En las cláusulas siguientes se presentan algunas directrices para la especificación de perfiles.

11.2.1 Directrices para la especificación de perfiles

En esta cláusula se presenta una lista de verificación de aspectos que han de tenerse en cuenta en cada perfil.

1) Especificación de un URI que identifique unívocamente el perfil, la información de contacto postal o electrónico del autor y proporcione referencia a perfiles definidos previamente que hayan sido actualizados o reemplazados por el actual.
2) Descripción del conjunto de interacciones entre las partes del perfil involucradas. Se deben incluir explícitamente todas las restricciones que existan sobre las aplicaciones utilizadas por cada parte y los protocolos que participan en cada interacción.
3) Identificación de las partes involucradas en cada interacción, incluyendo cuántas de ellas participan y si puede haber intermediarios.
4) Especificación del método de autenticación de las partes involucradas en cada interacción, indicando si dicha autenticación se requiere y cuáles son los tipos de autenticación aceptables.
5) Identificación del nivel de soporte de integridad de mensaje, incluido el mecanismo utilizado para garantizar dicha integridad.
6) Identificación del nivel de soporte de confidencialidad, especificando si una tercera parte puede ver los contenidos de mensajes y aserciones SAML, cuando el perfil requiera confidencialidad, y los mecanismos que se recomiendan para lograr dicha confidencialidad.
7) Identificación de los estados de error, incluidos aquéllos de cada participante, en particular los que reciben y procesan aserciones o mensajes SAML.

8) Identificación de las consideraciones de seguridad, incluido el análisis de las amenazas y la descripción de las medidas para hacerles frente.

9) Identificación de los identificadores de método de confirmación SAML que define y/o utiliza el perfil.

10) Identificación de los metadatos SAML pertinentes que define y/o utiliza el perfil.

11.2.2 Directrices para la especificación de perfiles de atributo

En esta cláusula se suministra una lista de verificación de aspectos que han de tenerse en cuenta en los perfiles de atributo.

1) Especificación de un URI que identifique unívocamente el perfil, la información de contacto postal o electrónico del autor y proporcione referencia a perfiles definidos previamente que hayan sido actualizados o reemplazados por el actual.

2) Sintaxis y restricciones de los valores aceptables de los atributos `NameFormat` y `Name` de los elementos `<Attribute>` SAML.

3) Todo atributo XML calificado en el espacio de nombres y definido por el perfil que pueda ser empleado en elementos `<Attribute>` SAML.

4) Reglas para establecer la igualdad de elementos `<Attribute>` SAML definidos por el perfil, que se empleen al procesar atributos, peticiones, etc.

5) Sintaxis y restricciones aplicables a los valores aceptables en el elemento `<AttributeValue>` SAML, indicando si se puede o si se debería utilizar el atributo XML `xsi:type`.

11.3 Identificadores de método de confirmación

En la cláusula 8 se define el elemento `<SubjectConfirmation>` como un `Método`, además de la `<SubjectConfirmationData>` opcional. La parte confiante debería emplear el elemento `<SubjectConfirmation>` para confirmar que la petición o el mensaje provienen de una entidad de sistema asociada con el sujeto de la aserción, dentro del contexto de un perfil particular.

El atributo `Method` indica el método específico que debería emplear la parte confiante para efectuar dicha determinación. Puede o no tener una relación con la autenticación efectuada anteriormente. A diferencia del contexto de autenticación, el método de confirmación de sujeto suele venir acompañado de información adicional, como por ejemplo un certificado o una clave, en el elemento `<SubjectConfirmationData>`, que permite a la parte confiante efectuar la verificación necesaria. Se define también un conjunto común de atributos que puede emplearse para restringir las condiciones bajo las cuales tiene lugar la verificación.

Se prevé que los perfiles definan y empleen varios valores para `<ConfirmationMethod>`, cada uno de los cuales corresponde a un caso diferente de utilización del SAML. Se definen los siguientes métodos con el fin de ser utilizados por los perfiles definidos en esta Recomendación y otros perfiles que puedan encontrarlos útiles.

11.3.1 Titular de clave

```
URI: urn:oasis:names:tc:SAML:2.0:cm:holder-of-key
```

Puede haber uno o varios elementos `<ds:KeyInfo>` dentro del elemento `<SubjectConfirmationData>`. También puede haber un atributo `xsi:type` en el elemento `<SubjectConfirmationData>` y, si lo hubiere, debe fijarse a **saml:KeyInfoConfirmationDataType** (el prefijo de espacio de nombre es arbitrario aunque debe hacer referencia al espacio de nombre de la aserción SAML).

Tal como se describe en la firma W3C, cada elemento `<ds:KeyInfo>` es titular de una clave o de información que permite a una aplicación obtener la clave. Se considera que el titular de determinada clave es el sujeto de la aserción efectuada por la parte asertante.

De conformidad con firma W3C, cada elemento `<ds:KeyInfo>` debe identificar una sola clave criptográfica. Es posible identificar varias claves utilizando elementos `<ds:KeyInfo>` separados, como por ejemplo cuando diferentes partes confiantes requieren diferentes claves de confirmación.

Ejemplo: El titular de la clave denominada "By-Tor" o el titular de la clave denominada "Snow Dog" puede confirmarse él mismo como el sujeto.

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
      <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
            <ds:KeyInfo>
                  <ds:KeyName>By-Tor</ds:KeyName>
            </ds:KeyInfo>
            <ds:KeyInfo>
                  <ds:KeyName>Snow Dog</ds:KeyName>
            </ds:KeyInfo>
      </SubjectConfirmationData>
</SubjectConfirmation>
```

11.3.2 Justificación de remitente

URI: urn:oasis:names:tc:SAML:2.0:cm:sender-vouches

Indica que no se dispone de ninguna otra información acerca del contexto de utilización de la aserción. La parte confiante debería emplear otros medios para establecer si es necesario procesarla más, sujeto a otras restricciones sobre la confirmación de utilización de los atributos que puedan estar presentes en el elemento `<SubjectConfirmationData>`.

11.3.3 Portador

URI: urn:oasis:names:tc:SAML:2.0:cm:bearer

El sujeto de la aserción es el portador de ella, con las restricciones opcionales a la confirmación que utiliza los atributos que puedan estar presentes en el elemento `<SubjectConfirmationData>`, conforme a la definición de la cláusula 8.

Ejemplo: El portador de la aserción puede confirmarse a sí mismo como el sujeto, siempre y cuando ésta se entregue en un mensaje enviado a "https://www.serviceprovider.com/saml/consumer" antes de la 1:37 PM GMT del 19 de marzo de 2004, como respuesta a la petición cuyo ID es "_1234567890".

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <SubjectConfirmationData InResponseTo="_1234567890"
            Recipient="https://www.serviceprovider.com/saml/consumer"
            NotOnOrAfter="2004-03-19T13:27:00Z"
      </SubjectConfirmationData>
</SubjectConfirmation>
```

11.4 Perfiles SSO del SAML

Se define un conjunto de perfiles con el fin de soportar el servicio de inscripción única (SSO, *single sign-on*) de los navegadores y otros dispositivos de clientes.

- Se define un perfil basado en el navegador web del protocolo de petición de autenticación en la cláusula 8, con el fin de soportar la inscripción única en la web.
- Se define un perfil adicional SSO en la web con el fin de soportar clientes ampliados.
- Se define un perfil de los protocolos de desconexión única (s*ingle logout*) y de identificador de nombre de la cláusula 8, tanto en el canal principal (navegador) como en los vínculos de canal secundario.
- Se define un perfil adicional para el descubrimiento de proveedor de identidad mediante los cookies.

11.4.1 Perfil SSO del navegador web

Cuando se trata de un perfil SSO de navegador web, el usuario de la web accede ya sea a un recurso con su proveedor de servicio o bien a un proveedor de identidad, de tal manera que el proveedor de servicio y el recurso deseado sean claros e implícitos. El usuario web se autentica (o ya se ha autenticado) con el proveedor de identidad, el cual a su vez produce una aserción de autenticación (probablemente utilizando información que proviene del proveedor de servicio) y el proveedor de servicio utiliza la aserción para establecer un contexto de seguridad para el usuario web. Durante este proceso, es posible establecer un identificador de nombre entre los proveedores para el principal, sujeto a los parámetros de la interacción y al consentimiento de las partes en cuestión.

Para implementar este caso, se utiliza un protocolo de petición de autenticación SAML, junto con los vínculos Redirect HTTP, POST HTTP y Artifact HTTP.

Se supone que el usuario utiliza un navegador comercial estándar y puede autenticarse ante el proveedor de identidad utilizando algunos medios que están fuera del alcance del SAML.

11.4.1.1 Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser`

Información de contacto: security-services-comment@lists.oasis-open.org

Identificadores de método de confirmación SAML: Este perfil utiliza el identificador de método de confirmación de "portador" V2.0 del SAML, `urn:oasis:names:tc:SAML:2.0:cm:bearer`.

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.1.2 Descripción general del perfil

En la figura 11-1 se muestra la plantilla básica para la utilización del SSO. El perfil describe los pasos siguientes. En un solo paso pueden existir uno o varios intercambios de mensajes reales dependiendo de la vinculación empleada para dicho paso y de otros comportamientos que dependen del tipo de implementación.

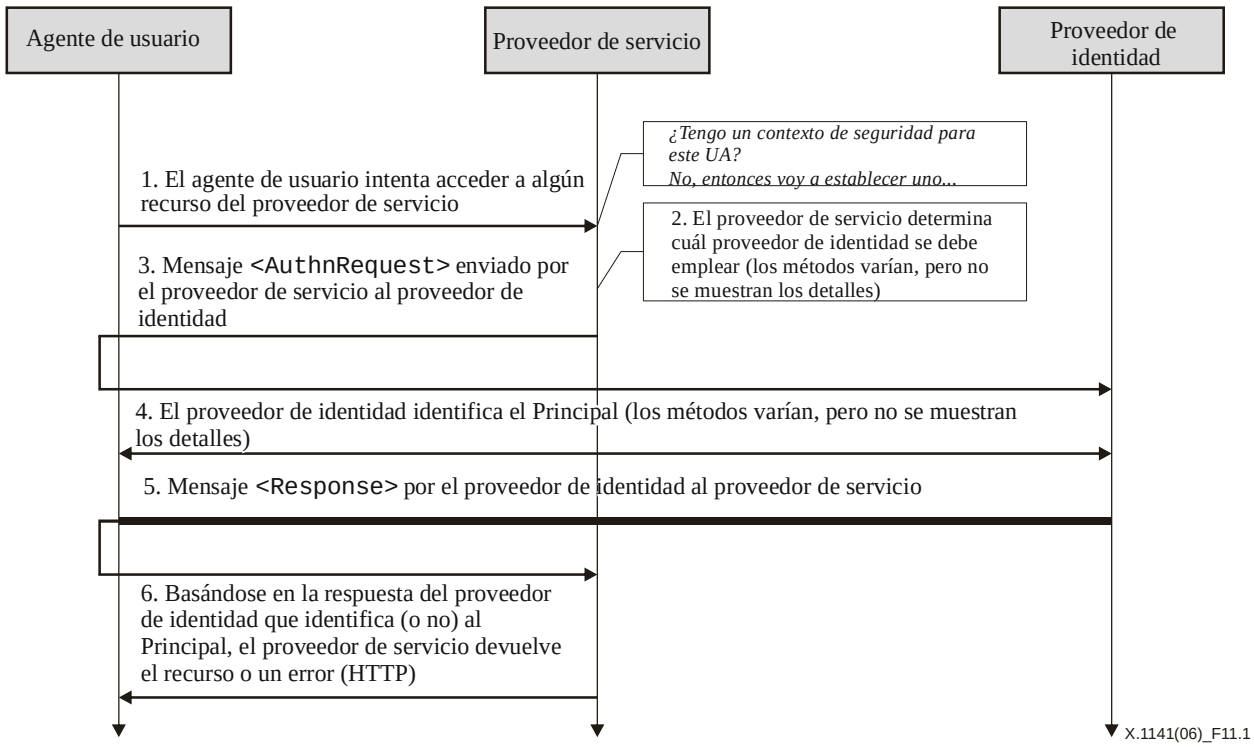

Figura 11-1/X.1141 – Plantilla básica para la utilización del SSO

1) Petición HTTP al proveedor de servicio

En el paso 1, el principal, a través de un agente de usuario HTTP, efectúa una petición HTTP al proveedor de servicio de un recurso asegurado sin que haya un contexto de seguridad.

2) El proveedor de servicio establece el proveedor de identidad

En el paso 2, el proveedor de servicio obtiene la ubicación de un punto extremo en un proveedor de identidad para el protocolo de petición de autenticación que soporta su vinculación preferida. Cómo se logra esto es algo que depende del tipo de implementación. El proveedor de servicio puede utilizar el perfil de descubrimiento de proveedor de identidad SAML que se describe en 8.7.4.

3) <AuthnRequest> enviado por el proveedor de servicio al de identidad

En el paso 3, el proveedor de servicio emite un mensaje `<AuthnRequest>` que el agente de usuario ha de entregar al proveedor de identidad. Se pueden utilizar los vínculos Redirect HTTP, POST HTTP o Artifact HTTP para transferir el mensaje al proveedor de identidad a través del agente de usuario.

4) El proveedor de identidad identifica al principal

En el paso 4, el proveedor de identidad identifica al principal utilizando medios que están fuera del alcance de este perfil. Es posible que sea necesaria una nueva autenticación o que se reutilice una sesión existente ya autenticada.

5) El proveedor de identidad envía una <Response> al proveedor de servicio

En el paso 5, el proveedor de identidad emite un mensaje `<Response>` que el agente de usuario ha de entregar al proveedor de servicio. Se puede utilizar la vinculación POST HTTP o la Artifact HTTP para transferir el mensaje al proveedor de servicio a través del agente de usuario. El mensaje puede indicar un error o incluir (por lo menos) una aserción de autenticación. No se debe utilizar la vinculación Redirect HTTP puesto que la respuesta en tal caso suele rebasar la longitud de URL permitida por la mayoría de los agentes de usuario.

6) El proveedor de servicio otorga o niega el acceso al principal

En el paso 6, tras haber recibido la respuesta del proveedor de identidad, el proveedor de servicio puede responder al agente del principal con su propio error o establecer su propio contexto de seguridad para él y devolver el recurso solicitado.

Un proveedor de identidad puede iniciar este perfil en el paso 5 y enviar un mensaje `<Response>` al proveedor de servicio sin necesidad de pasar por los pasos anteriores.

11.4.1.3 Descripción de perfil

Si el proveedor de servicio inicia el perfil, empiécese en la subcláusula 11.4.1.3.1. Si, en su lugar, lo inicia el proveedor de identidad, empiécese por la subcláusula 11.4.1.3.5. En las descripciones siguientes se hace referencia a:

Servicio de inscripción única

Éste es el punto extremo del protocolo de petición de autenticación en el proveedor de identidad al cual el agente de usuario entrega el mensaje `<AuthnRequest>` (o el artefacto que lo representa).

Servicio de consumidor de aserción

Éste es el punto extremo de protocolo de petición de autenticación en el proveedor de servicio al cual el agente de usuario entrega el mensaje `<Response>` (o el artefacto que lo representa).

11.4.1.3.1 Petición HTTP al proveedor de servicio

Si se accede en primer lugar al proveedor de servicio, es posible iniciar el perfil con una petición arbitraria de un recurso. No existen restricciones al formato de la petición. El proveedor de servicio es libre de utilizar cualquier medio para hacer corresponder las interacciones subsiguientes con la petición original. Cada una de las vinculaciones suministra un mecanismo RelayState que el proveedor de servicio puede emplear para hacer corresponder el intercambio de perfil con la petición original. Conviene que el proveedor de servicio restrinja al máximo la parte de la petición que queda al descubierto en el valor `RelayState`, salvo si la utilización del perfil no impone dichas medidas de privacidad.

11.4.1.3.2 El proveedor de servicio establece cuál es el proveedor de identidad

Este paso es una función del tipo de implementación que se tenga. El proveedor de servicio puede utilizar el perfil de descubrimiento de proveedor de identidad del SAML, que se describe en 11.4.3. Asimismo, puede decidir un reenvío del agente de usuario a otro servicio que esté en condiciones de establecer un proveedor de identidad adecuado. En este caso, el proveedor de servicio puede enviar una `<AuthnRequest>` (como en el próximo paso) a este servicio, que ha de ser reenviado al proveedor de identidad o que puede basarse en el servicio intermediario para emitir un mensaje `<AuthnRequest>` en su nombre.

11.4.1.3.3 El proveedor de servicio envía una <AuthnRequest> al proveedor de identidad

Una vez se haya escogido un proveedor de identidad, se establece la ubicación de su servicio de inscripción única, basándose en la vinculación SAML seleccionada por el proveedor de servicio para el envío del `<AuthnRequest>`. Es posible emplear metadatos a estos efectos. Como respuesta a una petición HTTP enviada por el agente de usuario, se devuelve una respuesta HTTP que contiene un mensaje `<AuthnRequest>` o un artefacto, dependiendo del tipo de vinculación SAML que se esté utilizando, y que se ha de entregar al servicio de inscripción única del proveedor de identidad.

El formato exacto que ha de seguir esta respuesta HTTP y la respuesta HTTP subsiguiente al servicio de inscripción única está definido por el tipo de vinculación SAML que se utiliza. En 11.4.1.4.1 se presentan reglas específicas de perfil para los contenidos del mensaje `<AuthnRequest>`. Si se utiliza la vinculación Redirect o POST HTTP, se entrega directamente el mensaje `<AuthnRequest>` al proveedor de identidad en este paso. Si se utiliza la vinculación Artifact HTTP, el proveedor de servicio emplea el perfil de resolución de artefacto que se define en 11.4.6, el cual devuelve la llamada al proveedor de servicio para obtener el mensaje `<AuthnRequest>`, utilizando, por ejemplo, la vinculación SOAP.

Se recomienda efectuar los intercambios HTTP en este paso a través del TLS 1.0, a fin de conservar la confidencialidad e integridad del mensaje. El mensaje <AuthnRequest> puede estar firmado, si se requiere autenticación de quién emite la petición. De haberla, la vinculación de artefacto también permite otras formas de autenticación del emisor de la petición cuando se deja de hacer referencia al artefacto.

El proveedor de identidad tiene que procesar el mensaje <AuthnRequest> conforme a lo descrito en la presente Recomendación. Esto puede afectar las interacciones subsiguientes con el agente de usuario, por ejemplo si se incluye el atributo IsPassive.

11.4.1.3.4 El proveedor de identidad identifica el principal

En cualquier instante del paso anterior o después de él, el proveedor de identidad debe establecer la identidad del principal (a menos que devuelva un error al proveedor de servicio). El atributo ForceAuthn <AuthnRequest>, si está presente y tiene un valor verdadero, obliga al proveedor de identidad a establecer la identidad reciente, en lugar de fiarse de una sesión existente que pueda tener con el principal. De lo contrario, en todos los demás casos, el proveedor de identidad puede emplear cualquier medio de autenticación del agente de usuario, sujeto a los requisitos incluidos en el <AuthnRequest> en la forma del elemento <RequestedAuthnContext>.

11.4.1.3.5 El proveedor de identidad envía <Response> al proveedor de servicio

Independientemente del éxito o fracaso del <AuthnRequest>, el proveedor de identidad debería enviar una respuesta HTTP al agente de usuario, que contenga un mensaje <Response> o un artefacto, dependiendo del tipo de vinculación SAML que se utilice, y que ha de ser entregado al servicio de consumidor de aserción del proveedor de servicio.

El formato exacto de esta respuesta HTTP y de la petición subsiguiente HTTP al servicio de consumidor de aserción va definido por el tipo de vinculación SAML que se esté empleando. En 11.4.1.4.2 se incluyen reglas específicas de perfil para los contenidos de <Response>. Si se está empleando la vinculación POST HTTP, el mensaje <Response> se entrega directamente al proveedor de servicio en este paso. Si se utiliza la vinculación Artifact HTTP, el proveedor de servicio emplea el perfil de resolución de artefacto definido en 11.4.6, que efectúa una llamada al proveedor de servicio para recuperar el mensaje <Response>, utilizando por ejemplo la vinculación SOAP.

Se puede encontrar la ubicación del servicio de consumidor de aserción utilizando metadatos. El proveedor de identidad ha de disponer de medios suficientes para establecer que dicha ubicación está controlada, de hecho, por el proveedor de servicio. Es posible que un proveedor de servicio indique la vinculación SAML y el servicio de consumidor de aserción que se emplea en su <AuthnRequest>, en cuyo caso el proveedor de identidad ha de emplearlo siempre que pueda.

Se recomienda que la petición HTTP de este paso de haga utilizando el TLS 1.0, con el fin de conservar la confidencialidad e integridad del mensaje. El elemento o los elementos <Assertion> en la <Response> deben ir firmados, si se utiliza la vinculación POST HTTP, y es posible que estén firmados en el caso de la vinculación Artifact HTTP.

El proveedor de servicio tiene que procesar el mensaje <Response> y cualquier elemento <Assertion> incluido, conforme a lo descrito en esta Recomendación.

11.4.1.3.6 El proveedor de servicio otorga o niega el acceso al agente de usuario

Para completar el perfil, el proveedor de servicio procesa la <Response> y la(s) <Assertion>, luego otorga o niega el acceso al recurso. El proveedor de servicio puede establecer un contexto de seguridad con el agente de usuario, a través de cualquier mecanismo de sesión. Todo uso ulterior de la(s) <Assertion> suministrada(s) es potestad del proveedor de servicio y de las otras partes confiantes, sujeto a cualquier restricción en el uso que esté incluida en ellas.

11.4.1.4 Utilización del protocolo de petición de autenticación

Este perfil se basa en el protocolo de petición de autenticación definido en esta Recomendación. En este caso, el proveedor de servicio es quien emite la petición y la parte confiante, mientras que el principal es quien presenta, el sujeto de la petición, y la entidad que confirma. Puede haber otras partes de transmisión o de confirmación siempre que el proveedor de identidad así lo desee.

11.4.1.4.1 Utilización de <AuthnRequest>

Un proveedor de servicio puede incluir cualquier contenido de mensaje descrito en la presente Recomendación. Todas las reglas de procesamiento han de ser conformes a esta Recomendación. El elemento <Issuer> ha de estar presente y debe contener el identificador único del proveedor de servicio que solicita; se debe ignorar el atributo Format o éste ha de tener un valor equivalente a urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Si el proveedor de identidad no puede o no podrá en un futuro satisfacer la petición, debe responder emitiendo un mensaje <Response> que contenga un código o códigos de estado de error adecuados.

Cuando el proveedor de servicio desee permitir al proveedor de identidad establecer un nuevo identificador del principal, de no haber ninguno, debe incluir un elemento <NameIDPolicy> en el que el atributo AllowCreate esté puesto a "verdadero". De lo contrario, sólo se podrá autenticar con éxito un principal para el cual el proveedor de identidad haya establecido previamente un identificador que pueda utilizar el proveedor de servicio.

El proveedor de servicio puede incluir un elemento <Subject> en la petición que establece la identidad real acerca de la cual desea recibir una aserción. Este elemento no podrá incluir ningún elemento <SubjectConfirmation>. Si el proveedor de identidad no reconoce que el principal corresponde a esta identidad, tiene que responder utilizando un mensaje <Response> que contenga un estado error y que no tenga ninguna aserción.

El mensaje <AuthnRequest> puede ir firmado (como si fuera dirigido por la vinculación SAML que se esté empleando). Si se utiliza la vinculación Artifact HTTP, la autenticación de las partes es facultativa y se puede utilizar cualquier mecanismo que sea permitido por la vinculación.

Si el <AuthnRequest> no está autenticado o no tiene protegida su integridad, no se podrá confiar en la información que contenga, que tendrá un carácter informativo. Independientemente de si se firma o no la petición, el proveedor de identidad ha de garantizar que se verifique la pertenencia de los elementos <AssertionConsumerServiceURL> o <AssertionConsumerServiceIndex> al proveedor de servicio a quien se ha enviado la respuesta. Si no se siguen cuidadosamente estas precauciones puede haber un ataque del tipo intromisión.

11.4.1.4.2 Utilización de <Response>

NOTA 1 (informativa) – PE26 (véase OASIS PE:2006) aclara el objetivo de esta cláusula. En el apéndice VIII se pueden encontrar más detalles al respecto.

Si el proveedor de identidad desea devolver un error, no puede incluir ninguna aserción en el mensaje <Response>. De lo contrario, si la petición es exitosa (o si la respuesta no está asociada con una petición), el elemento <Response> ha de cumplir con lo siguiente:

- Se puede prescindir del elemento <Issuer>, pero si lo hubiere tendría que incluir el identificador único del proveedor que emite la identidad; se puede ignorar el atributo Format u otorgarle un valor igual a urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

  NOTA 2 (informativa) – PE17 (véase OASIS PE:2006) sugiere reemplazar el párrafo anterior por el siguiente:

  Si el mensaje <Response> va firmado o si se cripta una aserción incluida, el elemento <Issuer> deberá entonces estar presente. De lo contrario, podría ser ignorado. Si lo hubiere, tendría que incluir el identificador único del proveedor que emite la identidad; se debe ignorar el atributo Format u otorgarle un valor de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- Debe incluir por lo menos una <Assertion>. Cada elemento <Issuer> de la aserción debe contener el identificador único del proveedor que emite la identidad; se debe ignorar el atributo Format u otorgarle un valor de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- El conjunto de una o varias aserciones debe contener por lo menos un <AuthnStatement> que refleje la autenticación del principal con el proveedor de identidad.
- Por lo menos una aserción que contenga un <AuthnStatement> ha de incluir un elemento <Subject> que tenga como mínimo un elemento <SubjectConfirmation> en el cual exista un Method urn:oasis:names:tc:SAML:2.0:cm:bearer. Si el proveedor de identidad soporta el perfil de desinscripción única, que se define en 11.4.4, toda declaración de autenticación de de este tipo tendrá que incluir un atributo SessionIndex que permita al proveedor de servicio emitir peticiones de desinscripción, sesión por sesión.
- El elemento <SubjectConfirmation> portador descrito anteriormente debe contener un elemento <SubjectConfirmationData> que a su vez contenga un atributo Recipient, que incluya el URL del servicio de consumidor de aserción del proveedor de servicio, y un atributo NotOnOrAfter que limite el intervalo de tiempo durante el cual se puede entregar la aserción. Puede también incluir un atributo Address que imponga restricciones a la dirección de cliente a la que se debe entregar la aserción. No podrá, en ningún caso, incluir un atributo NotBefore. Si el mensaje que lo contiene está en la respuesta a un <AuthnRequest>, el atributo InResponseTo debe corresponder con el ID de la petición.
- El proveedor de identidad tiene la potestad de incluir otras declaraciones y otros métodos de confirmación en la aserción o las aserciones. En particular, se pueden utilizar elementos <AttributeStatement>. Es posible que el <AuthnRequest> contenga un atributo XML AttributeConsumingServiceIndex que haga referencia a información acerca de los atributos deseados o requeridos, conforme a la cláusula 9. El proveedor de identidad puede hacer caso omiso de esto, o enviar otros atributos si así lo quisiere.

- La aserción o las aserciones que contienen una confirmación de sujeto portador pueden incluir un <AudienceRestriction> cuyo identificador único de proveedor de servicios sea equivalente al de <Audience>.
- De ser solicitado por el proveedor de servicio o si el proveedor de identidad así lo decidiere, se podrán incluir otras condiciones (y otros elementos <Audience>). (Por supuesto, el proveedor de servicio ha de entender y aceptar todas estas condiciones con el fin que de que se considere válida la aserción.) No es obligación del proveedor de identidad cumplir con el conjunto de <Conditions> en el <AuthnRequest>, si lo hubiere.

11.4.1.4.3 Reglas de procesamiento del mensaje <Response>

NOTA (informativa) – PE26 (véase OASIS PE:2006) aclara el objetivo de esta subcláusula. En el apéndice VIII se puede encontrar más información al respecto.

Independientemente del tipo de vinculación SAML empleado, el proveedor de servicio debe:

- Verificar todas las firmas presentes en la aserción o aserciones, o en la respuesta.
- Verificar que el atributo Recipient en cualquier <SubjectConfirmationData> portador corresponda al URL del servicio de consumidor de aserción al cual se entregó la <Response> o el artefacto.
- Verificar que no ha pasado el atributo NotOnOrAfter en ningún <SubjectConfirmationData> portador, sujeto a que se permita un sesgo de temporización entre los proveedores.
- Verificar que el atributo InResponseTo en el <SubjectConfirmationData> portador sea igual al ID de su mensaje <AuthnRequest> original, a menos que la respuesta no haya sido solicitada, en cuyo caso no debería estar presente dicho atributo.
- Verificar que todas las aserciones en las que se confíe sean válidas en otros casos.
- Si un <SubjectConfirmationData> portador incluye un atributo Address, el proveedor de servicio puede comparar la dirección del cliente del agente de usuario con dicho atributo.
- Toda aserción que no sea válida, o cuyos requisitos de confirmación de sujeto no puedan cumplirse, debería ser descartada y no debería emplearse para establecer un contexto de seguridad para el principal.
- Si un <AuthnStatement> que se utilice para establecer un contexto de seguridad para el principal contiene un atributo SessionNotOnOrAfter, se debería descartar el contexto de seguridad una vez se cumpla este tiempo, a menos que el proveedor de servicio restablezca la identidad del principal utilizando de nuevo este perfil.

11.4.1.4.4 Reglas de procesamiento del mensaje <Response> específicas del artefacto

Si se utiliza la vinculación Artifact HTTP para entregar la <Response>, la anulación de la referencia del artefacto que utiliza el perfil de resolución de artefacto debe ser autenticada mutuamente, protegida en su integridad y además confidencial.

El proveedor de identidad tiene que garantizar que sólo el proveedor de servicio a quien se haya enviado el mensaje <Response> recibirá dicho mensaje como resultado de una petición <ArtifactResolve>.

Cualquiera de los vínculos SAML que se utilice para anular la referencia del artefacto o firmas de mensaje se puede emplear para autenticar las partes y proteger los mensajes.

11.4.1.4.5 Reglas de procesamiento específicas al POST

NOTA (informativa) – PE26 (véase OASIS PE:2006) aclara el objetivo de esta subcláusula. Véase el apéndice VIII para más información al respecto.

Si se utiliza la vinculación POST HTTP para entregar la <Response>, la aserción o las aserciones incluidas deben ir firmadas.

El proveedor de servicio tiene que garantizar que no se reproduzcan las aserciones de portador, a través del mantenimiento del conjunto de los valores de ID utilizados el periodo del tiempo durante el cual se pueda considerar que la aserción es válida basándose en el atributo NotOnOrAfter en el <SubjectConfirmationData>.

11.4.1.5 Respuestas no solicitadas

Un proveedor de identidad puede iniciar este perfil mediante la entrega de un mensaje <Response> no solicitado a un proveedor de servicio.

Una <Response> no solicitada no puede incluir un atributo InResponseTo, ni debería tener ningún elemento <SubjectConfirmationData> de portador. Cuando se empleen metadatos, la <Response> o el artefacto deberían entregarse en el punto extremo <md:AssertionConsumerService> del proveedor de servicio que haya sido designado por defecto.

Cabe indicar que el proveedor de identidad puede incluir un parámetro "RelayState" específico de la vinculación que indique, sobre la base de acuerdos mutuos con el proveedor de servicio, cómo tratar interacciones posteriores con el agente de usuario. Puede tratarse del URL de un recurso en el proveedor de servicios. El proveedor de servicio debería estar preparado para procesar respuestas no solicitadas, atribuyendo una ubicación por defecto a la que se ha de enviar al agente de usuario tras el procesamiento exitoso de una respuesta.

11.4.1.6 Utilización de metadatos

En 11.4.2.5 se define un elemento de punto extremo, <md:SingleSignOnService>, que sirve para describir las vinculaciones y ubicación(es) soportadas, a las cuales el proveedor de servicio puede enviar peticiones a un proveedor de identidad que utilice este perfil.

Un proveedor de identidad puede utilizar el atributo WantAuthnRequestsSigned del elemento <md:IDPSSODescriptor> para documentar un requisito al que han de adherir las propuestas. También puede emplear el atributo AuthnRequestsSigned del <md:SPSSODescriptor> para documentar la intención de firmar todas sus peticiones.

Los proveedores pueden documentar la clave o claves empleadas para firmar las peticiones, respuestas y aserciones mediante los elementos <md:KeyDescriptor>, con el empleo de un atributo de firma. Al criptar elementos SAML, se pueden utilizar elementos <md:KeyDescriptor> que empleen un atributo de criptación, para documentar los algoritmos y configuración de criptación soportados, y las claves públicas utilizadas para recibir claves generales de criptación.

El elemento de punto extremo indexado <md:AssertionConsumerService> sirve para describir las vinculaciones y ubicaciones que se soportan, a las cuales puede enviar un proveedor de identidad respuestas a un proveedor de servicio, empleando este perfil. El atributo index sirve para distinguir los puntos extremos posibles que se pueden especificar haciendo referencia a ellos en el mensaje <AuthnRequest>. El atributo isDefault indica el punto extremo que se debe emplear cuando no haya sido especificado en una petición.

Un proveedor de servicio puede utilizar el atributo WantAssertionsSigned del elemento <md:SPSSODescriptor> para documentar un requisito consistente en que las aserciones entregadas con ese perfil han de ir firmadas. A esto se añade cualquier otro requisito impuesto a la firma por la utilización de una determinada vinculación. Si bien lo anterior no es obligatorio para el proveedor de identidad, se le hace notar que es probable que no baste con una aserción sin firma.

Si se entrega el mensaje de petición o de respuesta utilizando la vinculación Artifact HTTP, quien emite el artefacto debe proporcionar en sus metadatos por lo menos un elemento de punto extremo <md:ArtifactResolutionService>.

El <md:IDPSSODescriptor> puede contener los elementos <md:NameIDFormat>, <md:AttributeProfile> y <saml:Attribute> con el fin de indicar la capacidad general para soportar formatos particulares de identificador de nombre, perfiles de atributo o atributos y valores específicos. La capacidad de soportar cualquiera de dichas características durante un determinado intercambio de autenticación depende de las políticas y es potestad del proveedor de identidad.

También se puede emplear el elemento <md:SPSSODescriptor> para documentar la necesidad o el deseo del proveedor de servicio de que se entreguen atributos SAML junto con la información de autenticación. Es potestad del proveedor de identidad el que se incluyan realmente estos atributos. Puede haber uno o varios elementos <md:AttributeConsumingService> en sus metadatos, cada uno de los cuales contiene un atributo index que permite distinguir los diferentes servicios que se pueden especificar mediante referencia en el mensaje <AuthnRequest>. El atributo isDefault sirve para especificar un conjunto por defecto de requisitos de atributo.

11.4.2 Perfil de cliente o mandatario mejorados (ECP)

Un *cliente o mandatario mejorado* (ECP, *enhanced client or proxy*) es una entidad de sistema que sabe cómo contactar al proveedor de identidad adecuado, tal vez de una manera que depende del contexto, y que además soporta la vinculación SOAP inversa (PAOS) (véase la cláusula 10).

Un ejemplo en el que se utilizaría este perfil es el siguiente: Un principal, que dispone de un ECP, lo emplea bien sea para acceder a un recurso en el proveedor de servicio o para acceder a un proveedor de identidad tal que el proveedor de servicio y el recurso deseado sean claros o implícitos. El principal se autentica (o ya se ha autenticado) con el proveedor de identidad, que a su vez produce una aserción de autenticación (probablemente utilizando información del proveedor

de servicio). El proveedor de servicio utiliza luego la aserción y establece entonces un contexto de seguridad para el principal. Durante este proceso, se debe también crear un identificador de nombre entre los proveedores para el principal, sujeto a los parámetros de interacción y al consentimiento de éste.

El perfil se basa en el protocolo de petición de autenticación SAML, junto con la vinculación PAOS.

NOTA – Los medios que emplee el principal para autenticarse con un proveedor de identidad están fuera del alcance del SAML.

11.4.2.1 Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp` (éste también es el espacio de nombres objetivo que se atribuye en el esquema de perfil ECP correspondiente del anexo A).

Información de contacto: security-services-comment@lists.oasis-open.org

Identificadores de método de confirmación SAML: En este perfil se utiliza el identificador de método de confirmación de "portador" V2.0 SAML, `urn:oasis:names:tc:SAML:2.0:cm:bearer`.

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.2.2 Descripción general del perfil

Como se dijo anteriormente, el perfil ECP especifica las interacciones entre clientes y mandatarios mejorados y los proveedores de servicios y de identidad. Es una aplicación específica del perfil SSO que se describe en 11.4.1. Si este perfil no indica lo contrario, y si no hay particularidades relativas a la utilización de vinculaciones basadas en el navegador, se han de seguir las reglas que se indican en 11.4.1.

Un ECP es un cliente o un mandatario que satisface las dos condiciones siguientes:

- Tiene, o sabe cómo obtener, información acerca del proveedor de identidad que desea emplear el principal asociado con el ECP, en el contexto de una interacción con un proveedor de servicio.

De esta manera el proveedor de servicio puede emitir una petición de autenticación al ECP sin que sea necesario conocer o descubrir el proveedor de identidad adecuado (de hecho, se omite el paso 2 del perfil SSO de 11.4.1).

- Puede utilizar una vinculación SOAP reversa (PAOS), como se describe aquí, para una petición y respuesta de autenticación.

Siendo así, el proveedor de servicio está en condiciones de obtener una aserción de autenticación a través de un ECP al que, de otra manera, no se podría dirigir directamente ni estaría continuamente disponible (es decir, está fuera del contexto de la interacción inmediata). Asimismo, se incrementan los beneficios del SOAP a la vez que se utiliza un patrón y un perfil de intercambio bien definido para garantizar la interoperabilidad. Se puede pensar en el ECP como un intermediario SOAP entre le proveedor de servicio y el proveedor de identidad.

Un *cliente mejorado* puede ser un navegador o algún otro agente de usuario que soporta la funcionalidad que se describe en este perfil. Un *mandatario mejorado* es un mandatario HTTP que emula un cliente mejorado. A menos que se indique lo contrario, todas las declaraciones que se refieren a clientes tratan sobre clientes y mandatarios mejorados.

Puesto que el cliente mejorado envía y recibe mensajes en el cuerpo de las peticiones y respuestas HTTP, no existen restricciones arbitrarias para el tamaño de los mensajes de protocolo.

Este perfil mejora la vinculación SOAP reversa (PAOS) (véase la cláusula 10). Quienes lo implementen han de seguir las reglas para las indicaciones HTTP del soporte PAOS que se especifican en dicha vinculación, además de aquéllas especificadas en el presente perfil. Este perfil emplea un bloque de encabezamiento SOAP PAOS transportado entre el respondedor HTTP y el ECP, pero no define el PAOS propiamente dicho. Este perfil define bloques de encabezamiento SOAP que acompañan las peticiones y respuestas SAML. Los bloques de encabezamiento pueden componerse de otros bloques de encabezamiento SOAP si fuere necesario, por ejemplo el bloque de encabezamiento de seguridad de mensaje SOAP que añade características de seguridad cuando sea necesario, como en el caso de una firma digital aplicada a la petición de autenticación.

Se utilizan dos conjuntos de bloques de encabezamiento SOAP de petición/respuesta, a saber los bloques de encabezamiento PAOS para la información PAOS genérica y los bloques de encabezamientos específicos del perfil ECP que sirven para transportar información particular de la funcionalidad de perfil ECP.

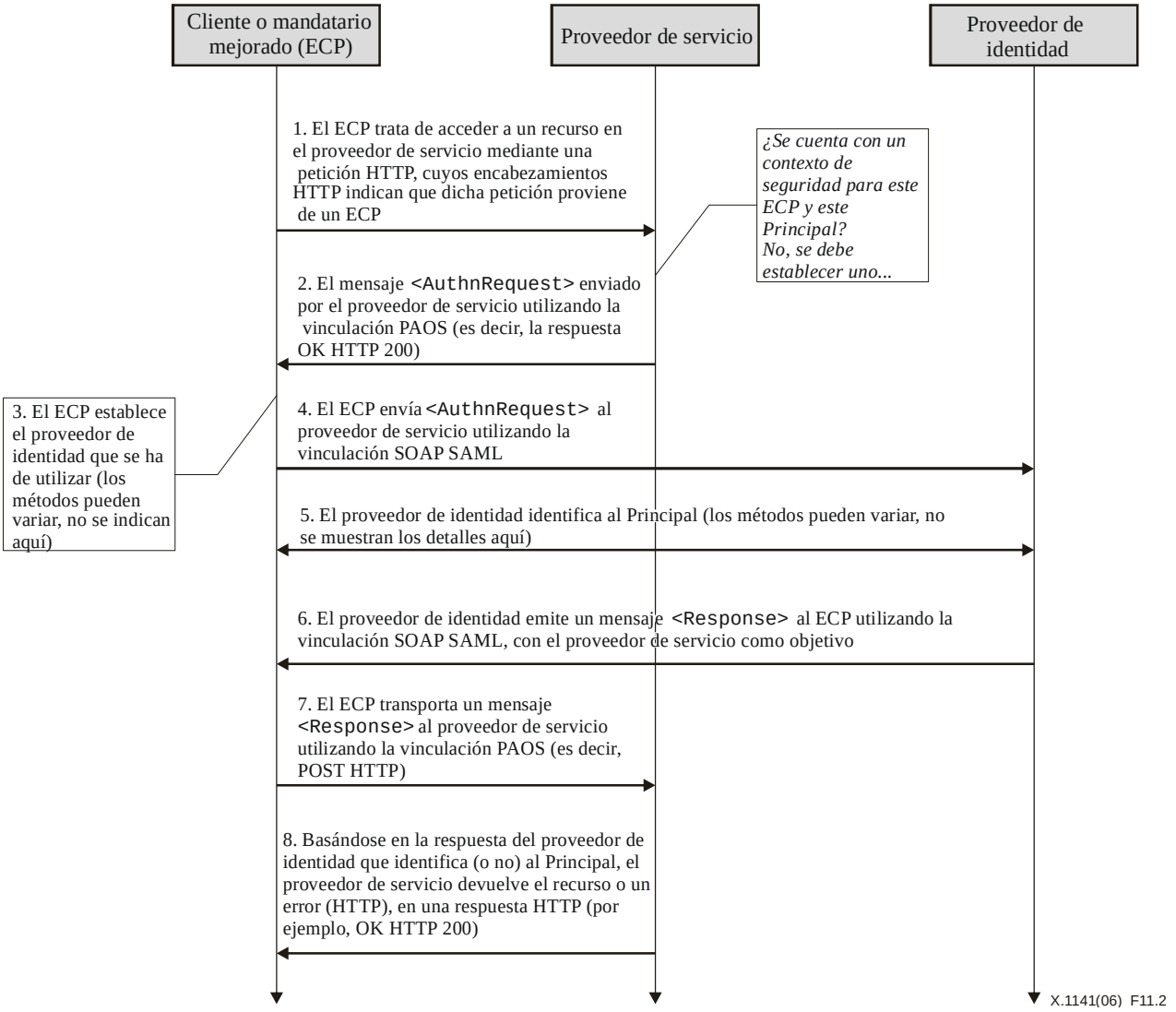

Figura 11-2/X.1141 – Flujo de procesamiento en el perfil ECP

La figura 11-2 ilustra la plantilla básica para el SSO que emplea un ECP. El perfil describe los pasos siguientes. Dentro de cada uno de ellos, puede haber uno o varios intercambios reales de mensajes dependiendo de la vinculación que se emplee para dicho paso y otros comportamientos que dependen de la implementación.

1) El ECP envía una petición HTTP al proveedor de servicios

En el paso 1, el principal, a través de un ECP, hace una petición HTTP de un recurso asegurado a un proveedor de servicio, en la que el proveedor de servicio no tiene un contexto de seguridad establecido para el ECP y el principal.

2) El proveedor de servicio envía <AuthnRequest> al ECP

En el paso 2, el proveedor de servicio envía un mensaje <AuthnRequest> al ECP, que debe ser entregado por este último al proveedor de identidad adecuado. En este caso se emplea la vinculación SOAP reversa (PAOS) (véase la cláusula 10).

3) El ECP establece quién es el proveedor de identidad

En el paso 3, el ECP obtiene la ubicación de un punto extremo en un proveedor de identidad a los efectos del protocolo de petición de autenticación que soporta su vinculación preferida. Cómo se logra esto es algo que depende del tipo de implementación utilizado. El ECP puede emplear el perfil de descubrimiento de proveedor de identidad SAML que se describe en 11.4.3.

NOTA (informativa) – PE18 (véase OASIS PE:2006) sugiere suprimir la última línea del párrafo anterior.

4) El ECP transporta <AuthnRequest> al proveedor de identidad

En el paso 4, el ECP transporta el <AuthnRequest> al proveedor de identidad identificado en el paso 3, utilizando una forma modificada de la vinculación SOAP SAML (véase la cláusula 10) con la permisión adicional para que el proveedor de identidad pueda intercambiar mensajes HTTP arbitrarios con el ECP antes de responder a la petición SAML.

5) El proveedor de identidad identifica al principal

En el paso 5, el proveedor de identidad identifica al principal utilizando medios que están fuera del alcance de este perfil. Es posible que se requiera un nuevo acto de autenticación o que se utilice una sesión existente ya autenticada.

6) El proveedor de identidad emite una <Response> al ECP dirigida al proveedor de servicio

En el paso 6, el proveedor de identidad emite un mensaje <Response>, utilizando la vinculación SOAP SAML, que ha de ser entregado por el ECP al proveedor de servicio. Este mensaje puede indicar un error o incluir (por lo menos) una aserción de autenticación.

7) El ECP transporta un mensaje <Response> hasta el proveedor de servicio

En el paso 7, el ECP transporta el mensaje <Response> hasta el proveedor de servicio utilizando la vinculación PAOS.

8) El proveedor de servicio otorga o niega el acceso al principal

En el paso 8, tras haber recibido el mensaje <Response> del proveedor de identidad, el proveedor de servicio establece su propio contexto de seguridad para el principal y devuelve el recurso solicitado, o bien responde al ECP del principal con un error.

11.4.2.3 Descripción de perfil

En las subcláusulas siguientes se proporcionan definiciones detalladas de cada uno de los pasos.

11.4.2.3.1 El ECP envía una petición HTTP al proveedor de servicio

El ECP envía una petición HTTP al proveedor de servicio, en la que se especifica algún recurso. Esta petición HTTP ha de ser conforme a la vinculación PAOS, es decir debe incluir los siguientes campos de encabezamiento HTTP:

1) El campo de encabezamiento `Accept` HTTP que indica la capacidad de aceptar el tipo MIME "`application/vnd.paos+xml`".
2) El campo de encabezamiento `PAOS` HTTP con, como mínimo, `urn:liberty:paos:2003-08`.
3) Más aún, en el campo de encabezamiento `PAOS` HTTP se debe especificar el soporte de este perfil como un valor de servicio, cuyo valor es `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`. Este valor debería corresponder al atributo de servicio en el bloque de encabezamiento SOAP de petición PAOS.

Por ejemplo, es posible que un agente de usuario solicite una página de un proveedor de servicio de la siguiente manera:

```
GET /index HTTP/1.1
Host: identity-service.example.com
Accept: text/html; application/vnd.paos+xml
PAOS: ver='urn:liberty:paos:2003-08' ;
'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

11.4.2.3.2 El proveedor de servicio envía <AuthnRequest> al ECP

Cuando el proveedor de servicio necesita un contexto de seguridad para el principal antes de permitir el acceso a un recurso especificado, es decir, antes de proveer un servicio o información, puede responder a la petición HTTP utilizando la vinculación PAOS con un mensaje <AuthnRequest> en la respuesta HTTP. El proveedor de servicio enviará una respuesta OK 200 HTTP al ECP que contenga un solo sobre SOAP.

El sobre SOAP ha de contener:

1) Un elemento <AuthnRequest> en el cuerpo del SOAP, destinado al recibiente último SOAP, el proveedor de identidad.
2) Un bloque de encabezamiento SOAP PAOS destinado al ECP utilizando el valor de actor SOAP `http://schemas.xmlsoap.org/soap/actor/next`. Este bloque de encabezamiento suministra información de control del tipo a cuál URL se va a enviar la respuesta en este patrón de intercambio de mensajes de solicitud de respuesta.

3) Un bloque de encabezamiento SOAP Request específico del perfil ECP, destinado al ECP que utiliza el actor SOAP `http://schemas.xmlsoap.org/soap/actor/next`. El bloque de encabezamiento Request ECP define información relacionada con la petición de autenticación que talvez necesite el ECP para procesarlo, por ejemplo una lista de proveedores de identidad que sean aceptables para el proveedor de servicio, si el ECP puede interactuar con el principal a través del cliente, y un nombre que pueda ser leído por una persona del proveedor de servicio, que probablemente se presente en una pantalla al principal.

El sobre SOAP puede incluir un bloque de encabezamiento SOAP RelayState ECP destinado al ECP, utilizando el valor de actor SOAP `http://schemas.xmlsoap.org/soap/actor/next`. El encabezamiento tiene información de estado que ha de ser devuelta por el ECP junto con la respuesta SAML.

11.4.2.3.3 El ECP encuentra el proveedor de identidad

El ECP establecerá cuál proveedor de identidad es adecuado y encaminará conforme a ello los mensajes SOAP.

11.4.2.3.4 El ECP envía <AuthnRequest> al proveedor de identidad

El ECP tiene que suprimir los bloques de encabezamiento PAOS, RelayState ECP y Request ECP antes de pasar el mensaje `<AuthnRequest>` al proveedor de identidad, utilizando una forma modificada de la vinculación SOAP SAML. Aunque la petición SAML se presenta de la manera usual en el SOAP, el proveedor de identidad puede responder a la petición HTTP del ECP con una respuesta HTTP que contenga, por ejemplo, una forma de registro HTML o alguna otra respuesta orientada a la presentación. Si bien puede haber una secuencia de intercambios HTTP, el proveedor de identidad final debe completar el intercambio SOAP SAML y devolver una respuesta SAML a través de la vinculación SOAP.

Es posible que el proveedor de servicio haya firmado él mismo el elemento `<AuthnRequest>`. En éste y otros casos, se han de seguir las reglas de mensaje especificadas en el perfil SSO de navegador de 11.4.1.4.1.

Antes o después de este paso, el proveedor de identidad tiene que establecer la identidad del principal utilizando algún medio, o debe devolver una `<Response>` de error, como se describe en 11.4.2.3.6.

11.4.2.3.5 El proveedor de identidad identifica el principal

En cualquier instante del paso anterior o como consecuencia de él, proveedor de identidad ha de establecer la identidad del principal (a menos que éste devuelva un error al proveedor de servicio). El atributo `ForceAuthn <AuthnRequest>`, si está presente y su valor es verdadero, obliga al proveedor de identidad a establecer recientemente esta entidad, en lugar de fiarse de una sesión existente que pudiese tener con el principal. De lo contrario, y en todos los demás casos, el proveedor de identidad puede emplear el medio que considere necesario para autenticar al agente de usuario, sujeto a cualquier requisito incluido en la `<AuthnRequest>` en la forma del elemento `<RequestedAuthnContext>`.

11.4.2.3.6 El proveedor de identidad envía una <Response> al ECP, destinada al proveedor de servicio

El proveedor de identidad devuelve un mensaje `<Response>` SAML (o un fallo SOAP) cuando se le presenta una petición de autenticación, tras haber establecido la identidad del principal. La respuesta SAML se transporta utilizando la vinculación SOAP SAML en un mensaje SOAP con un elemento `<Response>` en el cuerpo del SOAP, destinado al proveedor de servicio que es el receptor final del SOAP. Se deben seguir las reglas para la respuesta que se especifican en el perfil SSO de navegador de 11.4.1.4.2.

El mensaje de respuesta del proveedor de identidad ha de incluir un bloque de encabezamiento SOAP Response ECP específico del perfil, y puede contener un bloque de encabezamiento RelayState ECP, ambos dirigidos al ECP.

11.4.2.3.7 El ECP transporta el mensaje <Response> al proveedor de servicio

El ECP suprime el bloque o los bloques de encabezamiento, y puede añadir un bloque de encabezamiento SOAP Response PAOS y un bloque de encabezamiento RelayState ECP antes de reenviar la respuesta SOAP al proveedor de servicio, utilizando la vinculación PAOS.

Se suele emplear el bloque de encabezamiento SOAP `<paos:Response>` en la respuesta al proveedor de servicio para establecer una correlación entre esta respuesta y una petición anterior del proveedor de servicio. En este perfil, no se requiere el atributo `refToMessageID` de correlación, puesto que se puede utilizar a estos efectos el atributo `InResponseTo` de elemento `<Response>` SAML, aunque si el bloque de encabezamiento SOAP `<paos:Request>` tiene un `messageID`, se ha de utilizar el bloque de encabezamiento SOAP `<paos:Response>`.

El proveedor de servicio suele suministrar al ECP con su petición el valor de bloque de encabezamiento `<ecp:RelayState>`, aunque si el proveedor de identidad está produciendo una respuesta no solicitada (sin que se haya recibido la petición SAML del caso), se puede entonces incluir un bloque de encabezamiento RelayState que

indique, basándose en acuerdos mutuos con el proveedor de servicio, cómo tratar las interacciones subsiguientes con el ECP. Puede ser el URL de un recurso en el proveedor de servicio.

Si el proveedor de servicio incluye un bloque de encabezamiento SOAP `<ecp:RelayState>` en su petición al ECP, o si el proveedor de identidad pone un bloque de encabezamiento SOAP `<ecp:RelayState>` en su respuesta, el ECP debe entonces incluir un bloque de encabezamiento idéntico con la respuesta SAML que se envía al proveedor de servicio. El valor del proveedor de servicio para este bloque de encabezamiento (si lo hubiere) tiene prioridad.

11.4.2.3.8 El proveedor de servicio otorga o niega acceso al principal

Una vez el proveedor de servicio haya recibido la respuesta SAML en una petición HTTP (en un sobre SOAP que utiliza PAOS), puede responder con los datos de servicio en la respuesta HTTP. Al utilizar la respuesta, se deben seguir las reglas especificadas en el perfil SSO de navegador de 11.4.1.4.3 y 11.4.1.4.5. Es decir, las mismas reglas de procesamiento utilizadas al recibir la `<Response>` con la vinculación POST HTTP se aplican al PAOS.

11.4.2.4 Utilización del esquema de Perfil ECP

El esquema XML de Perfil ECP define los bloques de encabezamiento de petición/respuesta SOAP empleados por este perfil. A continuación se presenta un listado completo de este esquema.

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>

    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
```

```
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

En las subcláusulas siguientes se describe cómo han de emplearse estas construcciones XML.

11.4.2.4.1 Bloque de encabezamiento Request PAOS: SP a ECP

El bloque de encabezamiento Request PAOS indica la utilización del procesamiento PAOS e incluye los siguientes atributos:

– `responseConsumerURL` [Obligatorio]

  Especifica cuándo el ECP ha de enviar una respuesta de error. Sirve también para verificar si la respuesta del proveedor de identidad es correcta, comparando su ubicación con el `AssertionServiceConsumerURL` del bloque de encabezamiento de respuesta ECP. Este valor tiene que ser idéntico al `AssertionServiceConsumerURL` (o al URL que se hace referencia en los metadatos) transportado en el `<AuthnRequest>`.

  NOTA (informativa) – PE22 (véase OASIS PE:2006) sugiere que se cambie en la última frase `AssertionServiceConsumerURL` por `AssertionConsumerServiceURL`.

– `service` [Obligatorio]

  Indica el servicio PAOS que está siendo utilizado en el perfil de autenticación SAML. El valor debe ser `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`.

– `SOAP-ENV:mustUnderstand` [Obligatorio]

  El valor debe ser `1` (verdadero). Se debe generar un fallo SOAP si no se ha entendido el bloque de encabezamiento PAOS.

– `SOAP-ENV:actor` [Obligatorio]

  El valor debe ser `http://schemas.xmlsoap.org/soap/actor/next`.

– `messageID` [Facultativo]

  Permite la correlación opcional de respuesta. Se puede utilizar en este perfil aunque no es obligatorio, puesto que esta funcionalidad viene suministrada por la capa de protocolo SAML, a través del atributo `ID` en el `<AuthnRequest>` y del atributo `InResponseTo` en la `<Response>`.

El bloque de encabezamiento SOAP Request PAOS no tiene contenido de elementos.

11.4.2.4.2 Bloque de encabezamiento Request ECP: SP a ECP

El bloque de encabezamiento SOAP Request ECP sirve para transportar la información que necesita el ECP para procesar la petición de autenticación. Es obligatorio y su presencia indica que se utiliza este perfil. Contiene los siguientes elementos y atributos:

– `SOAP-ENV:mustUnderstand` [Obligatorio]

  El valor debe ser `1` (verdadero). Se debe generar un fallo SOAP si no se entiende el bloque de encabezamiento ECP.

– `SOAP-ENV:actor` [Obligatorio]

  El valor debe ser `http://schemas.xmlsoap.org/soap/actor/next`.

– `ProviderName` [Facultativo]

  Nombre legible por una persona del proveedor de servicio que emite la petición.

– `IsPassive` [Facultativo]

  Valor booleano, que cuando es `verdadero` indica que el proveedor de identidad y el cliente propiamente dicho no deben tomar control de la interfaz de usuario de quien emite la petición e interactuar con el principal de una manera que sea visible. Si no se proporciona un valor, se otorga por defecto el valor `verdadero`.

– `<saml:Issuer>` [Obligatorio]

  Este elemento debe contener el identificador único del proveedor de servicio que emite la petición; el atributo `Format` debe ignorarse o ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

– `<samlp:IDPList>` [Facultativo]

  Lista facultativa de proveedores de identidad que el proveedor de servicio reconoce y a partir de la cual el ECP puede tramitar la petición.

11.4.2.4.3 Bloque de encabezamiento RelayState ECP: SP a ECP

El bloque de encabezamiento SOAP RelayState ECP se emplea para transportar información de estado desde el proveedor de servicio, que la necesitará más adelante al procesar la respuesta del ECP. Es facultativo, pero si se le emplea, el ECP tiene que incluir un bloque de encabezamiento idéntico en la respuesta en el paso 5 en la figura 11-2. Contiene los siguientes atributos:

NOTA (informativa) – PE27 (véase OASIS PE:2006) sugiere que se reemplace en el texto anterior el paso 5 por el paso 7.

– `SOAP-ENV:mustUnderstand` [Obligatorio]

  El valor debe ser `1` (verdadero). Se debe generar un fallo SOAP si no se ha entendido el bloque de encabezamiento.

– `SOAP-ENV:actor` [Obligatorio]

  El valor debe ser `http://schemas.xmlsoap.org/soap/actor/next`.

El contenido del elemento bloque de encabezamiento es una cadena que incluye información de estado creada por quien emite la petición. De haberlo, el ECP debe incluir el mismo valor en un bloque de encabezamiento RelayState cuando responda al proveedor de servicio en el paso 5. El valor de cadena no puede rebasar 80 bytes en longitud y debería estar protegido en su integridad por el que emite la petición, sin importar que existan otras protecciones que puedan actuar o no durante la transmisión del mensaje.

A continuación se presenta un ejemplo de la petición de autenticación SOAP enviada por el proveedor de servicio al ECP:

```
<SOAP-ENV:Envelope
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Request xmlns:paos="urn:liberty:paos:2003-08"
       responseConsumerURL="http://identity-service.example.com/abc"
       messageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" SOAP-
ENV:mustUnderstand="1"
       service="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp">
    </paos:Request>
    <ecp:Request xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
       ProviderName="Service Provider X" IsPassive="0">
      <saml:Issuer>https://ServiceProvider.example.com</saml:Issuer>
      <samlp:IDPList>
        <samlp:IDPEntry ProviderID="https://IdentityProvider.example.com"
            Name="Identity Provider X"
            Loc="https://IdentityProvider.example.com/saml2/sso"
        </samlp:IDPEntry>
        <samlp:GetComplete>
        https://ServiceProvider.example.com/idplist?id=604be136-fe91-441e-
afb8
        </samlp:GetComplete>
      </samlp:IDPList>
    </ecp:Request>
    <ecp:RelayState
xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
```

```
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Como ya se indicó, el ECP suprime los bloques de encabezamiento PAOS y ECP del mensaje SOAP antes de que el proveedor de identidad reenvíe la petición de autenticación. A continuación se presenta un ejemplo de petición de autenticación enviada desde el ECP al proveedor de identidad:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.4.4 Bloque de encabezamiento Response ECP: IdP a ECP

Se debe utilizar el bloque de encabezamiento SOAP Response ECP en la respuesta del proveedor de identidad al ECP. Contiene los siguientes atributos:

– `SOAP-ENV:mustUnderstand` [Obligatorio]

  El valor debe ser `1` (verdadero). Se generará un fallo SOAP si no se entiende el bloque de encabezamiento ECP.

– `SOAP-ENV:actor` [Obligatorio]

  El valor debe ser `http://schemas.xmlsoap.org/soap/actor/next`.

– `AssertionConsumerServiceURL` [Obligatorio]

  Lo fija el proveedor de identidad basándose en el mensaje `<AuthnRequest>` o en los metadatos del proveedor de servicios que ha obtenido.

El ECP debe confirmar que este valor corresponde al que ha obtenido en la `responseConsumerURL` en el bloque de encabezamiento SOAP Request PAOS que recibió del proveedor de servicio. Puesto que el `responseConsumerURL` puede ser relativo mientras que el `AssertionConsumerServiceURL` es absoluto, tal vez sea necesario algún procesamiento o alguna normalización.

Este mecanismo se utiliza a los efectos de seguridad para confirmar el destino correcto de la respuesta. Si los valores no corresponden, el ECP tiene que generar una respuesta de fallo SOAP para el proveedor de servicio y no deberá devolver la respuesta SAML.

El encabezamiento SOAP Response ECP no contiene elementos.

El siguiente es un ejemplo de respuesta de IdP a ECP:

```
<SOAP-ENV:Envelope
      xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ecp:Response SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
AssertionConsumerServiceURL="https://ServiceProvider.example.com/ecp_assert
ion_consumer"/>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.4.5 Bloque de encabezamiento Response PAOS: ECP a SP

El bloque de encabezamiento Response PAOS incluye los siguientes atributos:

– `SOAP-ENV:mustUnderstand` [Obligatorio]

  El valor debe ser `1` (verdadero). Se debe generar un fallo SOAP si no se entiende el bloque de encabezamiento PAOS.

– `SOAP-ENV:actor` [Obligatorio]

  El valor debe ser `http://schemas.xmlsoap.org/soap/actor/next`.

– `refToMessageID` [Facultativo]

  Permite la correlación con la petición PAOS. El ECP debe añadir este atributo facultativo (y el bloque de encabezamiento como un todo) cuando la petición PAOS correspondiente especifique el atributo `messageID`. En el SAML se permite la funcionalidad equivalente mediante la correlación `<AuthnRequest>` y `<Response>`.

El encabezamiento SOAP Response PAOS no tiene contenido de elementos.

El siguiente es un ejemplo de respuesta de ECP a SP:

```
<SOAP-ENV:Envelope
      xmlns:paos="urn:liberty:paos:2003-08"
      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Response refToMessageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next/" SOAP-
ENV:mustUnderstand="1"/>
    <ecp:RelayState
xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
      SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.5 Consideraciones relativas a la seguridad

El mensaje `<AuthnRequest>` debería ir firmado. Conforme a las reglas especificadas por el perfil SSO de navegador, se deberían firmar las aserciones incluidas en la `<Response>`. La entrega de la respuesta en el sobre SOAP a través del PAOS es análoga en esencia al empleo de la vinculación POST HTTP y de las medidas adecuadas para corregir problemas de seguridad que se estén empleando con dicha vinculación.

Conviene proteger la integridad de los encabezamientos SOAP, empleando por ejemplo la seguridad de mensaje SOAP o mediante el TLS en cada intercambio HTTP con el cliente.

El proveedor de servicio debería autenticarse ante el ECP, empleando por ejemplo la autenticación TLS en el lado del servidor.

El ECP debería autenticarse ante el proveedor de identidad, de tal manera que se pueda sostener una sesión autenticada. Todo tipo de intercambios HTTP posteriores a la entrega del mensaje `<AuthnRequest>` y que tienen lugar antes de que el proveedor de identidad devuelva una `<Response>` se debe asociar de modo seguro con la petición original.

NOTA (informativa) – PE20 (véase OASIS PE:2006) sugiere que se añada el párrafo siguiente acerca de las consideraciones relativas a los metadatos ECP:

Las reglas especificadas en el perfil SSO del navegador de la cláusula 11 también se aplican en este caso. En particular, se puede utilizar el elemento de punto extremo indexado `<md:AssertionConsumerService>` cuya vinculación sea `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`, para describir la vinculación y las ubicaciones soportadas a las cuales el proveedor de identidad puede enviar respuestas a un proveedor de servicio que utilice este perfil. Asimismo, se puede emplear el punto extremo `<md:SingleSignOnService>` cuya vinculación sea `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`, para describir la vinculación y la ubicación o ubicaciones soportadas a las cuales el proveedor de servicio puede enviar peticiones a un proveedor de identidad que utiliza este perfil.

11.4.3 Perfil de descubrimiento de proveedor de identidad

En esta cláusula se define un perfil mediante el cual el proveedor de servicio puede descubrir cuáles son los proveedores de identidad que está empleando un principal mediante el perfil SSO de navegador web. En aquellas configuraciones en las que existe más de un proveedor de identidad, los proveedores de servicio requieren medios para descubrir cuál o cuáles proveedores de identidad emplea el principal. El perfil de descubrimiento se basa en una cookie escrita en un dominio común entre los proveedores de identidad y los proveedores de servicio en cada configuración. El dominio que predetermina la configuración se conoce como el dominio común en este perfil, y la cookie que contiene la lista de los proveedores de identidad se denomina cookie de dominio común.

Queda fuera de alcance de este perfil determinar cuáles entidades contienen servidores web en el dominio común.

NOTA (informativa) – PE32 (véase OASIS PE:2006) sugiere añadir lo siguiente para describir la información requerida:
Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:idp-discovery`
Información de contacto: security-services-comment@lists.oasis-open.org

11.4.3.1 Cookie de dominio común

El nombre de la cookie debe ser "`_saml_idp`". El formato de su valor tiene que ser un conjunto de uno o varios valores URI codificados en base 64, separados por el carácter espacio. Cada URI es el identificador único de un proveedor de identidad, conforme a la definición de la cláusula 7. El conjunto del final de valores va entonces codificado URL.

El servicio de escritura de cookie de dominio común debería añadir el identificador único del proveedor de identidad a la lista. Si el identificador ya se encuentra en dicha lista, puede suprimirlo y añadirlo de nuevo. Con ello, se busca que la sesión de proveedor de identidad establecida más recientemente sea la última en la lista.

Se debe crear una cookie con un prefijo de trayecto igual a "/". El dominio se debe poner a ".[common-domain]" en donde [common-domain] es el dominio común establecido dentro de la configuración que se utiliza con este perfil. Debe haber un espacio delante. Se debe marcar la cookie como segura.

Conviene que la sintaxis de la cookie sea conforme a RFC 2965 del IETF. Puede ser válida por una sesión o persistente. La selección depende del tipo de implementación, aunque debería ser uniforme para todos los proveedores de identidad de una configuración determinada.

11.4.3.2 Creación de la cookie de dominio común

Después de que el proveedor de identidad autentique el principal, se puede establecer la cookie de dominio común. Cómo se haga, es algo que depende de la implementación, siempre y cuando la cookie se cree conforme a los parámetros mencionados anteriormente. A continuación se presenta una posible estrategia de implementación que no debería considerarse como normativa. El proveedor de identidad puede:

- Haber establecido anteriormente un alias DNS y uno IP para sí mismo en el dominio común.
- Redirigir el agente de usuario hacia sí mismo utilizando el alias DNS, con un URL que especifique "https" como esquema URL. La estructura del URL es propia de la implementación y puede incluir información de sesión necesaria para identificar el agente de usuario.
- Crear la cookie en el agente de usuario al que se redirigió mediante los parámetros que se acaban de especificar.
- Redirigir el agente de usuario de nuevo hacia sí mismo, o, cuando sea necesario, hacia el proveedor de servicio.

11.4.3.3 Obtención de la cookie de dominio común

Cuando un proveedor de servicio necesita descubrir cuáles proveedores de identidad son empleados por un principal, invoca un intercambio destinado a presentar la cookie de dominio común al proveedor de servicio después de que haya sido leída por un servidor HTTP en el dominio común.

Si el proveedor de servicio se encarga del servidor HTTP en el dominio común o si se han establecido otros acuerdos al respecto, el proveedor de servicio puede utilizar dicho servidor en el dominio común para retransmitir su `<AuthnRequest>` al proveedor de identidad, con miras a un proceso optimizado de inscripción única.

Los métodos utilizados por el proveedor de servicio para leer la cookie dependen de la implementación, siempre y cuando el agente de usuario sea capaz de presentar cookies que hayan sido configuradas conforme a los parámetros de 11.4.3.1. A continuación se describe una posible estrategia que no debería considerarse como normativa. Además, puede ser subóptima en el caso de algunas aplicaciones. El proveedor puede:

- Haber establecido anteriormente un alias DNS y uno IP para sí mismo en el dominio común.
- Redirigir el agente de usuario hacia sí mismo utilizando el alias DNS, con un URL que especifique "https" como esquema URL. La estructura del URL es propia de la implementación y puede incluir información de sesión necesaria para identificar al agente de usuario.
- Redirigir el agente de usuario de nuevo hacia sí mismo, o, cuando sea necesario, hacia el proveedor de servicio.

11.4.4 Perfil de desinscripción única

Tras la autenticación de un principal ante un proveedor de identidad, la entidad que autentica puede establecer una sesión con el principal (que suele hacerse por medio de una cookie, la reescritura de un URL u otro método que dependa de la implementación). Luego, el proveedor de identidad puede enviar aserciones al proveedor de servicio o a

otras partes confiantes, basándose en este evento de autenticación; una parte confiante puede utilizarlo para establecer *su propia* sesión con el principal.

En tal caso, el proveedor de identidad puede actuar como autoridad de sesión y las partes confiantes como participantes en ella. Más tarde, es posible que el principal desee terminar su propia sesión bien sea con un participante único de sesión o con todos los participantes en determinada sesión que gestiona la autoridad de sesión. El primer caso está fuera del alcance de esta Recomendación. El segundo, no obstante, puede satisfacerse utilizando el perfil del protocolo de gestión única SAML (véase 11.4).

Un principal (o un administrador que termine una sesión de un principal) puede decidir que se termina esta sesión "global" ya sea contactando la autoridad de sesión o un participante en la sesión. De igual manera, un proveedor de identidad que funja como autoridad de sesión puede "*él mismo*" actuar como participante de sesión en situaciones en las cuales sea la parte confiante para otras aserciones de proveedor de identidad que tengan que ver con el principal.

El perfil permite que el protocolo se combine con una vinculación síncrona, como por ejemplo la vinculación SOAP, o con vinculaciones "de canal frontal" asíncronas, como por ejemplo la Redirect HTTP, POST o Artifact. Se puede requerir una vinculación de canal frontal cuando, por ejemplo, existe un estado de sesión de principal solamente en un agente de usuario, cuya forma es una cookie, y se requiere una interacción directa entre el agente de usuario y el participante o la autoridad de sesión. Como se explica a continuación, conviene que, en la medida de lo posible, los participantes de sesión empleen una vinculación "de canal de frontal" cuando inician este perfil, con el fin de maximizar la probabilidad de que la autoridad de gestión de sesión pueda propagar con éxito a todos los participantes la señal de desinscripción.

11.4.4.1 Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:logout`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.4.2 Descripción general del perfil

En la figura 11-3 se muestra la plantilla básica que corresponde a una desinscripción simple.

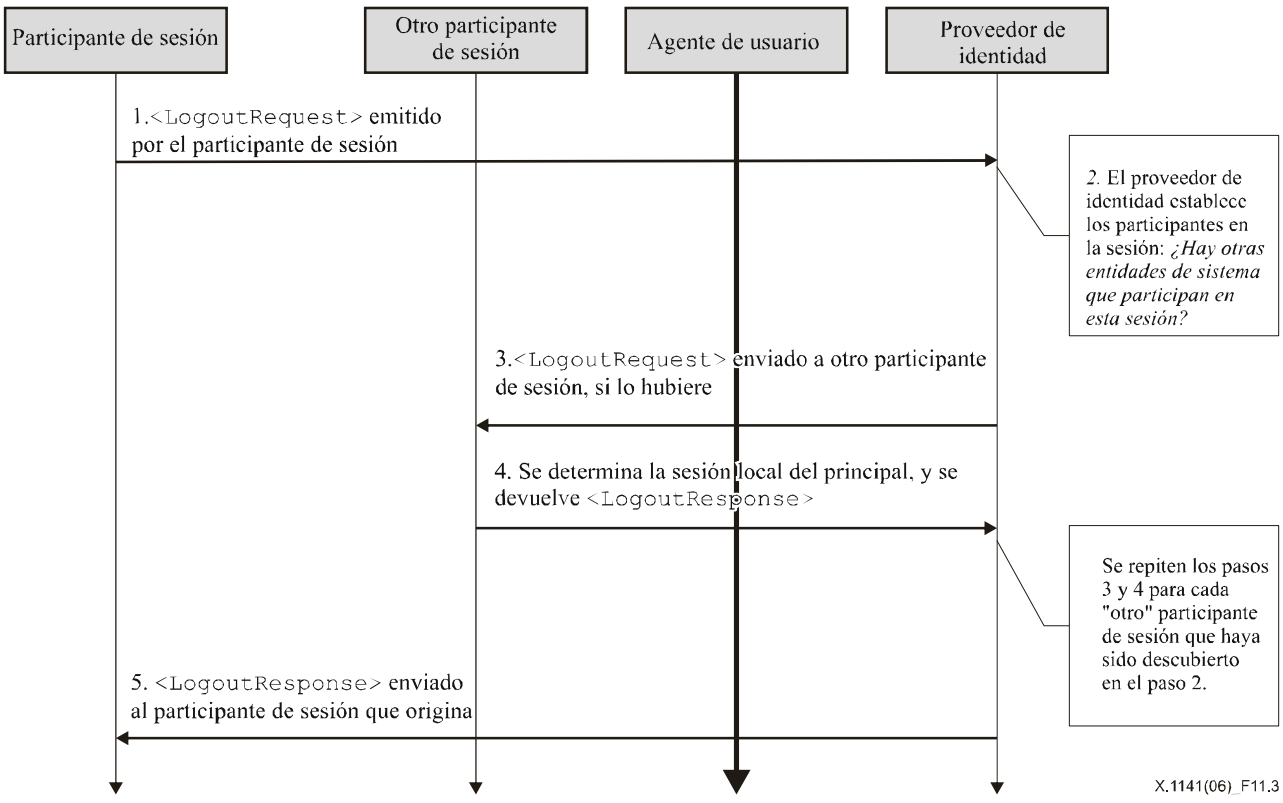

Figura 11-3/X.1141 – Plantilla de la desinscripción única

En la figura anterior se indica el agente de usuario utilizando un tipo de letra con un tono de gris más leve para indicar que el intercambio de mensajes puede pasar a través de él o puede ser un intercambio directo entre entidades de sistema, según el tipo de vinculación SAML que se use para implementar ese perfil.

El perfil describe los pasos siguientes. Dentro de cada paso, puede haber uno o varios intercambios de mensajes dependiendo de la vinculación que se utilice para dicho paso y de otros comportamientos característicos de dicha implementación.

1) <LogoutRequest> enviado por un participante de sesión a un proveedor de identidad

En el paso 1, el participante de sesión inicia la desinscripción única y termina una o varias sesiones de principal al enviar un mensaje `<LogoutRequest>` al proveedor de identidad del cual recibió la aserción correspondiente de autenticación. La petición se puede enviar directamente al proveedor de identidad o a través del agente de usuario.

2) El proveedor de identidad establece quiénes son los participantes en la sesión

En el paso 2, el proveedor de identidad emplea los contenidos del mensaje `<LogoutRequest>` (o en el caso de ser él mismo quien inicia la desinscripción, algún otro mecanismo) para establecer cuál es la sesión o las sesiones que se están terminando. De no haber otros participantes en la sesión, el perfil sigue con el paso 5. De lo contrario, se repiten los pasos 3 y 4 para cada participante de sesión que haya sido identificado.

3) <LogoutRequest> enviado por el proveedor de identidad al participante o autoridad de sesión

En el paso 3, el proveedor de identidad envía un mensaje `<LogoutRequest>` a un participante o una autoridad de sesión relacionado con una o varias de las sesiones que se están terminando. La petición se puede enviar directamente a la entidad o a través del agente de usuario (si esto último no implica ninguna contradicción con la forma de la petición del paso 1).

4) El participante o la autoridad de sesión envía <LogoutResponse> al proveedor de identidad

En el paso 4, un participante o una autoridad de sesión termina la sesión o las sesiones del principal conforme a la petición (de ser posible) y devuelve un `<LogoutResponse>` al proveedor de identidad. La respuesta se puede devolver directamente o a través de un agente de usuario (si esto último no implica ninguna contradicción con la forma de la petición del paso 3).

5) El proveedor de identidad envía <LogoutResponse> al participante de sesión

En el paso 5, el proveedor de identidad envía un mensaje `<LogoutResponse>` al participante de sesión que originó la petición. La respuesta se puede devolver directamente o a través del agente de usuario (si esto último no implica ninguna contradicción con la forma de la petición del paso 1).

Un proveedor de identidad (que actúe como autoridad de sesión) puede iniciar este perfil en el paso 2 y enviar `<LogoutRequest>` a todos los participantes de sesión, con lo cual evitaría el paso 5.

11.4.4.3 Descripción de perfil

Si un participante de sesión inicia el perfil, hay que empezar por 11.4.4.3.1. Si lo hace el proveedor de identidad, con 11.4.4.3.2. En las descripciones a continuación, hay que tener en cuenta lo siguiente:

– **Servicio de desinscripción única**

Se trata del punto extremo de protocolo de desinscripción única en un proveedor de identidad o en un participante de sesión al cual se entregan los mensajes `<LogoutRequest>` o `<LogoutResponse>` (o un artefacto que los represente). Se pueden utilizar los mismos puntos extremos o puntos extremos diferentes para las peticiones y respuestas.

11.4.4.3.1 <LogoutRequest> enviado por un participante de sesión a un proveedor de identidad

Si un participante de sesión inicia el perfil de desinscripción, éste estudia la o las aserciones de autenticación que recibe y que tienen que ver con la o las sesiones que se van a terminar, y recolecta el valor o los valores `SessionIndex` que recibe del proveedor de identidad. Cuando participan varios proveedores de identidad, el perfil debe repetir este procedimiento para cada uno de ellos independientemente.

Para iniciar este perfil, el participante de sesión envía un mensaje `<LogoutRequest>` al punto extremo de petición de servicio de desinscripción única del proveedor de identidad que contiene uno o varios elementos `<SessionIndex>` aplicables. Debe haber por lo menos un elemento. Se pueden emplear metadatos para encontrar la ubicación de este punto extremo y las vinculaciones que soporta el proveedor de identidad.

Vinculaciones asíncronas (de canal frontal)

Conviene que el participante de sesión (si está presente el agente de usuario del principal) utilice una vinculación asíncrona, como Redirect, POST o Artifact HTTP (véase la cláusula 10), para enviar la petición al proveedor de identidad a través del agente de usuario. El proveedor de identidad debería entonces propagar todo mensaje de desinscripción que se ha requerido a los otros participantes de sesión, tal como se requiere, utilizando una vinculación bien sea síncrona o asíncrona. Se prefiere el empleo de una vinculación asíncrona para la petición original puesto que otorga al proveedor de identidad la mejor probabilidad de propagación con éxito de la desinscripción a otros participantes de la sesión durante el paso 3 en 11.4.4.2.

Si se utiliza la vinculación Redirect o POST HTTP, se entrega en este paso el mensaje `<LogoutRequest>` al proveedor de identidad. Si en su lugar se utiliza la vinculación Artifact HTTP, quiere decir que el proveedor de identidad está empleando el perfil de resolución de artefacto definido en 11.4.6, que hace una llamada al participante sesión para recuperar el mensaje `<LogoutRequest>`, a través de por ejemplo la vinculación SOAP.

Se recomienda efectuar los intercambios HTTP de este paso utilizando el TLS 1.0 a fin de mantener la integridad y la confidencialidad del mensaje. El mensaje `<LogoutRequest>` debe ir firmado cuando se utilicen las vinculaciones POST o Redirect HTTP. La vinculación Artifact HTTP, si la hubiere, también provee medios de autenticación del emisor de la petición cuando se deje de hacer referencia al artefacto.

Cada una de estas vinculaciones suministra un mecanismo RelayState que puede ser utilizado por el participante de sesión para hacer corresponder el intercambio de perfil con la petición original. Conviene que el participante de sesión oculte al máximo la información del valor RelayState a menos que el perfil que se esté utilizando no requiera que se tomen dichas medidas de privacidad.

Vinculaciones síncronas (de canal posterior)

De otra parte, el participante de sesión puede utilizar una vinculación síncrona, como la SOAP (véase la cláusula 10), para enviar la petición directamente al proveedor de identidad. En tal caso, el proveedor de identidad debería propagar todos los mensajes de desinscripción que sean necesarios a los participantes de sesión adicionales, utilizando la vinculación síncrona. Quien emite la petición debe autenticarse ante el proveedor de identidad, bien sea firmando el `<LogoutRequest>` o mediante cualquier otro mecanismo soportado por la vinculación.

En 11.4.4.4.1 se incluyen reglas específicas del perfil que atañen a los contenidos de mensaje `<LogoutRequest>`.

11.4.4.3.2 El proveedor de identidad establece quiénes son los participantes en la sesión

Si el proveedor de identidad inicia el perfil de desinscripción, o si recibe un mensaje `<LogoutRequest>` válido, tramita la petición, debe examinar el identificador y los elementos `<SessionIndex>` y luego determinar el conjunto de sesiones que deben terminarse.

El proveedor de identidad sigue entonces los pasos 3 y 4 de la figura 11-3 para cada entidad que participa en la o en las sesiones que están siendo terminadas, distinta del participante de sesión solicitante (si lo hubiere), como se describe en 8.2.7.

11.4.4.3.3 <LogoutRequest> enviado por el proveedor de identidad al participante o a la autoridad de sesión

Con el fin de propagar la desinscripción, el proveedor de identidad envía su propio `<LogoutRequest>` a una autoridad o a un participante de sesión de una sesión que se esté terminando. La petición se envía mediante una vinculación SAML acorde con la capacidad de quien responde y la disponibilidad del agente de usuario en el proveedor de identidad.

En general, la vinculación con la cual se recibió la petición original en el paso 1 de la figura 11-3 no indica cuál vinculación se ha de utilizar en este paso, salvo que como se indicó en el paso 1, la utilización de una vinculación síncrona, que evite el agente de usuario, impide al proveedor de identidad emplear una vinculación similar para la propagación de peticiones adicionales.

En 11.4.4.4.1 se presentan reglas específicas del perfil para los contenidos del mensaje `<LogoutRequest>`.

11.4.4.3.4 El participante en la sesión o la autoridad de sesión envía <LogoutResponse> al proveedor de identidad

El participante en la sesión o la autoridad de sesión debe procesar el mensaje `<LogoutRequest>` conforme a 8.2.7. Tras lo cual, o al encontrar un error, debe enviar un mensaje `<LogoutResponse>` que contenga el código de estatus adecuado al proveedor de entidad de la identidad que solicita, con el fin de completar el intercambio de protocolo SAML.

Vinculaciones síncronas (canal posterior)

Si el proveedor de identidad utiliza una vinculación síncrona, tal como la SOAP (véase la cláusula 10), la respuesta se envía directamente para completar dicha comunicación. Quien responde debe autenticarse a sí mismo ante el proveedor de identidad que solicita, bien sea firmando el `<LogoutResponse>` o mediante cualquier otro mecanismo que soporte dicha vinculación.

Vinculaciones asíncronas (canal frontal)

Si el proveedor de identidad utiliza una vinculación asíncrona, como Redirect, POST o Artifact HTPP (véase la cláusula 10), se devuelve entonces el `<LogoutResponse>` (o artefacto) a través del agente de usuario al punto extremo de respuesta de servicio de desinscripción única del proveedor de identidad. Es posible utilizar metadatos para encontrar la ubicación este punto extremo y las vinculaciones que soporta el proveedor de identidad. Se puede utilizar cualquier tipo de vinculación asíncrona que soporten ambas entidades.

Si se utiliza la vinculación Redirect o POST HTTP, se entrega el mensaje `<LogoutResponse>` al proveedor de identidad en este paso. Si se emplea la vinculación Artifact HTTP, el proveedor de identidad utiliza el perfil de resolución de artefacto definido en 11.4.6, el cual efectúa una llamada a la entidad que responde para recuperar el mensaje `<LogoutResponse>`, utilizando por ejemplo la vinculación SOAP.

Se recomienda que los intercambios HTTP en este paso se hagan mediante el TLS 1.0 con el fin de conservar la confidencialidad e integridad del mensaje. El mensaje `<LogoutResponse>` debe ir firmado si se utiliza la vinculación POST o Redirect HTTP. La vinculación Artifact HTTP, si la hubiere, suministra también medios alternativos para autenticar a quien emite la respuesta cuando se acaba la referencia del artefacto.

En 11.4.4.4.2 se incluyen reglas específicas del perfil para los contenidos del mensaje `<LogoutResponse>`.

11.4.4.3.5 El proveedor de identidad envía <LogoutResponse> al participante de sesión

Tras procesar la `<LogoutRequest>` del participante de sesión original, como se describe en los pasos anteriores, el proveedor de identidad debe responder a la petición original con un `<LogoutResponse>` que contenga el código de estado adecuado para completar el intercambio de protocolo SAML.

La respuesta se envía al participante de sesión original, utilizando una vinculación SAML coherente con la vinculación utilizada en la petición original, la capacidad de quien responde, y la disponibilidad del agente de usuario en el proveedor de identidad. Si se supone que en el paso 1 de la figura 11-3 se utilizó la vinculación asíncrona, se puede utilizar cualquier vinculación soportada por ambas entidades.

En 11.4.4.4.2 se incluyen reglas específicas del perfil para los contenidos del mensaje `<LogoutResponse>`.

11.4.4.4 Utilización del protocolo de desinscripción única

En esta cláusula se describe la utilización de `<LogoutRequest>` y `<LogoutResponse>`.

11.4.4.4.1 Utilización de <LogoutRequest>

El elemento `<Issuer>` debe estar presente y contener el identificador único de la entidad que solicita; hay que ignorar el atributo `Format` o éste ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

La entidad que solicita debe autenticarse ante la que responde y garantizar la integridad de mensaje, bien sea firmando el mensaje o utilizando un mecanismo específico de la vinculación.

El principal debe identificarse en la petición mediante un identificador que **corresponda fuertemente** con el identificador en la aserción de autenticación emitida por quien solicita o que ha sido recibida en relación con la sesión que está terminando, conforme a las reglas de correspondencia que se definen en 8.2.7.

Si quien solicita es un participante en la sesión, debe incluir por lo menos un elemento `<SessionIndex>` en la petición. Si es una autoridad de sesión (o si actúa en su nombre), puede entonces ignorar cualquiera de dichos elementos para indicar la terminación de todas las sesiones que se aplican al principal.

NOTA (informativa) – PE38 (véase OASIS PE:2006) aclara el párrafo anterior de la siguiente manera:

Si quien solicita es un participante en la sesión, debe incluir por lo menos un elemento `<SessionIndex>` en la petición. (Conforme a 11.4, el participante en la sesión siempre recibe un atributo `SessionIndex` en los elementos `<saml:AuthnStatement>` que recibe para iniciar la sesión.) Si quien solicita es una autoridad de sesión (o actúa en su nombre), puede entonces ignorar todos los elementos de este tipo para indicar la terminación de todas las sesiones aplicables al principal.

11.4.4.4.2 Utilización de <LogoutResponse>

El elemento `<Issuer>` debe estar presente y contener el identificador único de la entidad que responde; hay que ignorar el atributo `Format` o éste ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Quien responde debe autenticarse ante quien solicita y garantizar la integridad del mensaje, ya sea firmándolo o mediante un mecanismo específico de la vinculación.

11.4.4.5 Utilización de metadatos

El elemento de punto extremo `<md:SingleLogoutService>` describe las vinculaciones y ubicaciones soportadas a las cuales una entidad puede enviar peticiones y respuestas a través de este perfil. Quien solicita, si está criptando el identificador del principal, puede emplear el elemento `<md:KeyDescriptor>` de quien responde con un atributo de criptación para establecer el algoritmo adecuado y la configuración que se ha de emplear, junto con una clave pública que sirve para entregar una clave de criptación general.

11.4.5 Perfil de gestión de identificador de nombre

En el caso soportado por el perfil de gestión de identificador de nombre, un proveedor de identidad ha intercambiado con un proveedor de servicio algún tipo de identificador persistente para un principal, con lo cual pueden compartir un identificador común durante un cierto periodo de tiempo. Es probable que el proveedor de identidad quiera entonces notificar al proveedor de servicio de un cambio en el formato o en el valor que empleará para identificar el mismo principal en el futuro. De otra parte, puede ocurrir que el proveedor de servicio desee añadir su propio "alias" al principal con el fin de garantizar que el proveedor de identidad lo incluya al comunicarse con él en el futuro acerca del principal. Para terminar, uno de los proveedores puede querer informar al otro que no volverá a enviar o aceptar mensajes que utilicen determinado identificador. Con el fin de implementar dichos casos, se utiliza un perfil del protocolo de gestión de identificador de nombre SAML.

NOTA (informativa) – PE12 (véase OASIS PE:2006) sugiere que se cambie la segunda frase del párrafo anterior por la siguiente: Es probable que el proveedor de identidad quiera entonces notificar al proveedor de servicio de un cambio en el valor que empleará para identificar el mismo principal en el futuro.

El perfil permite que se combine el protocolo con una vinculación síncrona, como la SOAP o vinculaciones asíncronas de "canal frontal", como el Redirect, POST o Artifact HTTP. Puede ser necesario utilizar una vinculación de canal frontal, por ejemplo, cuando se requiera la interacción directa entre el agente de usuario y el proveedor que responde para poder efectuar el intercambio.

11.4.5.1 Información necesaria

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:nameid-mgmt`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.5.2 Descripción general de perfil

En la figura 11-4 se muestra la plantilla básica del perfil de gestión de identificador de nombre.

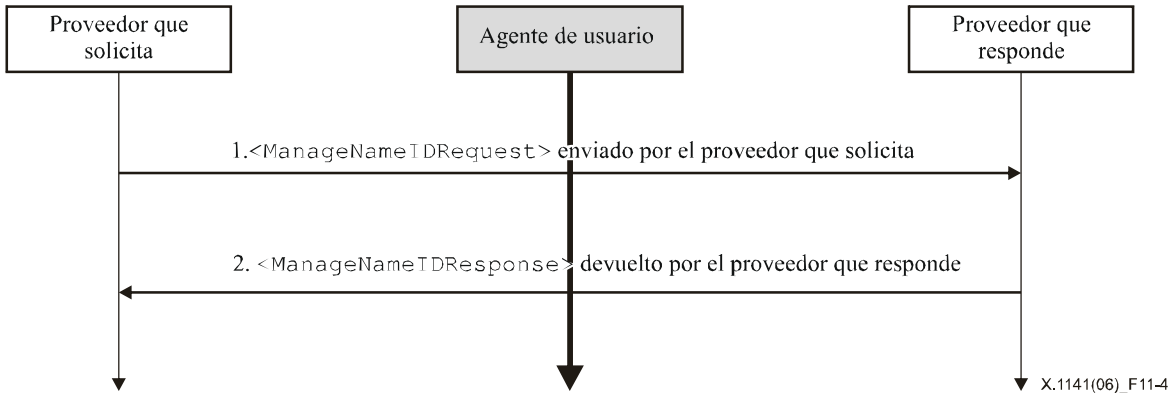

Figura 11-4/X.1141 – Perfil de gestión de identificador de nombre

Se presenta el agente de usuario con un tono de gris más ligero para indicar que el intercambio de mensajes puede pasar a través de él o puede ser directo entre las entidades del sistema, dependiendo del tipo de vinculación SAML que se utilice para implementar el perfil.

El perfil describe los pasos siguientes. En un paso determinado, puede haber uno o varios intercambios reales de mensajes dependiendo de la vinculación empleada para dicho paso y de otro comportamiento que dependa de la implementación en cuestión.

1) <ManageNameIDRequest> enviado por el proveedor de identidad o de servicio que solicita

En el paso 1, un proveedor de identidad o de servicio inicia el perfil enviando un mensaje `<ManageNameIDRequest>` a otro proveedor al que desea informar acerca de un cambio. La petición se puede enviar directamente al proveedor que responde o a través del agente de usuario.

2) <ManageNameIDResponse> enviado por el proveedor de identidad o de servicio que responde

En el paso 2, el proveedor que responde (tras haber procesado la petición) envía un mensaje `<ManageNameIDResponse>` al proveedor que originó la petición. La respuesta puede ir directamente a éste o a través del agente de usuario (si es el caso y con arreglo a la petición del paso 1).

11.4.5.3 Descripción del perfil

En las descripciones a continuación téngase en cuenta lo siguiente:

Servicio de gestión de identificador de nombre

Éste es el punto extremo de protocolo de gestión de identificador de nombre en un proveedor de identidad de servicio al cual se entregan los mensajes `<ManageNameIDRequest>` o `<ManageNameIDResponse>` (o un artefacto que lo represente). Se puede utilizar el mismo o varios puntos extremos para las peticiones y las respuestas.

11.4.5.3.1 <ManageNameIDRequest> enviado por el proveedor de identidad o de servicio que solicita

Al iniciar el perfil, el proveedor que solicita envía un mensaje `<ManageNameIDRequest>` al punto extremo de petición de servicio de gestión de identificador de nombre de otro proveedor. Se pueden emplear metadatos para encontrar la ubicación de este punto extremo y las vinculaciones que soporta el proveedor que responde.

– **Vinculaciones síncronas (de canal posterior)**

El proveedor que solicita puede emplear una vinculación síncrona, como el SOAP (véase la cláusula 10), para enviar la petición directamente a otro proveedor. El que solicita debe autenticarse con el otro proveedor, bien sea firmando el `<ManageNameIDRequest>` o mediante otro mecanismo soportado por la vinculación.

– **Vinculaciones asíncronas (de canal frontal)**

En su lugar, el proveedor que solicita puede (si está presente el agente de usuario del principal) utilizar una vinculación asíncrona, como la Redirect, POST o Artifact HTTP (véase la cláusula 10), para enviar la petición al otro proveedor a través del agente de usuario.

Si se emplea la vinculación Redirect o POST HTTP, se entrega el mensaje `<ManageNameIDRequest>` al otro proveedor en este paso. Si se utiliza la vinculación Artifact HTTP, el otro proveedor utiliza el perfil de resolución de artefacto que se define en 11.4.6, que hace una llamada al proveedor que solicita para recuperar el mensaje `<ManageNameIDRequest>`, mediante por ejemplo la vinculación SOAP.

Se recomienda que el intercambio HTTP de este paso se haga utilizando el TLS 1.0 con el fin de garantizar la confidencialidad y la integridad del mensaje. El mensaje `<ManageNameIDRequest>` debe ir firmado siempre que se utilice la vinculación POST o Redirect HTTP. La vinculación Artifact HTTP, si la hubiere, suministra también medios alternativos para autenticar a quien emite la petición cuando se acaba la referencia del artefacto.

Cada una de estas vinculaciones suministra un mecanismo RelayState que puede ser empleado por el proveedor que solicita para hacer corresponder el intercambio de perfil con la petición original. Conviene que el proveedor que solicita mantenga tanto como se pueda en secreto la información del valor RelayState, salvo si el perfil que se utiliza no requiere dichas medidas de privacidad.

En 11.4.5.4.1 se incluyen reglas específicas del perfil para los contenidos del mensaje `<ManageNameIDRequest>`.

11.4.5.3.2 <ManageNameIDResponse> enviado por el proveedor de identidad de servicio que responde

El destinatario debe procesar el mensaje `<ManageNameIDRequest>`. Tras hacerlo, o haber encontrado un error, debe enviar un mensaje al proveedor que emitió la petición, que contenga un código de estado adecuado, con el fin de completar el intercambio de protocolo SAML.

– **Vinculaciones síncronas (de canal posterior)**

Si el proveedor que solicita ha empleado una vinculación síncrona, como la SOAP (véase la cláusula 10), la respuesta se devuelve directamente con el fin de completar la comunicación síncrona. Quien responde debe autenticarse con el proveedor que solicita, bien sea firmando el <ManageNameIDResponse> o a través de cualquier otro mecanismo soportado por la vinculación.

– **Vinculaciones asíncronas (de canal frontal)**

Si el proveedor que solicita ha empleado una vinculación asíncrona, como las vinculaciones Redirect, POST o Artifact HTTP (véase la cláusula 10), se devuelve entonces el <ManageNameIDResponse> (o artefacto) a través del agente de usuario al punto extremo de respuesta de servicio de gestión de identificador de nombre del proveedor que solicita. Se pueden emplear metadatos para ubicar este punto extremo y las vinculaciones que soporta el proveedor que solicita. Es posible emplear cualquier vinculación que soporten ambas entidades.

Si se emplea la vinculación Redirect POST o HTTP, el mensaje <ManageNameIDResponse> se entrega al proveedor que solicita en este paso. Si se utiliza la vinculación Artifact HTTP, el proveedor que solicita utiliza el perfil de resolución de artefacto que se define en 11.4.6, que efectúa una llamada al proveedor que responde con el fin de recuperar el mensaje <ManageNameIDResponse>, a través de, por ejemplo, la vinculación SOAP.

Se recomienda que se efectúen los intercambios HTTP de este paso mediante el TLS 1.0, con el fin de garantizar la confidencialidad e integridad del mensaje. El mensaje <ManageNameIDResponse> debe ir firmado siempre que se empleen las vinculaciones POST o Redirect HTTP. La vinculación Artifact HTTP, si la hubiere, suministra también otra manera de autenticar a quien emite la respuesta cuando se deja de hacer referencia al artefacto.

En 11.4.5.4.2 se incluyen reglas específicas del perfil para los contenidos del mensaje <ManageNameIDResponse>.

11.4.5.4 Utilización del protocolo de gestión de identificador de nombre

Esta cláusula trata la utilización de ManageNameIDRequest y ManageNameIDResponse.

11.4.5.4.1 Utilización de <ManageNameIDRequest>

El elemento <Issuer> debe estar presente y contener el identificador único de la entidad que solicita; hay que ignorar el atributo Format o éste ha de tener un valor igual a urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Quien solicita debe autenticarse ante quien responde y garantizar la integridad del mensaje, bien sea firmándolo o utilizando un mecanismo específico de la vinculación.

11.4.5.4.2 Utilización de <ManageNameIDResponse>

El elemento <Issuer> debe estar presente y debe contener el identificador único de la entidad que responde; hay que ignorar el atributo Format o éste ha de tener un valor igual a urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Quien responde debe autenticarse ante el solicitante y garantizar la integridad del mensaje, bien sea firmándolo o a través de un mecanismo específico de la vinculación.

11.4.5.5 Utilización de metadatos

El elemento de punto extremo <md:ManageNameIDService> describe las vinculaciones y ubicación o ubicaciones soportadas a las cuales una entidad puede enviar peticiones y respuestas utilizando este perfil. Una entidad que solicita, si se cripta el identificador del principal, puede utilizar el elemento <md:KeyDescriptor> de quien responde con un atributo de criptación para establecer el algoritmo de criptación adecuado y la configuración que se debe emplear, junto con una clave pública que se ha de utilizar al presentar la clave general de criptación.

11.4.6 Perfil de resolución de artefacto

En la cláusula 10 se define un protocolo de resolución de artefacto para dejar de hacer referencia a un artefacto SAML en un mensaje de protocolo correspondiente. La vinculación Artifact HTTP (véase la cláusula 10) se vale de este mecanismo para hacer pasar mensajes de protocolo por referencias. En este perfil se describe la utilización de este protocolo con una vinculación síncrona, como la SOAP definida en la cláusula 10.

11.4.6.1 Información requerida

Identificación: urn:oasis:names:tc:SAML:2.0:profiles:artifact

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.6.2 Descripción general del perfil

El intercambio de mensajes y las reglas de procesamiento básicas que gobiernan este perfil se definen ampliamente en la cláusula 8, que trata los mensajes que se deben intercambiar, junto con las vinculaciones utilizadas para ello. En la cláusula 10 se define la vinculación del intercambio de mensajes al SOAP V1.1. A menos que se indique específicamente lo contrario en esta Recomendación, se aplican todos los requisitos definidos en dichas cláusulas.

En la figura 11-5 se muestra la plantilla básica del perfil de resolución de artefacto.

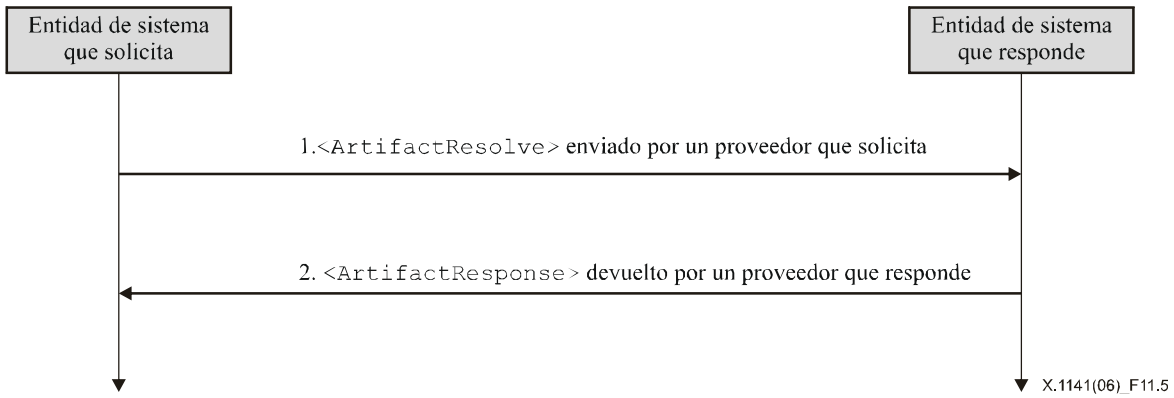

Figura 11-5/X.1141 – Plantilla básica del perfil de resolución de artefacto

El perfil describe los pasos siguientes.

1) <ArtifactResolve> enviado por la entidad que solicita

En el paso 1, una entidad solicitante inicia el perfil al enviar un mensaje `<ArtifactResolve>` a un emisor de artefacto.

2) <ArtifactResponse> enviado por la entidad que responde

En el paso 2, quien responde (tras procesar la petición) envía un mensaje `<ArtifactResponse>` al que solicita.

11.4.6.3 Descripción de perfil

En la siguiente descripción, téngase en cuenta lo siguiente:

– **Servicio de resolución de artefacto**

Éste es el punto extremo del protocolo de resolución de artefacto en un emisor de artefacto al cual se entregan los mensajes `<ArtifactResolve>`.

11.4.6.3.1 <ArtifactResolve> enviado por una entidad solicitante

Para iniciar el perfil, una entidad solicitante que haya recibido un artefacto y establecido el remitente a través del `SourceID`, envía un mensaje `<ArtifactResolve>` que contenga el artefacto a un punto extremo de servicio de resolución de artefacto de emisor de artefacto. Se pueden utilizar metadatos para encontrar la ubicación de este punto extremo y las vinculaciones que soporta el emisor de artefacto.

El solicitante debe utilizar una vinculación síncrona, como la SOAP (véase la cláusula 10), para enviar la petición directamente al emisor de artefactos. El solicitante debe autenticarse con quien responde, bien sea firmando el mensaje `<ArtifactResolve>` o mediante cualquier otro mecanismo soportado por la vinculación. Los perfiles específicos que utilicen la vinculación Artifact HTTP pueden imponer otros requisitos adicionales que hagan obligatoria la autenticación.

En 11.4.6.4.1 se incluyen reglas específicas del perfil para los contenidos del mensaje `<ArtifactResolve>`.

11.4.6.3.2 <ArtifactResponse> enviado por la entidad que responde

El emisor de artefacto debe procesar el mensaje `<ArtifactResolve>` conforme a la cláusula 8. Tras hacerlo o tras haber encontrado un error, debe devolver un mensaje `<ArtifactResponse>` que contenga un código de estado adecuado a la entidad que origina la petición, con el fin de completar el intercambio de protocolo SAML. Si tiene éxito, se incluirá también el mensaje de protocolo SAML al que ya no se hace referencia que corresponde al artefacto.

Quien responde debe autenticarse ante el solicitante, bien sea firmando el `<ArtifactResponse>` o mediante cualquier mecanismo soportado por la vinculación.

En 11.4.6.4.2 se incluyen reglas específicas de perfil para los contenidos del `<ArtifactResponse>`.

11.4.6.4 Utilización del protocolo de resolución de artefacto

Esta cláusula trata la utilización de `ArtifactResolve` y `ArtifactResponse`.

11.4.6.4.1 Utilización de <ArtifactResolve>

El elemento `<Issuer>` debe estar presente y debe contener el identificador único de la entidad que solicita; se debe ignorar el atributo `Format` o éste ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

El solicitante debería autenticarse ante quien responde y garantizar la integridad del mensaje, bien sea firmándolo o bien sea a través de cualquier otro mecanismo específico de la vinculación. Es posible que perfiles específicos que utilicen la vinculación Artifact HTTP impongan requisitos adicionales de tal manera que sea obligatoria la autenticación.

11.4.6.4.2 Utilización de <ArtifactResponse>

El elemento `<Issuer>` debe estar presente y contener el identificador único del emisor de artefacto; se debe ignorar el atributo `Format` o éste ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Quien responde debe autenticarse ante el solicitante y garantizar la integridad del mensaje, bien sea firmándolo o a través de un mecanismo específico de la vinculación.

11.4.6.5 Utilización de metadatos

En la cláusula 9 se define un elemento de punto extremo indexado, `<md:ArtifactResolutionService>`, que sirve para describir las vinculaciones y la ubicación o ubicaciones soportadas a las cuales una entidad solicitante puede enviar peticiones utilizando este perfil. El atributo `index` se emplea para distinguir los puntos extremos posibles que pueden ser especificados por referencia en el campo `EndpointIndex` del artefacto.

11.4.7 Perfil de consulta o petición de aserción

En la cláusula 10 se define un protocolo para solicitar aserciones existentes mediante la referencia a ellas o mediante consultas basadas en un sujeto y en un criterio adicional específico de la declaración. En este perfil se describe la utilización de este protocolo mediante una vinculación síncrona, como la SOAP definida en la cláusula 10.

11.4.7.1 Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:query`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

11.4.7.2 Descripción general del perfil

El intercambio de mensajes y las reglas de procesamiento básicas que gobiernan este perfil se describen ampliamente en la cláusula 8 que define los mensajes que han de intercambiarse, junto con las vinculaciones que se utilizan para dicho intercambio. En la cláusula 10 se define la vinculación del intercambio de mensajes al SOAP V1.1. A menos que se indique lo contrario, se aplican todos los requisitos definidos en dichas especificaciones.

En la figura 11-6 se muestra la plantilla básica para el perfil de consulta o petición de aserción.

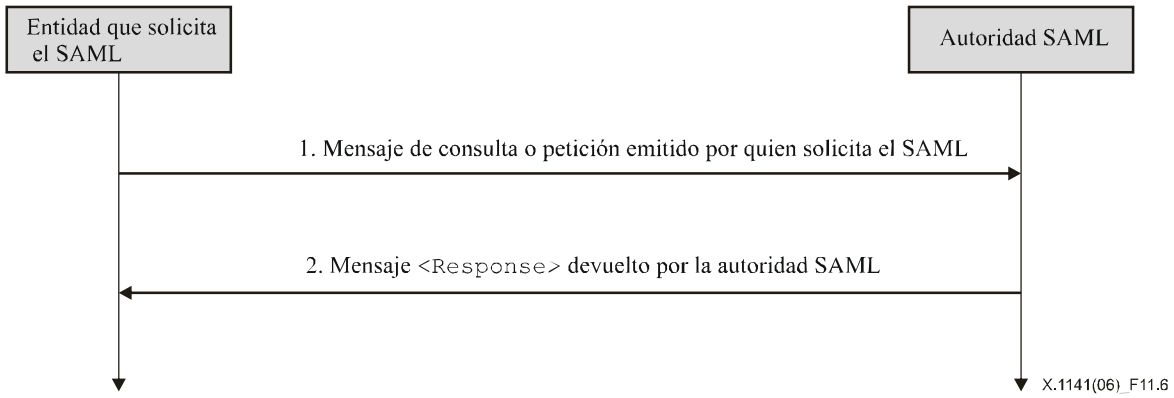

Figura 11-6/X.1141 – Plantilla básica para el perfil de consulta o petición

En este perfil se describen los siguientes pasos:

1) Consulta o petición enviada por quien solicita el SAML

En el paso 1, una entidad que solicita SAML inicia el perfil enviando un mensaje <AssertionIDRequest>, <SubjectQuery>, <AuthnQuery>, <AttributeQuery> o <AuthzDecisionQuery> a la autoridad SAML.

2) <Response> emitido por la autoridad SAML

En el paso 2, la autoridad SAML que responde (tras haber procesado la consulta o petición) envía un mensaje <Response> a quien solicita el SAML.

11.4.7.3 Descripción del perfil

En la siguiente descripción téngase en cuenta lo siguiente:

– **Servicio de consulta o petición**

Éste es el punto extremo de protocolo de consulta o petición en una autoridad SAML al cual se entregan los mensajes <AssertionIDRequest>.

11.4.7.3.1 Consulta o petición emitida por quien solicita el SAML

Para iniciar el perfil, quien solicita el SAML envía un mensaje <AssertionIDRequest>, <SubjectQuery>, <AuthnQuery>, <AttributeQuery> o <AuthzDecisionQuery> al punto extremo de servicio de consulta a petición de la autoridad SAML. Se pueden emplear metadatos para ubicar este punto extremo y las vinculaciones que soporta la autoridad SAML.

El solicitante SAML debe emplear una vinculación síncrona, como la SOAP (véase la cláusula 10), para enviar la petición directamente al proveedor de identidad. El solicitante debe autenticarse con la autoridad SAML bien sea firmando el mensaje o mediante cualquier otro mecanismo soportado por la vinculación.

En 11.4.7.4.1 se incluyen reglas específicas de perfil para los contenidos de estos mensajes.

1.4.7.3.2 <Response> emitido por la autoridad SAML

La autoridad SAML debe procesar el mensaje de consulta o petición conforme a la cláusula 8. Tras haberlo procesado o haber encontrado un error, debe devolver un mensaje <Response> que contenga un código de estado adecuado a quien solicita el SAML, con el fin de completar el intercambio de protocolo SAML. Si la petición tiene éxito al localizar una o varias aserciones que corresponden, se incluirán también en la respuesta.

Quien responde debería autenticarse ante el solicitante, bien sea firmando el <Response> o utilizando cualquier otro mecanismo que soporte la vinculación.

En 11.4.7.4.2 se incluyen reglas específicas del perfil para los contenidos del mensaje <Response>.

11.4.7.4 Utilización del protocolo de consulta o petición

En esta cláusula se define el punto extremo del protocolo de consulta o petición en una autoridad SAML al cual se entregan los mensajes de consulta.

11.4.7.4.1 Utilización de la consulta o petición

El elemento `<Issuer>` debe estar presente.

El solicitante debería autenticarse ante quien responde y garantizar la integridad del mensaje, bien sea firmándolo o empleando un mecanismo específico de la vinculación.

11.4.7.4.2 Utilización de <Response>

El elemento `<Issuer>` debe estar presente y contener el identificador único de la autoridad SAML que responde; hay que ignorar el atributo `Format` o éste ha de tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. No siempre corresponderá con el elemento `<Issuer>` en la aserción o aserciones devueltas.

Quien responde debería autenticarse ante el solicitante y garantizar la integridad del mensaje, bien sea firmándolo o a través de un mecanismo específico de la vinculación.

11.4.7.5 Utilización de metadatos

En la cláusula 9 se definen varios elementos de punto extremo `<md:AssertionIDRequestService>`, `<md:AuthnQueryService>`, `<md:AttributeService>` y `<md:AuthzService>`, que sirven para describir las vinculaciones y ubicación o ubicaciones que se soportan a las cuales un solicitante puede enviar peticiones o consultas utilizando este perfil.

La autoridad SAML, si está criptando las aserciones o los contenidos de la aserción resultantes para determinada entidad, puede utilizar el elemento `<md:KeyDescriptor>` de la entidad con un atributo de utilización de criptación para establecer un algoritmo y una configuración de criptación adecuados que se deban emplear, junto con una clave pública que sirva para entregar la clave general de criptación.

Los diferentes descriptores de función pueden contener elementos `<md:NameIDFormat>`, `<md:AttributeProfile>` y `<saml:Attribute>` (cuando corresponda) para indicar la capacidad general de soporte de determinados formatos de identificador de nombre, perfiles de atributo o atributo y valores específicos. La capacidad de soportar cualquiera de dichas características durante una petición determinada depende de la política de la autoridad y es de su potestad.

11.4.8 Perfil de correspondencia de identificador de nombre

En la cláusula 8.2.6 se define un protocolo de correspondencia de identificador de nombre que hace corresponder el identificador de nombre del principal con un identificador de nombre diferente para el mismo principal. Este perfil describe el empleo de dicho protocolo con una vinculación síncrona, como la SOAP que se define en la cláusula 10, y proporciona directrices adicionales relativas a la protección de la privacidad del principal, con la criptación y limitando la utilización del identificador al que se hace corresponder.

11.4.8.1 Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:nameidmapping`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualización: Ninguna.

11.4.8.2 Descripción general del perfil

El intercambio de mensajes y las reglas de procesamiento básicas que gobiernan este perfil se describen ampliamente en la cláusula 8 que define los mensajes que han de intercambiarse, junto con la vinculación utilizada para ello. En la cláusula 10 se define la vinculación del intercambio de mensajes en la versión 1.1 del SOAP. A menos que se especifique lo contrario, se aplican todos los requisitos definidos en dichas especificaciones.

En la figura 11-7 se muestra la plantilla básica para el perfil de correspondencia de identificador de nombre.

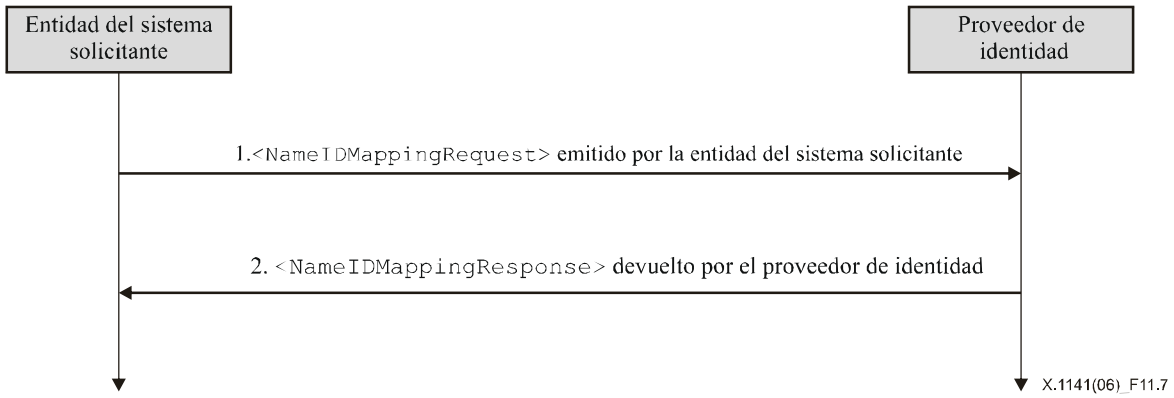

Figura 11-7/X.1141 – Plantilla básica para el perfil de identificador de nombre

El perfil describe los pasos siguientes:

1) <NameIDMappingRequest> emitido por la entidad solicitante

En el paso 1, un solicitante inicia el perfil enviando un mensaje `<NameIDMappingRequest>` a un proveedor de identidad.

2 <NameIDMappingResponse> emitido por el proveedor de identidad

En el paso 2, el proveedor de identidad que responde (tras el procesamiento de la petición) envía un mensaje `<NameIDMappingResponse>` al solicitante.

11.4.8.3 Descripción del perfil

En esta cláusula se utiliza el servicio de correspondencia de identificador de nombre, que es el punto extremo del protocolo de correspondencia de identificador de nombre en un proveedor de identidad al cual se entregan los mensajes `<NameIDMappingRequest>`.

11.4.8.3.1 <NameIDMappingRequest> emitido por la entidad solicitante

Para iniciar el perfil, un solicitante envía un mensaje `<NameIDMappingRequest>` al punto extremo de servicio de correspondencia de identificador de nombre del proveedor de identidad. Se pueden utilizar metadatos para encontrar la ubicación de este punto extremo y las vinculaciones que soporta el proveedor de identidad.

El solicitante debe utilizar una vinculación síncrona, como la SOAP (véase la cláusula 10), para enviar la petición directamente al proveedor de identidad. El solicitante debe autenticarse ante el proveedor de identidad bien sea firmando el `<NameIDMappingRequest>` o mediante cualquier otro mecanismo soportado por la vinculación.

En 11.4.8.4.1 se incluyen reglas específicas de perfil para los contenidos del mensaje `<NameIDMappingRequest>`.

11.4.8.3.2 <NameIDMappingResponse> emitido por el proveedor de identidad

El proveedor de identidad debe procesar el mensaje `<ManageNameIDRequest>` con arreglo a la cláusula 8. Tras procesarlo o haber encontrado un error, debe devolver al solicitante un mensaje `<NameIDMappingResponse>` que contenga un código de estatus adecuado con el fin de completar el intercambio de protocolo SAML.

Quien responde debe autenticarse ante el solicitante, bien sea firmando el `<NameIDMappingResponse>` o mediante otro mecanismo soportado por la vinculación.

En 11.4.8.4.2 se definen reglas específicas de perfil para los contenidos del mensaje `<NameIDMappingResponse>`.

11.4.8.4 Utilización del protocolo de correspondencia de identificador de nombre

En la cláusula 8 se define un protocolo de correspondencia de identificador de nombre para hacer corresponder un identificador de nombre de principal con otro identificador de nombre para el mismo principal. En esta cláusula se describe la utilización de este protocolo y se dan algunas otras directrices para proteger la privacidad del principal, tales como la limitación del empleo del identificador al que se hizo corresponder.

11.4.8.4.1 Utilización de <NameIDMappingRequest>

El elemento `<Issuer>` debe estar presente.

El solicitante debe autenticarse ante quien responde y garantizar la integridad del mensaje, bien sea firmándolo o a través de otro mecanismo específico de la vinculación.

11.4.8.4.2 Utilización de <NameIDMappingResponse>

El elemento `<Issuer>` debe estar presente y contener el identificador único del proveedor de identidad que responde; se debe ignorar el atributo `Format` o éste deberá tener un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Quien responde debe autenticarse ante el solicitante y garantizar la integridad del mensaje, bien sea firmándolo o a través de un mecanismo específico de la vinculación.

La criptación W3C, 2.2.3, define la utilización de la criptación para la confidencialidad de un identificador de nombre. En la mayoría de los casos, el proveedor de identidad debería criptar el identificador de nombre correspondido que devuelve al solicitante, con el fin de proteger la privacidad del principal. El solicitante puede extraer el elemento `<EncryptedID>` y ponerlo en mensajes o aserciones de protocolo ulteriores.

Limitaciones de la utilización del identificador correspondido

El proveedor de identidad puede aplicar límites adicionales a la utilización del identificador resultante, devolviendo el identificador de nombre correspondido en la forma de una `<Assertion>` que contenga el identificador en su `<Subject>` pero sin ninguna declaración. La aserción se cripta entonces y el resultado que se utiliza como elemento `<EncryptedData>` en el `<EncryptedID>` se devuelve al solicitante. La aserción se puede incluir en un elemento `<Conditions>` para limitar la utilización, conforme a la cláusula 8, como por ejemplo las restricciones basadas en el tiempo o la utilización por partes confiantes específicas y ha de ir firmada a efectos de protección de la integridad.

11.4.8.5 Utilización de metadatos

En esta cláusula se define un elemento de punto extremo, `<md:NameIDMappingService>`, que sirve para describir las vinculaciones y la ubicación o ubicaciones soportadas a las cuales un solicitante puede enviar peticiones utilizando este perfil.

El proveedor de identidad, si está criptando el identificador resultante para determinada entidad, puede utilizar este elemento `<md:KeyDescriptor>` de la entidad con un atributo de uso de criptación para establecer un algoritmo de criptación adecuado y la configuración que se debe emplear, junto con una clave pública que sirve para entregar la clave general de criptación.

11.4.9 Perfiles de atributo SAML

Los perfiles de atributo suministran las definiciones necesarias para restringir la expresión de atributo SAML cuando se trata de tipos de información de atributo particulares o cuando se interactúa con sistemas externos que requieren mayor rigurosidad. En esta cláusula se especifican el perfil de atributo básico SAML, el perfil X.500/LDAP y los perfiles UUID y XACML.

11.4.9.1 Perfil de atributo básico

El perfil de atributo `básico` especifica una denominación simplificada, aunque no única, de atributos SAML, junto con valores de atributos que se basan en los tipos de datos Datatypes W3C incorporados, con lo cual se elimina la necesidad de extender los esquemas para hacer válida la sintaxis.

Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:basic`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

Denominación de atributo SAML

El atributo XML `NameFormat` en los elementos `<Attribute>` debe ser `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`.

El atributo XML `Name` debe ajustarse a las reglas especificadas para dicho formato, como se define en la cláusula 8.

– **Comparación de nombre de atributo**

Dos elementos <Attribute> se refieren al mismo atributo SAML si y solamente si los valores de sus atributos XML Name son iguales (en el sentido que se describe en la cláusula 8).

Atributos XML específicos de perfil

No se definen atributos XML adicionales para emplear con el elemento <Attribute>.

Valores de atributo SAML

El tipo de esquema de los contenidos del elemento <AttributeValue> debe ser uno de los definidos en el anexo A. El atributo xsi:type debe estar presente y debe tener el valor adecuado.

Ejemplo de perfil de atributo básico

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic"
             Name="FirstName">
      <saml:AttributeValue xsi:type="xs:string">By-
Tor</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.2 Perfil de atributo X.500/LDAP

Los directorios basados en la serie de Recomendaciones UIT-T X.500 y en RFC 3377 del IETF se suelen emplear con frecuencia. Se utiliza el esquema de directorio para establecer un modelo de la información que debe almacenarse en dicho directorio. En particular, en X.500 se emplean las definiciones de tipo de atributo para especificar la sintaxis y otras características de los atributos, la unidad básica de almacenamiento de información en un directorio (esta Recomendación se refiere a ellas como "atributos de directorio"). Se definen los tipos de atributo de directorio en esquemas en las especificaciones X.500 y LDAP, en esquema en otros documentos públicos (como por ejemplo el esquema inetOrgperson (véase RFC 2798 del IETF)), y el esquema definido a efectos privados. En cualquiera de estos casos, es útil utilizar dichos tipos de atributo de directorio en el contexto de declaraciones de atributo SAML, sin tener que crear manualmente las definiciones de atributo específica de SAML para ellos, y todo esto de una manera interoperable.

El perfil de atributo X.500/LDAP define un convenio común para la denominación y representación de dichos atributos cuando se los expresa como atributos SAML.

Información requerida

Identificación: urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500 (se trata también del espacio de nombres objetivo atribuido en el esquema de perfil X.500/LDAP correspondiente del anexo A).

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

Denominación de atributo SAML

El atributo XML NameFormat en los elementos <Attribute> debe ser urn:oasis:names:tc:SAML:2.0:attrname-format:uri.

Para construir nombres de atributo, se utiliza el espacio de nombres oid URN que se describe en RFC 3061 del IETF. En dicho método, el atributo XML Name se basa en el identificador de objeto atribuido al tipo de atributo directorio.

Ejemplo:

urn:oid:2.5.4.3

Puesto que en los procedimientos X.500 es necesario que cada tipo de atributo se identifique mediante un identificador único de objeto, este esquema de denominación garantiza que los nombres de atributo SAML que con él se establecen no son ambiguos.

Para que las personas los puedan leer, existe también un requisito en algunas aplicaciones que han de llevar un nombre de cadena opcional junto con el URN OID (como se define en RFC 3061 del IETF). A estos efectos se puede utilizar también el atributo facultativo XML FriendlyName (que se define en la cláusula 8). Si la definición del tipo de atributo de directorio incluye uno o varios descriptores (nombres cortos) para el tipo de atributo, el valor FriendlyName, si lo hubiere, debería ser uno de los descriptores definidos.

Dos elementos <Attribute> se refieren al mismo atributo SAML si y solamente si sus valores de atributo XML Name son iguales en el sentido de RFC 3061 del IETF. El atributo FriendlyName no tiene nada que ver en dicha comparación.

Atributos XML específicos del perfil

No se definen atributos XML adicionales para su utilización con el elemento <Attribute>.

Valores de atributo SAML

Los definiciones de tipo de atributo de directorio que se emplean en los directorios X.500 nativos especifican la sintaxis del atributo que utiliza ASN.1. Para su utilización en LDAP, las definiciones de atributo de directorio incluyen además una sintaxis LDAP la cual especifica cómo se deben representar los atributos o valores de aserción que son conformes a la sintaxis, cuando se las transfiere en el protocolo LDAP (conocido como la codificación específica LDAP). La codificación específica LDAP suele producir caracteres Unicode en forma UTF-8. Este perfil de atributo SAML especifica la forma de los valores de atributo SAML solamente para aquellos atributos de directorio que tengan sintaxis LDAP. Es posible que en futuras extensiones de este perfil se definan formatos de valor de atributo para atributos de directorio cuya sintaxis especifiquen otras codificaciones.

Con el fin de representar las reglas de codificación que se utilizan para determinado valor de atributo, el elemento <AttributeValue> debe contener un atributo XML denominado Encoding definido en el espacio de nombre XML urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500.

Para todo atributo de directorio en una sintaxis, cuya codificación específica LDAP produzca exclusivamente cadenas de caracteres UTF-8 como valores, se codifica el valor de atributo SAML como la simple cadena UTF-8 con el contenido del elemento <AttributeValue> sin espacios en blanco adicionales. En tales casos, se debe poner el atributo XML xsi:type a **xs:string**. El atributo XML Encoding específico del perfil se proporciona con un valor de LDAP.

Algunas de las sintaxis de atributo LDAP (y los OID asociados) a las cuales se aplica lo anterior son:

Descripción de tipo de atributo	1.3.6.1.4.1.1466.115.121.1.3
Cadena de bits	1.3.6.1.4.1.1466.115.121.1.6
Booleana	1.3.6.1.4.1.1466.115.121.1.7
Cadena de país	1.3.6.1.4.1.1466.115.121.1.11
DN	1.3.6.1.4.1.1466.115.121.1.12
Cadena de directorio	1.3.6.1.4.1.1466.115.121.1.15
Número de teléfono facsímil	1.3.6.1.4.1.1466.115.121.1.22
Tiempo generalizado	1.3.6.1.4.1.1466.115.121.1.24
Cadena IA5	1.3.6.1.4.1.1466.115.121.1.26
ENTERO	1.3.6.1.4.1.1466.115.121.1.27
Descripción de sintaxis LDAP	1.3.6.1.4.1.1466.115.121.1.54
Descripción de regla de correspondencia	1.3.6.1.4.1.1466.115.121.1.30
Descripción de utilización de regla de correspondencia	1.3.6.1.4.1.1466.115.121.1.31
Nombre y UID facultativo	1.3.6.1.4.1.1466.115.121.1.34
Descripción de forma de nombre	1.3.6.1.4.1.1466.115.121.1.35
Cadena numérica	1.3.6.1.4.1.1466.115.121.1.36
Descripción de clase de objeto	1.3.6.1.4.1.1466.115.121.1.37
Cadena de octetos	1.3.6.1.4.1.1466.115.121.1.40
OID	1.3.6.1.4.1.1466.115.121.1.38
Otra casilla de correo	1.3.6.1.4.1.1466.115.121.1.39
Dirección postal	1.3.6.1.4.1.1466.115.121.1.41
Dirección de presentación	1.3.6.1.4.1.1466.115.121.1.43
Cadena imprimible	1.3.6.1.4.1.1466.115.121.1.44
Aserción de subcadena	1.3.6.1.4.1.1466.115.121.1.58
Número de teléfono	1.3.6.1.4.1.1466.115.121.1.50
Tiempo UTC	1.3.6.1.4.1.1466.115.121.1.53

Para todas las otras sintaxis LDAP, el valor de atributo se codifica, como el contenido del elemento <AttributeValue>, mediante la codificación de base 64 que abarca el valor de atributo LDAP codificado en cadena de octetos ASN.1. Se ha de poner el atributo XML xsi:type a **xs:base64Binary**. El atributo XML Encoding específico de perfil se suministra con un valor "LDAP".

Al comparar valores de atributo SAML a los efectos de igualdad, se deben seguir las reglas de correspondencia especificadas para el directorio en cuestión para el tipo de atributo de directorio en cuestión (por ejemplo se distingue entre mayúsculas y minúsculas).

Esquema específico de perfil

El siguiente esquema muestra cómo se define el atributo XML `Encoding` específico del perfil (véase al anexo A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published
in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

Ejemplo

A continuación se presenta un ejemplo de una correspondencia del atributo de directorio "givenName", que representa el nombre de pila del sujeto de la aserción SAML. Su identificador de objeto es {joint-iso-itu-t(2) ds(5) attributeType(4) givenName(42)} y su sintaxis LDAP es cadena de directorio.

```
<saml:Attribute
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
             Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string"
             x500:Encoding="LDAP">Steven</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.3 Perfil de atributo UUID

El perfil de atributo UUID normaliza la expresión de los valores UUID como nombres y valores de atributo SAML. Se aplica siempre que el sistema origen del atributo identifique un atributo o su valor mediante un UUID.

Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:UUID`

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

Denominación de atributo SAML

El atributo XML `NameFormat` en los elementos `<Attribute>` debe ser `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Si la representación subyacente del nombre de un atributo es un UUID, se utiliza entonces el espacio de nombres `UUID` URN descrito en la Rec. UIT-T Rec. X.667. En este caso, el atributo XML `Name` se basa en la forma URN del UUID subyacente que identifica dicho atributo.

Ejemplo:

```
urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

Si la representación subyacente del nombre de atributo no es un UUID, se puede emplear cualquier forma de URI en el atributo XML `Name`.

Para que las personas puedan leerlo, puede también imponerse el requisito de que algunas aplicaciones lleven un nombre de cadena facultativo junto con el URI. A estos efectos se puede emplear el atributo XML facultativo `FriendlyName`.

Dos elementos `<Attribute>` se refieren al mismo atributo SAML si, y solamente si, sus valores de atributos XML `Name` son iguales en el sentido de la Rec. UIT-T X.667. En esta comparación no cumple ninguna función el atributo `FriendlyName`.

Atributos XML específicos de perfil

No se definen atributos XML adicionales para ser utilizados con el elemento `<Attribute>`.

Valores de atributo SAML

En los casos en el que el valor del atributo también es un UUID, hay que utilizar la misma sintaxis URN descrita anteriormente para expresar dicho valor dentro del elemento `<AttributeValue>`. Se debe fijar el atributo XML `xsi:type` a **xs:anyURI**.

Si el valor de atributo no es un UUID, no existe ninguna restricción a la utilización del elemento `<AttributeValue>`.

Ejemplo

A continuación se muestra un ejemplo de un atributo de registro ampliado DCE, la configuración "pre_auth_req", cuyo UUID es bien conocido: 6c9d0ec8-dd2d-11cc-abdd-080009353559 y cuyo valor es entero.

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
             Name="urn:uuid:6c9d0ec8-dd2d-11cc-abdd-080009353559"
             FriendlyName="pre_auth_req">
      <saml:AttributeValue xsi:type="xs:integer">1</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.4 Perfil de atributo XACML

Es posible emplear las aserciones de atributo SAML como información para efectuar decisiones de autorización conforme a la Rec. UIT-T X.1142. Al tratarse de un formato de atributo diferente del de XACML, hay que realizar una correspondencia. El perfil de atributo XACML permite dicha correspondencia puesto que normaliza la denominación, la sintaxis de valor y otros metadatos adicionales de atributos. Los atributos SAML generados con arreglo a este perfil pueden hacerse corresponder automáticamente con atributos XACML y utilizarse para la toma de decisiones de autorización XACML.

Información requerida

Identificación: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML` (se trata también del espacio de nombres objetivo atribuido en el esquema de perfil XACML correspondiente del anexo A).

Información de contacto: security-services-comment@lists.oasis-open.org

Descripción: Véase más adelante.

Actualizaciones: Ninguna.

Denominación de atributo SAML

El atributo XML `NameFormat` en los elementos `<Attribute>` debe ser `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

El atributo XML `Name` debe seguir las reglas especificadas para dicho formato, conforme a lo definido en la cláusula 8.

Para que pueda ser leído por las personas, es posible que existan requisitos para algunas aplicaciones que tengan que llevar un nombre de cadena adicional junto con el URN OID. En tal caso, se puede emplear el atributo XML facultativo `FriendlyName` (que se define en la cláusula 8), aunque éste no se puede traducir a un atributo XACML equivalente.

Dos elementos `<Attribute>` se refieren al mismo atributo SAML si y solamente si sus valores de atributos XML `Name` son iguales en una comparación binaria. En esta comparación el atributo `FriendlyName` no juega ningún papel.

Atributos XML específicos de perfil

El XACML requiere que cada atributo tenga un tipo de datos explícito. Con el fin de suministrar dicho valor de tipo de datos, se define un nuevo atributo XML con valor URI que se denomina `DataType`, en el espacio de nombres XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`.

Los elementos `<Attribute>` SAML que conforman este perfil deben incluir el atributo `DataType` calificado de espacio de nombres, o el valor que se supone debe ser igual a `http://www.w3.org/2001/XMLSchema#string`.

Si se emplean valores no normalizados, hay que ampliar cada PDP XACML que esté utilizando atributos SAML correspondidos con valores de `DataType` no estándar, con el fin de soportar los nuevos tipos de datos.

Valores de atributo SAML

La sintaxis del contenido del elemento `<AttributeValue>` ha de corresponder al tipo de datos expresado en el atributo XML `DataType` específico del perfil que aparece en el elemento `<Attribute>` vástago. Para los tipos de datos que corresponden a los definidos en la cláusula 8, se debería también emplear el atributo XML `xsi:type` en el elemento o los elementos `<AttributeValue>`.

Esquema específico del perfil

El siguiente esquema muestra cómo es definido el atributo XML `DataType` específico de perfil (anexo A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
              Custom schema for XACML attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

Ejemplo

A continuación se presenta un ejemplo de una correspondencia del atributo LDAP/X.500 "givenName", que representa el nombre de pila del sujeto de la aserción SAML. De igual manera, se indica que un solo atributo SAML puede corresponder a varios perfiles de atributo cuando ellos son compatibles entre sí.

```
<saml:Attribute
xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
      xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:LDAP"
             xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
             ldapprof:Encoding="LDAP"
             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
             Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string">By-
Tor</saml:AttributeValue>
</saml:Attribute>
```

NOTA – (informativa): PE39 (véase OASIS PE:2006) aclara el anterior ejemplo reemplazándolo por lo siguiente:

```
<saml:Attribute
  xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
      xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:AttributeValue:LDAP"
           xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
           ldapprof:Encoding="LDAP"
           NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
           Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>
</saml:Attribute>
```

12 Contexto de autenticación del SAML

En la presente Recomendación se describe una sintaxis para definir las declaraciones del contexto de autenticación y una lista inicial de las clases del contexto de autenticación.

12.1 Conceptos del contexto de autenticación

Si una parte confiante ha de confiar en la autenticación de un principal a través de una autoridad de autenticación, es posible que la parte confiante exija información adicional a la propia aserción a fin de evaluar el nivel de confianza que puede depositar en esa aserción. En esta Recomendación se define un esquema XML para crear declaraciones del contexto de autenticación, es decir, documentos XML que posibilitarán que la autoridad de autenticación proporcione dicha información a la parte confiante. Además, se definen varias clases de contexto de autenticación, es decir, las categorías a las que podrán corresponder muchas de las declaraciones del contexto de autenticación, simplificando por consiguiente su interpretación.

El SAML no recomienda un solo protocolo, tecnología o política para los procesos que permiten que las autoridades de autenticación expidan identidades a los principales y que éstos se autentiquen ulteriormente y por sí mismos ante la autoridad de autenticación. Las distintas autoridades de autenticación elegirán diferentes tecnologías, seguirán diferentes procesos y estarán vinculadas por distintas obligaciones jurídicas con relación a la forma en la que autentican a los principales.

Las decisiones adoptadas por una autoridad de autenticación en ese sentido dependerán en gran medida de los requisitos de las partes confiantes con las que interactúa dicha autoridad. Los propios requisitos serán determinados por la naturaleza del servicio (es decir, la sensibilidad de cualquier información intercambiada, el valor financiero asociado, la tolerancia al riesgo de las partes confiantes, etc.) que proporcionará la parte confiante al principal.

Por consecuencia, para cualquier servicio salvo uno trivial, si la parte confiante va a depositar suficiente confianza en las aserciones de autenticación que recibe de una autoridad de autenticación, será necesario que conozca las tecnologías, protocolos y procesos que se aplicaron o se siguieron en el mecanismo de autenticación original en el que se fundamentó la aserción de autenticación. Si la parte confiante dispone de esta información y confía en el origen de la aserción en cuestión, estará mejor habilitada para adoptar una decisión justificada y bien fundada en relación con los servicios a los que se debería permitir el acceso al sujeto de la aserción de autenticación.

El contexto de autenticación se define como la información adicional a la propia aserción de autenticación que puede exigir la parte confiante antes de adoptar una decisión justificada con respecto a una aserción de autenticación. Ese contexto puede incluir, aunque no es el único, el método de autenticación real.

12.2 Declaración del contexto de autenticación

Si una parte confiante tiene que confiar en la autenticación de otra entidad mediante una autoridad de autenticación, es posible que la parte confiante exija información adicional a la propia autenticación que le permitirá colocar la autenticación en un contexto de gestión de riesgo. La información puede incluir:

- Los mecanismos de identificación del usuario inicial (por ejemplo, cara a cara, en línea, secreto compartido).
- Los mecanismos para reducir al mínimo el compromiso de la documentación de identidad (referencias) (por ejemplo, frecuencia de renovación de las referencias, generación de claves en el lado del cliente).
- Los mecanismos para almacenar y proteger las referencias (por ejemplo, tarjeta inteligente, reglas de las contraseñas).
- El mecanismo o método de autenticación (por ejemplo, contraseña).

Las variaciones y permutaciones de las características antes enumeradas garantizan que no todas las aserciones de autenticación serán idénticas con relación a la confianza que les deposita la parte confiante; una aserción de autenticación particular se caracterizará por los valores de cada una de estas (y otras) variables.

Una autoridad de autenticación del SAML puede entregar la información de contexto de autenticación adicional a una parte confiante en la forma de una declaración de contexto de autenticación, es decir, un documento XML que se incorpora directamente o al que se hace referencia en la aserción de autenticación que la autoridad de autenticación proporciona a la parte confiante.

Los peticionarios del SAML pueden exigir que una autenticación cumpla con un contexto de autenticación especificado, identificándolo en una petición de autenticación. Un peticionario también puede especificar que una autenticación debe ser llevada a cabo con un contexto de autenticación que *excede* algún valor establecido (para una definición de "excede" que haya sido acordada).

12.2.1 Modelo de datos

Una declaración de contexto de autenticación particular definida en esta Recomendación capturará las características de los procesos, procedimientos y mecanismos que permiten que la autoridad de autenticación verifique el sujeto antes de emitir una identidad que proteja los secretos en los que se basan las autenticaciones subsiguientes y los mecanismos utilizados para esta autenticación. Estas características se categorizan en el esquema del contexto de autenticación como sigue:

- Identificación – Características que describen los procesos y mecanismos que emplea la autoridad de autenticación para crear inicialmente una asociación entre un sujeto y la identidad (o el nombre) que permitirá que se dé a conocer el sujeto.
- Protección técnica – Características que describen la manera en la que se mantiene la seguridad del "secreto" (cuyo conocimiento o posesión permite que el sujeto pueda autenticarse ante la autoridad de autenticación).
- Protección operacional – Características que describen los controles de seguridad en materia de procedimientos que emplea la autoridad de autenticación (por ejemplo, auditorías de seguridad, archivo de los registros).
- Método de autenticación – Características que definen los mecanismos que permiten que el sujeto de la aserción expedida se autentique ante la autoridad de autenticación (por ejemplo, una contraseña en oposición a una tarjeta inteligente).
- Acuerdos de rectoría – Características que describen el marco jurídico (por ejemplo, restricciones de responsabilidad y obligaciones contractuales) subyacente al evento de autenticación y/o su infraestructura de autenticación técnica.

12.2.2 Capacidad de ampliación

El esquema de declaración del contexto de autenticación tiene puntos bien definidos con posibilidad de ampliación mediante un elemento `<Extension>`. Las autoridades de autenticación pueden aprovechar este elemento para insertar detalles adicionales del contexto de autenticación para las aserciones del SAML que expiden (suponiendo que la parte confiante consumidora podrá interpretar estas extensiones). Estos elementos adicionales deben colocarse en espacios de nombre XML independientes de los del esquema básico o de clase de declaración de contexto de autenticación que se aplican a la propia declaración.

12.2.3 Reglas de procesamiento

En la cláusula 8 se especifican las reglas de procesamiento adicionales para las declaraciones del contexto de autenticación. Estas reglas de procesamiento equivalen a instalaciones que comparten interpretaciones comunes de la resistencia o calidad relativa de las declaraciones del contexto de autenticación particular y no pueden ser expresadas en términos absolutos o proporcionadas como reglas que se deben seguir durante las implementaciones.

12.2.4 Esquema

La presente cláusula no es normativa.

En el apéndice VI se enumeran todos los esquemas XML de tipos de contexto de autenticación y el propio esquema XML de contexto de autenticación, que se emplean para la validación de las declaraciones generalizadas individuales.

12.3 Clases del contexto de autenticación

El número de permutaciones de las diferentes características garantiza que hay un número teóricamente infinito de contextos de autenticación únicos. La consecuencia es que, en teoría, se esperaría que cualquier parte confiante particular pueda analizar sintácticamente las declaraciones del contexto de autenticación arbitrarias y, más importante,

analizar la declaración para evaluar la "calidad" de la aserción de autenticación asociada. La ejecución de una evaluación de ese tipo no es tarea trivial.

Afortunadamente, existe la posibilidad de una optimización. En la práctica, muchos de los contextos de autenticación corresponden a categorías determinadas por las prácticas y la tecnología en la industria. Por ejemplo, muchos contextos de autenticación del explorador web B2C serán definidos (parcialmente) por el principal que recibe la autenticación de la autoridad de autenticación mediante la presentación de una contraseña a través de una sesión protegida TLS. En el mundo de las empresas es común autenticar apoyándose en certificados. Por supuesto, el contexto de autenticación completo no se limita a los datos específicos de la forma en la que se autentica el principal. Sin embargo, a menudo el método de autenticación es la característica más *visible* y por consiguiente, puede ser útil como un clasificador para una clase de contextos de autenticación relacionados.

En esta Recomendación el concepto se expresa como la definición de una serie de *clases de contexto de autenticación*. Cada clase define un subconjunto pertinente de todo el conjunto de contextos de autenticación. Las clases se han elegido de manera que representen las prácticas y técnicas actuales de las tecnologías de autenticación, y proporcionen un método abreviado conveniente a las partes afirmadora y confiante cuando hacen referencia a las cuestiones relativas al contexto de autenticación.

Por ejemplo, una autoridad de autenticación puede incluir, junto con la declaración completa del contexto de autenticación que proporciona a una parte confiante, una aserción en la que se indique que el contexto de autenticación pertenece también a una clase de contexto de autenticación. Esta aserción representa, para algunas partes confiantes, un detalle suficiente para poder asignar un nivel de confianza adecuado a la aserción de autenticación asociada. Otras partes confiantes preferirían examinar la propia declaración completa del contexto de autenticación. De manera similar, la capacidad de una parte confiante para hacer referencia a una clase de contexto de autenticación en lugar de estar obligada a enumerar todos los detalles de una declaración del contexto de autenticación específica, simplificará la forma en la que la parte confiante puede expresar sus deseos y/o necesidades a una autoridad de autenticación.

12.3.1 Ventajas de las clases de contexto de autenticación

La introducción de la capa de clases adicional y la definición de una lista inicial de clases representativas y flexibles deberían permitir:

- Facilitar el establecimiento de un acuerdo entre la autoridad de autenticación y la parte confiante en cuanto a los contextos de autenticación aceptables, al poner a su disposición un marco de debate.
- Facilitar que las partes confiantes puedan indicar sus preferencias cuando solicitan una aserción de autenticación aumentada de una autoridad de autenticación.
- Simplificar a las partes confiantes la carga del procesamiento de las declaraciones del contexto de autenticación, al poner a su disposición la opción de quedar satisfechas con la clase asociada.
- Proteger a las partes confiantes de la repercusión de las nuevas tecnologías de autenticación.
- Facilitar que las autoridades de autenticación publiquen sus capacidades de autenticación, por ejemplo, mediante el lenguaje de descripción de servicios web (*WSDL*).

12.3.2 Reglas de procesamiento

En la cláusula 8 se describen reglas de procesamiento adicionales para las clases del contexto de autenticación. En casi todos los aspectos, estas reglas de procesamiento equivalen a instalaciones en las que se comparten interpretaciones comunes de la resistencia o calidad relativa de las clases del contexto de autenticación particulares y no pueden ser expresadas en términos absolutos o proporcionadas como reglas que se deben respetar durante las implementaciones.

12.3.3 Capacidad de ampliación

Tal y como en el caso del esquema de la declaración del contexto de autenticación principal, los esquemas de la clase de contexto de autenticación independientes permiten la inclusión del elemento `<Extension>` en algunos emplazamientos de la estructura de árbol. En general, cuando el elemento `<Extension>` se presenta como un vástago de un elemento `<xs:choice>`, esta opción se suprime creando la definición del esquema de clase pertinente como una restricción del tipo básico. Cuando el elemento `<Extension>` se presenta como un vástago facultativo de un elemento `<xs:sequence>`, se permite que el elemento `<Extension>` permanezca además de cualquier elemento necesario.

Por consiguiente, las declaraciones del contexto de autenticación pueden incluir el elemento `<Extension>` (con elementos adicionales en espacios de nombre diferentes) y seguir siendo conformes a los esquemas de clase del contexto de autenticación (si cumplen con los demás requisitos del esquema, por supuesto).

Los esquemas de clase del contexto de autenticación restringen las definiciones de tipo en el esquema del contexto de autenticación básico. Los propios esquemas de clase del contexto de autenticación pueden restringirse aun más, como un punto de extensión. Sus definiciones de tipo sirven como tipos básicos en algún otro esquema (definidos potencialmente por alguna comunidad que desea una clase de contexto de autenticación definida con mayor rigor). Para

evitar incoherencias lógicas, cualquier extensión de esquema de ese tipo sólo puede restringir aún más las definiciones de tipo del esquema de clase. Para hacer cumplir esta restricción, los esquemas de clase del contexto de autenticación se definen con el atributo `finalDefault="extension"` en el elemento `<schema>` a fin de impedir este tipo de derivación.

12.3.4 Esquemas

En las siguientes subcláusulas se enumeran en orden alfabético las clases del contexto de autenticación. El orden de las clases no implica ninguna otra categorización. Los implementadores pueden decidir qué clases se han de soportar de acuerdo con las directrices de conformidad en esta Recomendación (véase la cláusula 13). Las clases se identifican de manera inequívoca mediante URI que tengan la siguiente raíz inicial:

```
urn:oasis:names:tc:SAML:2.0:ac:classes
```

Los esquemas de clase se definen como restricciones de partes del esquema "types" del contexto de autenticación básico. Se dice que los ejemplares de XML válidos con referencia a un esquema de clase del contexto de autenticación dado tienen *conformidad* con esa clase del contexto de autenticación.

Como el esquema de clase importa y redefine los elementos y tipos en el espacio de nombre del esquema de clase, una declaración del contexto de autenticación conforme con la clase no se valida simultáneamente con referencia al esquema del contexto de autenticación básico.

12.3.4.1 Protocolo Internet

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase protocolo Internet se aplica cuando la autenticación de un principal se lleva a cabo mediante la utilización de una dirección IP proporcionada.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="IPAddress"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.2 InternetProtocolPassword (Contraseña del protocolo Internet)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword`

Obsérvese que este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase Contraseña del protocolo Internet se aplica cuando la autenticación de un principal se lleva a cabo mediante la utilización de una dirección IP proporcionada, además de un nombre de usuario/contraseña.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
```

```
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.3 Kerberos

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase puede aplicarse cuando la autenticación del principal se realizó mediante la presentación de una contraseña ante una autoridad de autenticación local, a fin de adquirir un tique Kerberos. A continuación, este tique se emplea para una autenticación de red subsiguiente.

NOTA 1 – Es posible que la autoridad de autenticación indique (a través de esta clase de contexto) un tipo de datos de preautenticación que fue aplicado por el centro de distribución de claves Kerberos [norma RFC 1510 del IETF] cuando se realizó la autenticación del principal. El método utilizado por la autoridad de autenticación para obtener esta información queda fuera del alcance de esta Recomendación, pero se recomienda firmemente que se despliegue un método de confianza que permita pasar el tipo de datos de preautenticación y cualquier otro detalle del contexto relacionado con Kerberos (por ejemplo, la vida útil del tique) a la autoridad de autenticación.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
```

```
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
              <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
  </xs:redefine>

</xs:schema>
```

A continuación se ilustra un ejemplar de XML conforme a este esquema de clase:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos">

  <AuthnMethod>

    <PrincipalAuthenticationMechanism preauth="0">
      <RestrictedPassword>
        <Length min="4"/>
      </RestrictedPassword>
    </PrincipalAuthenticationMechanism>

    <Authenticator>
      <AuthenticatorSequence>
        <SharedSecretChallengeResponse
method="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
      </AuthenticatorSequence>
    </Authenticator>

  </AuthnMethod>

</AuthenticationContextDeclaration>
```

NOTA 2 – La utilización de SSL se presenta en el apéndice IV.

12.3.4.4 MobileOneFactorUnregistered (Abonado móvil no registrado con autenticación mediante un factor)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Refleja que no se emplean procedimientos de registro de un abonado móvil y la autenticación del dispositivo móvil sin exigir interacción explícita del usuario de extremo. Esta clase de contexto sólo puede autenticar el dispositivo pero nunca al usuario; resulta útil cuando los servicios distintos de los del operador móvil desean añadir una autenticación de dispositivo segura a su proceso de autenticación.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnre
gistered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
```

```
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTA – La utilización de SSL se presenta en el apéndice IV.

12.3.4.5 MobileTwoFactorUnregistered (Abonado móvil no registrado con autenticación mediante dos factores)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Refleja que no se emplean procedimientos de registro del abonado móvil y una autenticación basada en dos factores, tal como la utilización de un dispositivo seguro y un PIN de usuario. Esta clase de contexto es útil cuando un servicio distinto del operador del servicio móvil desea enlazar su ID de abonado a un servicio móvil con autenticación mediante dos factores, captando datos del teléfono móvil durante la admisión.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnre
gistered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
```

```
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.6 MobileOneFactorContract (Abonado con contrato del servicio móvil y autenticación mediante un factor)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Refleja procedimientos de registro de un abonado con un contrato de servicio móvil y una autenticación mediante un solo factor. Por ejemplo, un dispositivo de firma digital con memoria de almacenamiento de claves resistente a la manipulación, tal como el MSISDN móvil, pero sin exigir un PIN o una prueba biométrica para la autenticación del usuario en tiempo real.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
```

```
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.7 MobileTwoFactorContract (Abonado con contrato del servicio móvil y autenticación mediante dos factores)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Refleja procedimientos de registro de un abonado con un contrato de servicio móvil y una autenticación basada en dos factores. Por ejemplo, un dispositivo de firma digital con memoria de almacenamiento de claves resistente a la manipulación, tal como el SIM GSM, que exige una prueba explícita de la identidad y propósito del usuario, tal como un PIN o una prueba biométrica.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorCont
ract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-
2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
```

```
            <xs:element ref="ComplexAuthenticator"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ComplexAuthenticatorType">
    <xs:complexContent>
      <xs:restriction base="ComplexAuthenticatorType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SharedSecretChallengeResponse"/>
            <xs:element ref="SharedSecretDynamicPlaintext"/>
          </xs:choice>
          <xs:element ref="Password"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
```

```
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTA – La utilización de SSL se presenta en el apéndice IV.

12.3.4.8 Contraseña

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase Contraseña puede aplicarse cuando el proceso de autenticación del principal ante la autoridad de autenticación se realiza a través de la presentación de una contraseña en una sesión HTTP no protegida.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
```

```
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A continuación se presenta un ejemplar XML que es conforme al esquema de clase de contexto:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password">

  <AuthnMethod>
    <Authenticator>
      <AuthenticatorSequence>
        <RestrictedPassword>
          <Length min="4"/>
        </RestrictedPassword>
      </AuthenticatorSequence>
    </Authenticator>
  </AuthnMethod>

</AuthenticationContextDeclaration>
```

12.3.4.9 PasswordProtectedTransport (Transporte protegido mediante una contraseña)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase PasswordProtectedTransport puede aplicarse cuando el proceso de autenticación del principal ante la autoridad de autenticación se realiza a través de la presentación de una contraseña en una sesión protegida.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTr
ansport"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTA – La utilización de SSL se presenta en el apéndice IV.

12.3.4.10 PreviousSession (Sesión anterior)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase PreviousSession puede aplicarse cuando el proceso de autenticación del principal ante una autoridad de autenticación se llevó a cabo en algún momento anterior aplicando cualquier contexto de autenticación soportado por dicha autoridad. Por consiguiente, cuando la autoridad de autenticación confirme un evento de autenticación subsiguiente a la parte confiante, dicho evento podrá estar separado significativamente en tiempo de la petición de acceso al recurso actual del principal.

El contexto de la sesión autenticada con anterioridad no se incluye en esta clase de contexto explícitamente porque el usuario no se autenticó durante esta sesión, y por lo tanto no debe aplicarse el mecanismo empleado por el usuario para autenticarse en la sesión anterior, como parte de una decisión relativa a si se autoriza ahora su acceso a un recurso.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.11 Clave pública – X.509

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:X509`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase de contexto X509 indica que el principal llevó a cabo su autenticación mediante una firma digital y que la clave fue validada como parte de una infraestructura de clave pública X.509.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
```

```
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.12 Clave pública – Privacidad bastante aceptable (PGP)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PGP`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase de contexto PGP indica que el principal llevó a cabo su autenticación mediante una firma digital y que la clave fue validada como parte de una infraestructura de clave pública PGP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
```

```
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.13 Clave pública – Infraestructura de clave pública única (SPKI)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase de contexto SPKI indica que el principal llevó a cabo su autenticación mediante una firma digital y que la clave fue validada como parte de una infraestructura SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
```

```
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.14 Clave pública – Firma digital XML

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase de contexto indica que el principal llevó a cabo su autenticación mediante una firma digital conforme a las reglas de procesamiento especificadas en las reglas de firma XML del W3C.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
```

```
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.15 Smartcard (Tarjeta inteligente)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase Smartcard se identifica cuando un principal lleva a cabo su autenticación ante una autoridad de autenticación mediante una tarjeta inteligente.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.16 SmartcardPKI (Infraestructura de clave pública con tarjeta inteligente)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase SmartcardPKI puede aplicarse cuando un principal realiza su autenticación ante una autoridad de autenticación mediante un mecanismo de autenticación basado en dos factores y utilizando una tarjeta inteligente con clave privada incorporada y un PIN.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
```

```
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.17 SoftwarePKI (Infraestructura de clave pública con software)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase SoftwarePKI puede aplicarse cuando un principal emplea un certificado X.509 almacenado en software para llevar a cabo su autenticación ante la autoridad de autenticación.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
```

```
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

12.3.4.18 Telefonía

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase se emplea para indicar que el principal llevó a cabo su autenticación mediante un número telefónico de línea fija, transportándola por un protocolo de telefonía como el ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
```

```
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.19 Telefonía (nómada)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Indica que el principal está "itinerante" (roaming) (empleando probablemente una tarjeta telefónica) y que realiza su autenticación mediante el número de línea, un sufijo de usuario y un elemento contraseña.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
```

```
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.20 Telefonía (personalizado)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalTelephony`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase se emplea para indicar que el principal llevó a cabo su autenticación mediante un número telefónico de línea fija y un sufijo de usuario, transportándola por un protocolo de telefonía como el ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.21 Telefonía (autenticado)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase se emplea para indicar que el principal llevó a cabo su autenticación mediante un número telefónico de línea fija, un sufijo de usuario y un elemento contraseña.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTeleph
ony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

12.3.4.22 Contraseña distante segura

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase SecureRemotePassword puede aplicarse cuando la autenticación se realizó por medio de una contraseña distante segura como se especifica en RFC 2945 del IETF.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI" fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.23 Autenticación del cliente basada en certificado del TLS

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

Esta clase se emplea para indicar que el principal llevó a cabo su autenticación mediante un certificado de cliente asegurado con el transporte TLS.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
```

```
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTA – La utilización de SSL se presenta en el apéndice IV.

12.3.4.24 TimeSyncToken (Testigo de sincronía de tiempo)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken`

Este URI se emplea también como el espacio de nombre objetivo en el esquema de clase del contexto de autenticación correspondiente en el anexo A.

La clase TimeSyncToken puede aplicarse cuando un principal realiza su autenticación a través de un testigo de sincronización de tiempo.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Token"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TokenType">
```

```
      <xs:complexContent>
        <xs:restriction base="TokenType">
          <xs:sequence>
            <xs:element ref="TimeSyncToken"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TimeSyncTokenType">
      <xs:complexContent>
        <xs:restriction base="TimeSyncTokenType">
          <xs:attribute name="DeviceType" use="required">
            <xs:simpleType>
              <xs:restriction base="DeviceTypeType">
                <xs:enumeration value="hardware"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="SeedLength" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="64"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="DeviceInHand" use="required">
            <xs:simpleType>
              <xs:restriction base="booleanType">
                <xs:enumeration value="true"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.25 No especificado

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Unspecified`

La clase Unspecified indica que la autenticación se realizó a través de medios no especificados.

13 Requisitos de conformidad para el SAML

En la presente cláusula se describen las características obligatorias y facultativas para las implementaciones que alegan conformidad con el SAML.

En esta Recomendación se define un número determinado de perfiles nombrados. Cada perfil (distinto de los perfiles de atributos) describe los detalles de los flujos del mensaje SAML seleccionado y también puede considerarse como una funcionalidad indivisible que podría ser implementada por un componente de software. La implementación de un perfil exige la utilización de una vinculación para cada intercambio de mensajes incluido en el perfil. Una vinculación puede considerarse como una técnica de implementación específica para lograr un intercambio de mensajes.

En esta cláusula se enumeran todos los distintos perfiles que se definen en esta Recomendación. Para cada perfil, se enumeran los flujos de mensaje SAML V2.0 pertinentes, y para cada flujo de mensaje se describe también el conjunto de posibles vinculaciones. La combinación de un perfil, un intercambio de mensajes y una vinculación seleccionada se denomina *característica* del SAML V2.0.

En esta cláusula se describe también la matriz de conformidad del SAML V2.0, y se identifica cierto número de *modos operacionales* o cometidos diferentes. Dicha matriz describe el conjunto de características que debe ser implementado por cada modo operacional.

13.1 Perfiles del SAML y posibles implementaciones

En el cuadro 1 se enumeran todos los perfiles definidos por los perfiles del SAML. Para cada perfil, se describen también los flujos del protocolo de mensajes que se encuentran dentro del perfil. Para cada flujo de mensaje, se presenta una lista de las vinculaciones pertinentes en la última columna.

Cuadro 1/X.1141 – Implementaciones posibles

Perfil	Flujos del mensaje	Vinculación
Web SSO	`<AuthnRequest>` from SP to IdP	HTTP Redirect
		HTTP POST
		HTTP Artifact
	IdP `<Response>` to SP	HTTP POST
		HTTP Artifact
Cliente/mandatario mejorado SSO	ECP to SP, SP to ECP to IdP	PAOS
	IdP to ECP to SP, SP to ECP	PAOS
Descubrimiento de proveedor de identidad	Cookie setter	HTTP
	Cookie getter	HTTP
Desinscripción única	`<LogoutRequest>`	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
	`<LogoutResponse>`	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
Gestión de identificador de nombre	`<ManageNameIDRequest>`	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
	`<ManageNameIDResponse>`	HTTP Redirect
		SOAP
Resolución de artefacto	`<ArtifactResolve>`, `<ArtifactResponse>`	SOAP
Consulta de autenticación	`<AuthnQuery>`, `<Response>`	SOAP
Consulta de atributo	`<AttributeQuery>`, `<Response>`	SOAP
Consulta de decisión de autorización	`<AuthzDecisionQuery>`, `<Response>`	SOAP
Petición de afirmación por identificador	`<AssertionIDRequest>`, `<Response>`	SOAP
Concordancia de identificador de nombre	`<NameIDMappingRequest>`, `<NameIDMappingResponse>`	SOAP
Vinculación URI SAML	GET, HTTP Response	HTTP
Perfil de atributo UUID		
Perfil de atributo DCE PAC		
Perfil de atributo X.500		
Perfil de atributo XACML		
Metadatos	Consumption	
	Exchange	

13.2 Conformidad

En esta cláusula se describen los requisitos de conformidad técnicos del SAML V2.0.

13.2.1 Modos operacionales

En esta Recomendación se emplea la frase "modo operacional" para describir un cometido que puede ser desempeñado por un componente de software para lograr la conformidad con el SAML. Los modos operacionales son:

- IdP – Proveedor de identidad
- IdP Lite – Proveedor de identidad ligero
- SP – Proveedor de servicio
- SP Lite – Proveedor de servicio ligero
- ECP – Cliente/mandatario mejorado
- Autoridad de atributo del SAML
- Autoridad de decisión de autorización del SAML
- Autoridad de autenticación del SAML
- Peticionario del SAML

13.2.2 Matriz de las características

En las siguientes matrices (véase el cuadro 2) se identifican conjuntos de requisitos de conformidad únicos mediante un triplete del cuadro 1 con el siguiente formato: perfil, mensaje(s), vinculación. El componente mensaje no se incluye cuando resulta obvio a partir del contexto.

Cuadro 2/X.1141 – Matriz de características

Característica	IdP	IdP lite	SP	SP lite	ECP
Web SSO, `<AuthnRequest>`, HTTP redirect	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Web SSO, `<Response>`, HTTP POST	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Web SSO, `<Response>`, HTTP artifact	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Resolución de artefacto, SOAP	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Cliente/mandatario mejorado SSO, PAOS	Obligatorio	Obligatorio	Obligatorio	Obligatorio	Obligatorio
Gestión de identificador de nombre, HTTP redirect (IdP-initiated)	Obligatorio	Prohibido	Obligatorio	Prohibido	N/A
Gestión de identificador de nombre, SOAP (IdP-initiated)	Obligatorio	Prohibido	Facultativo	Prohibido	N/A
Gestión de identificador de nombre, HTTP redirect NOTA (informativa) – En PE11 (véase OASIS PE:2006) se sugiere añadir (SP-initiated)	Obligatorio	Prohibido	Obligatorio	Prohibido	N/A
Gestión de identificador de nombre, SOAP (SP-initiated)	Obligatorio	Prohibido	Facultativo	Prohibido	N/A
Desinscripción única (IdP-initiated) – HTTP redirect	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Desinscripción única (IdP-initiated) – SOAP	Obligatorio	Facultativo	Obligatorio	Facultativo	N/A
Desinscripción única (SP-initiated) – HTTP redirect	Obligatorio	Obligatorio	Obligatorio	Obligatorio	N/A
Desinscripción única (SP-initiated) – SOAP	Obligatorio	Facultativo	Obligatorio	Facultativo	N/A
Descubrimiento de proveedor de identidad (cookie)	Obligatorio	Obligatorio	Facultativo	Facultativo	N/A

NOTA 1 (informativa) – En PE16 (véase OASIS PE:2006) se sugiere sustituir N/A por "Facultativo" en la última fila y la última columna del cuadro 2.

NOTA 2 (informativa) – En PE25 (véase OASIS PE:2006) se sugiere añadir lo siguiente al final del cuadro 2:

Característica	IdP	IdP Lite	SP	SP Lite	ECP
Estructuras de metadatos	Facultativo	Facultativo	Facultativo	Facultativo	N/A
Interfuncionamiento de metadatos	Facultativo	Facultativo	Facultativo	Facultativo	N/A

NOTA 3 (informativa) – En PE29 (véase OASIS PE:2006) se sugiere añadir lo siguiente al final del cuadro 2:

Característica	IdP	IdP Lite	SP	SP Lite	ECP
Petición de identificador de afirmación	Facultativo	N/A	N/A	N/A	N/A
Vinculación URI SAML	Facultativo	N/A	N/A	N/A	N/A

En el cuadro 3 se presenta un resumen de los modos operacionales que extienden los modos IdP o SP antes definidos. Éstos deben entenderse como una combinación de un modo IdP o SP del cuadro anterior con la característica ampliada correspondiente que se establece a continuación.

Cuadro 3/X.1141 – IdP, SP ampliados

Característica	IdP ampliado	SP ampliado
Apoderado de proveedor de identidad	Obligatorio	Obligatorio
Concordancia del identificador de nombre, SOAP	Obligatorio	Obligatorio

En el cuadro 4 se presenta un resumen de los requisitos de conformidad para las autoridades y los peticionarios del SAML.

Cuadro 4/X.1141 – Matriz de la autoridad y el peticionario del SAML

Característica	Autoridad de autenticación del SAML	Autoridad de atributo del SAML	Autoridad de decisión de autorización del SAML	Peticionario del SAML
Consulta de autenticación, SOAP	Obligatorio	Facultativo	Facultativo	Facultativo
Consulta de atributo, SOAP	Facultativo	Obligatorio	Facultativo	Facultativo
Consulta de decisión de autorización, SOAP	Facultativo	Facultativo	Obligatorio	Facultativo
Petición de afirmación por identificador, SOAP	Obligatorio	Obligatorio	Obligatorio	Facultativo
Vinculación URI SAML	Obligatorio	Obligatorio	Obligatorio	Facultativo

NOTA 4 (informativa) – En PE25 y PE42 (véase OASIS PE:2006) se sugiere modificar el cuadro 4 anterior como sigue:

Característica	Autoridad de autenticación del SAML	Autoridad de atributo del SAML	Autoridad de decisión de autorización del SAML	Peticionario del SAML
Consulta de autenticación, SOAP	Obligatorio	N/A	N/A	Facultativo
Consulta de atributo, SOAP	N/A	Obligatorio	N/A	Facultativo
Consulta de decisión de autorización, SOAP	N/A	N/A	Obligatorio	Facultativo
Petición de afirmación por identificador, SOAP	Obligatorio	Obligatorio	Obligatorio	Facultativo
Vinculación URI SAML	Obligatorio	Obligatorio	Obligatorio	Facultativo
Estructuras de metadatos	Facultativo	Facultativo	Facultativo	Facultativo
Interfuncionamiento de datos	Facultativo	Facultativo	Facultativo	Facultativo

13.2.3 Implementación de los identificadores definidos en el SAML

Todos los modos operacionales pertinentes deben implementar los siguientes identificadores que se definen en el SAML:

- Todos los identificadores de formato de nombre de atributo que se definen en la cláusula 8.
- Todos los identificadores de formato de identificador de nombre que se definen en la cláusula 8.

Las implementaciones del SAML conformes deben permitir la utilización de todas las constantes de identificador (véanse 8.1 y 8.2) cuando producen y consumen mensajes del SAML. Los productores de estos mensajes deben poder crear mensajes y los consumidores deben poder procesarlos con cualquiera de las constantes que se definen en estas cláusulas.

En Identificadores de nombre persistentes e Identificadores de nombre transientes se definen reglas de procesamiento normativas orientadas al productor de esos identificadores. Las implementaciones conformes están obligadas a soportar todas las reglas de procesamiento normativas. Los demás identificadores no especifican reglas de procesamiento normativas. Por consiguiente, la generación y el consumo de estos identificadores tiene significado sólo cuando las partes generadora y consumidora cuentan con un acuerdo definido externamente acerca de la interpretación semántica de los identificadores.

NOTA – En este contexto, "proceso" significa que la implementación tiene que analizar y tratar satisfactoriamente el identificador sin ningún fallo o devolución de un error. La forma en la que la implementación trata el identificador una vez procesado en este nivel, queda fuera del alcance de esta Recomendación.

Una implementación del SAML puede proporcionar los recursos antes descritos a través de un soporte directo de los identificadores o de la utilización de interfaces de programación soportadas. Las interfaces previstas para esta finalidad han de permitir la ampliación en forma de programa de la implementación del SAML a fin de poder tratar todos los identificadores que no son tratados de manera nativa por la implementación.

13.2.4 Implementación de elementos criptados

Todos los modos operacionales pertinentes deben tener la capacidad de procesar o generar los siguientes elementos criptados en cualquier contexto donde estén obligados a procesar o generar los elementos no criptados correspondientes, concretamente `<saml:NameID>`, `<saml:Assertion>` o `<saml:Attribute>`:

- `<saml:EncryptedID>`
- `<saml:EncryptedAssertion>`
- `<saml:EncryptedAttribute>`

13.2.5 Modelos de seguridad para las vinculaciones SOAP y URI

La implementación de los siguientes modelos de seguridad es obligatoria para todos los perfiles implementados utilizando la vinculación SOAP así como para la vinculación URI del SAML. Las autoridades y los peticionarios del SAML tienen que aplicar los siguientes métodos de autenticación:

- Sin autenticación de cliente o servidor.
- Autenticación básica HTTP con y sin TLS 1.0. El peticionario del SAML debe enviar con derecho de preferencia el encabezamiento de autorización con la petición inicial.
- Autenticación del servidor HTTP por TLS 1.0 con certificado en el lado del servidor.
- Autenticación mutua HTTP por TLS 1.0 con certificado en ambos lados, del servidor y de un cliente.

Si una autoridad del SAML aplica TLS 1.0, está obligada a emplear un certificado en el lado del servidor.

NOTA 1 (informativa) – En PE25 (véase OASIS PE:2006) se sugiere añadir la nueva subcláusula sobre Estructuras de metadatos siguiente:

Las implementaciones que alegan conformidad con el SAML pueden declarar cada conformidad del modo operacional a los metadatos del SAML eligiendo la opción de Estructuras de metadatos. Por lo que se refiere a cada modo operacional, la conformidad implica los siguientes:

Implementar los metadatos del SAML de conformidad con el formato de metadatos del SAML ampliable cada vez que la entidad par del interfuncionamiento tiene la oportunidad, como se establece en las especificaciones del SAML, de depender de la existencia de los metadatos del SAML. Cuando se elige la opción de estructuras de metadatos se produce el efecto de exigir que los metadatos estén disponibles para la entidad par del interfuncionamiento. La característica de interfuncionamiento de los metadatos que se describe más adelante, ofrece un medio para satisfacer este requisito.

Referenciar, consumir y observar estrictamente los metadatos del SAML, de conformidad con una entidad par de interfuncionamiento, cuando ya han expirado o ya no son válidos en memoria intermedia los metadatos conocidos y pertinentes para esa entidad par, la operación particular y el intercambio en curso, siempre que los metadatos estén disponibles y no estén prohibidos por causa de una política o la operación particular y ese intercambio específico.

NOTA 2 (informativa) – En PE25 (véase OASIS PE:2006) se sugiere añadir la nueva subcláusula acerca del interfuncionamiento de los metadatos siguiente:

La elección de la opción de interfuncionamiento de los metadatos exige la oferta de implementación, además de cualquier otro mecanismo, la publicación del emplazamiento conocido y el mecanismo de resolución que se describe en la cláusula 9, Metadatos del SAML.

13.3 Firma digital XML y criptación XML

El SAML V2.0 emplea la firma XML para implementar funcionalidad de firma y criptación XML a los efectos de la integridad y la autenticación del origen. Asimismo, utiliza la criptación XML para poder implementar confidencialidad, incluyendo identificadores, aserciones y atributos criptados.

13.3.1 Algoritmos de firma XML

En el documento W3C XML Signature, 6.1, se exige la utilización de lo siguiente:

- Compendio: SHA-1.
- MAC: HMAC-SHA1.
- Canonización XML: CanonicalXML (sin comentarios).
- Transformada: Firma encapsulada.

Por consiguiente, las implementaciones del SAML V2.0 conformes deben implementar lo anterior.

Además, para facilitar el interfuncionamiento, las implementaciones SAML V2.0 conformes deben implementar lo siguiente:

- Firma: RSAwithSHA1 (recomendada en el documento W3C Signature, que resulta necesaria para efectos del interfuncionamiento).

Aunque la firma XML exige el algoritmo de firma DSAwithSHA1, el SAML V2.0 no lo exige, pero se recomienda.

NOTA – El NIST (National Institute of Standards and Technology) alienta la utilización de SHA-256 (algoritmo de troceo seguro con claves codificadas en 256 bits) en lugar de SHA-1.

13.3.2 Algoritmos de criptación XML

- El documento W3C XML Encryption, 5.2.1 y 5.2.2 exige la aplicación de los siguientes algoritmos: Criptación de bloque: Triplete DES, AES-128, AES-256.
- Transporte de clave: RSA-v1.5, RSA-OAEP.

Por consiguiente, las implementaciones del SAML V2.0 conformes deben implementar estos algoritmos.

13.4 Utilización de TLS 1.0

Cuando el SAML V2.0 emplea TLS 1.0, los servidores deben autenticar a los clientes aplicando un certificado X.509 v3. El cliente ha de establecer la identidad del servidor basándose en el contenido del certificado (por lo general, examinando el campo DN del sujeto del certificado).

13.4.1 Vinculación SOAP y URI del SAML

Las implementaciones con capacidad para aplicar TLS deben implementar el protocolo de cifrado TLS_RSA_WITH_3DES_EDE_CBC_SHA y pueden implementar el protocolo de cifrado TLS_RSA_AES_128_CBC_SHA.

Las implementaciones con capacidad de aplicar FIPS TLS deben implementar el protocolo de cifrado TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA correspondiente y pueden implementar el protocolo de cifrado TLS_RSA_FIPS_AES_128_CBC_SHA correspondiente.

13.4.2 Perfiles SSO de web del SAML

Las implementaciones con capacidad para aplicar TLS deben implementar el protocolo de cifrado TLS_RSA_WITH_3DES_EDE_CBC_SHA (véase RFC 2246 del IETF).

Anexo A

Esquemas del SAML

Este anexo es parte integrante de esta Recomendación y proporciona una lista de los esquemas del SAML necesarios.

A.1 Esquema de la aserción del SAML

A continuación se presenta una lista del esquema de la aserción del SAML.

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-assertion-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New assertion schema for SAML V2.0 namespace.
        </documentation>
    </annotation>
    <attributeGroup name="IDNameQualifiers">
        <attribute name="NameQualifier" type="string" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
    </attributeGroup>
    <element name="BaseID" type="saml:BaseIDAbstractType"/>
    <complexType name="BaseIDAbstractType" abstract="true">
        <attributeGroup ref="saml:IDNameQualifiers"/>
    </complexType>
    <element name="NameID" type="saml:NameIDType"/>
    <complexType name="NameIDType">
        <simpleContent>
            <extension base="string">
                <attributeGroup ref="saml:IDNameQualifiers"/>
                <attribute name="Format" type="anyURI" use="optional"/>
                <attribute name="SPProvidedID" type="string" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="EncryptedElementType">
        <sequence>
            <element ref="xenc:EncryptedData"/>
            <element ref="xenc:EncryptedKey" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="EncryptedID" type="saml:EncryptedElementType"/>
    <element name="Issuer" type="saml:NameIDType"/>
```

```
    <element name="AssertionIDRef" type="NCName"/>
    <element name="AssertionURIRef" type="anyURI"/>
    <element name="Assertion" type="saml:AssertionType"/>
    <complexType name="AssertionType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="saml:Subject" minOccurs="0"/>
            <element ref="saml:Conditions" minOccurs="0"/>
            <element ref="saml:Advice" minOccurs="0"/>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element ref="saml:Statement"/>
                <element ref="saml:AuthnStatement"/>
                <element ref="saml:AuthzDecisionStatement"/>
                <element ref="saml:AttributeStatement"/>
            </choice>
        </sequence>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
    </complexType>
    <element name="Subject" type="saml:SubjectType"/>
    <complexType name="SubjectType">
        <choice>
            <sequence>
                <choice>
                    <element ref="saml:BaseID"/>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
                </choice>
                <element ref="saml:SubjectConfirmation" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
        </choice>
    </complexType>
    <element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
    <complexType name="SubjectConfirmationType">
        <sequence>
            <choice minOccurs="0">
                <element ref="saml:BaseID"/>
                <element ref="saml:NameID"/>
                <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
        </sequence>
        <attribute name="Method" type="anyURI" use="required"/>
    </complexType>
    <element name="SubjectConfirmationData"
type="saml:SubjectConfirmationDataType"/>
    <complexType name="SubjectConfirmationDataType" mixed="true">
        <complexContent>
            <restriction base="anyType">
                <sequence>
                    <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="NotBefore" type="dateTime" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
                <attribute name="Recipient" type="anyURI" use="optional"/>
                <attribute name="InResponseTo" type="NCName" use="optional"/>
                <attribute name="Address" type="string" use="optional"/>
                <anyAttribute namespace="##other" processContents="lax"/>
            </restriction>
        </complexContent>
    </complexType>
    <complexType name="KeyInfoConfirmationDataType" mixed="false">
        <complexContent>
            <restriction base="saml:SubjectConfirmationDataType">
                <sequence>
```

```
                    <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
                </sequence>
            </restriction>
        </complexContent>
    </complexType>
    <element name="Conditions" type="saml:ConditionsType"/>
    <complexType name="ConditionsType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Condition"/>
            <element ref="saml:AudienceRestriction"/>
            <element ref="saml:OneTimeUse"/>
            <element ref="saml:ProxyRestriction"/>
        </choice>
        <attribute name="NotBefore" type="dateTime" use="optional"/>
        <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
    </complexType>
    <element name="Condition" type="saml:ConditionAbstractType"/>
    <complexType name="ConditionAbstractType" abstract="true"/>
    <element name="AudienceRestriction" type="saml:AudienceRestrictionType"/>
    <complexType name="AudienceRestrictionType">
        <complexContent>
            <extension base="saml:ConditionAbstractType">
                <sequence>
                    <element ref="saml:Audience" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Audience" type="anyURI"/>
    <element name="OneTimeUse" type="saml:OneTimeUseType" />
    <complexType name="OneTimeUseType">
        <complexContent>
            <extension base="saml:ConditionAbstractType"/>
        </complexContent>
    </complexType>
    <element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>
    <complexType name="ProxyRestrictionType">
    <complexContent>
        <extension base="saml:ConditionAbstractType">
            <sequence>
                <element ref="saml:Audience" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Count" type="nonNegativeInteger" use="optional"/>
        </extension>
  </complexContent>
    </complexType>
    <element name="Advice" type="saml:AdviceType"/>
    <complexType name="AdviceType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
    </complexType>
    <element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
    <element name="Statement" type="saml:StatementAbstractType"/>
    <complexType name="StatementAbstractType" abstract="true"/>
    <element name="AuthnStatement" type="saml:AuthnStatementType"/>
    <complexType name="AuthnStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:SubjectLocality" minOccurs="0"/>
                    <element ref="saml:AuthnContext"/>
                </sequence>
                <attribute name="AuthnInstant" type="dateTime" use="required"/>
                <attribute name="SessionIndex" type="string" use="optional"/>
```

```
                <attribute name="SessionNotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SubjectLocality" type="saml:SubjectLocalityType"/>
    <complexType name="SubjectLocalityType">
        <attribute name="Address" type="string" use="optional"/>
        <attribute name="DNSName" type="string" use="optional"/>
    </complexType>
    <element name="AuthnContext" type="saml:AuthnContextType"/>
    <complexType name="AuthnContextType">
        <sequence>
            <choice>
                <sequence>
                    <element ref="saml:AuthnContextClassRef"/>
                    <choice minOccurs="0">
                        <element ref="saml:AuthnContextDecl"/>
                        <element ref="saml:AuthnContextDeclRef"/>
                    </choice>
                </sequence>
                <choice>
                    <element ref="saml:AuthnContextDecl"/>
                    <element ref="saml:AuthnContextDeclRef"/>
                </choice>
            </choice>
            <element ref="saml:AuthenticatingAuthority" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AuthnContextClassRef" type="anyURI"/>
    <element name="AuthnContextDeclRef" type="anyURI"/>
    <element name="AuthnContextDecl" type="anyType"/>
    <element name="AuthenticatingAuthority" type="anyURI"/>
    <element name="AuthzDecisionStatement"
type="saml:AuthzDecisionStatementType"/>
    <complexType name="AuthzDecisionStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
                <attribute name="Decision" type="saml:DecisionType"
use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <simpleType name="DecisionType">
        <restriction base="string">
            <enumeration value="Permit"/>
            <enumeration value="Deny"/>
            <enumeration value="Indeterminate"/>
        </restriction>
    </simpleType>
    <element name="Action" type="saml:ActionType"/>
    <complexType name="ActionType">
        <simpleContent>
            <extension base="string">
                <attribute name="Namespace" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <element name="Evidence" type="saml:EvidenceType"/>
    <complexType name="EvidenceType">
        <choice maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
```

```
            <element ref="saml:EncryptedAssertion"/>
        </choice>
    </complexType>
    <element name="AttributeStatement" type="saml:AttributeStatementType"/>
    <complexType name="AttributeStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <choice maxOccurs="unbounded">
                    <element ref="saml:Attribute"/>
                    <element ref="saml:EncryptedAttribute"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
    <element name="Attribute" type="saml:AttributeType"/>
    <complexType name="AttributeType">
        <sequence>
            <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Name" type="string" use="required"/>
        <attribute name="NameFormat" type="anyURI" use="optional"/>
        <attribute name="FriendlyName" type="string" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AttributeValue" type="anyType" nillable="true"/>
    <element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
</schema>
```

A.2 Esquema del contexto de autenticación del SAML

A continuación se presenta el esquema del contexto de autenticación del SAML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0.
          This is just an include of all types from the Shema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schema>
```

A.3 Esquema del contexto de autenticación del SAML AuthenticatedTelephony (Telefonía autenticada)

A continuación se presenta el esquema del contexto de autenticación del SAML relacionado con la telefonía.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
```

```
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.4 Esquema del contexto de autenticación del SAML específico del protocolo Internet (IP)

A continuación se presenta el esquema del contexto de autenticación del SAML específico del IP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:element ref="IPAddress"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

</xs:redefine>

</xs:schema>
```

A.5 Esquema del contexto de autenticación del SAML relativo a la contraseña del protocolo Internet (IPPWord, *Internet protocol password*)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la IPPWord.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
```

```
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.6 Esquema del contexto de autenticación del SAML relativo a Kerberos

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a Kerberos.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
```

```
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

A.7 Esquema del contexto de autenticación del SAML relativo al servicio móvil con un factor registrado (MobileOneFactor-reg)

A continuación se presenta el esquema de la clase de contexto de autenticación del SAML relativo al MobileOneFactorContract registrado.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
```

```
      V2.0 (March, 2005):
        New authentication_u99 context class schema for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:complexType name="AuthnContextDeclarationBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnContextDeclarationBaseType">
        <xs:sequence>
          <xs:element ref="Identification" minOccurs="0"/>
          <xs:element ref="TechnicalProtection" minOccurs="0"/>
          <xs:element ref="OperationalProtection" minOccurs="0"/>
          <xs:element ref="AuthnMethod"/>
          <xs:element ref="GoverningAgreements" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="ZeroKnowledge"/>
            <xs:element ref="SharedSecretChallengeResponse"/>
            <xs:element ref="SharedSecretDynamicPlaintext"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
```

```
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="smartcard"/>
              <xs:enumeration value="MobileDevice"/>
              <xs:enumeration value="MobileAuthCard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:complexContent>
      <xs:restriction base="SecurityAuditType">
        <xs:sequence>
          <xs:element ref="SwitchAudit"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.8 Esquema del contexto de autenticación del SAML relativo al servicio móvil con un factor no registrado (MobileOneFactor-unreg)

A continuación se presenta el esquema de la clase de contexto de autenticación del SAML relativo al MobileOneFactorContract no registrado.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregister
ed"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
```

```
        <xs:element ref="AuthnMethod"/>
        <xs:element ref="GoverningAgreements" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:ID" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthnMethodBaseType">
      <xs:sequence>
        <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
```

```
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="MobileDevice"/>
              <xs:enumeration value="MobileAuthCard"/>
              <xs:enumeration value="smartcard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:complexContent>
      <xs:restriction base="SecurityAuditType">
        <xs:sequence>
          <xs:element ref="SwitchAudit"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="IdentificationType">
    <xs:complexContent>
      <xs:restriction base="IdentificationType">
        <xs:sequence>
          <xs:element ref="GoverningAgreements"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="nym">
```

```
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.9 Esquema del contexto de autenticación del SAML relativo al servicio móvil con dos factores registrados (MobileTwoFactor-reg)

A continuación se presenta el esquema de la clase de contexto de autenticación del SAML relativo al MobileTwoFactorContract registrado.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="ComplexAuthenticator"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:complexContent>
    <xs:restriction base="ComplexAuthenticatorType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
        </xs:choice>
        <xs:element ref="Password"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
```

```
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="KeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="mediumType">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:complexContent>
    <xs:restriction base="IdentificationType">
      <xs:sequence>
        <xs:element ref="PhysicalVerification"/>
        <xs:element ref="WrittenConsent"/>
        <xs:element ref="GoverningAgreements"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

A.10 Esquema del contexto de autenticación del SAML relativo al servicio móvil con dos factores no registrados (MobileTwoFactor-unreg)

A continuación se presenta el esquema de la clase de contexto de autenticación del SAML relativo al MobileTwoFactorUnregistered.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregister
ed"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
```

```
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="ComplexAuthenticator"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:complexContent>
    <xs:restriction base="ComplexAuthenticatorType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
        </xs:choice>
        <xs:element ref="Password"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="KeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="mediumType">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:complexContent>
    <xs:restriction base="IdentificationType">
      <xs:sequence>
        <xs:element ref="GoverningAgreements"/>
```

```
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="nym">
          <xs:simpleType>
            <xs:restriction base="nymType">
              <xs:enumeration value="anonymity"/>
              <xs:enumeration value="pseudonymity"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

</xs:redefine>

</xs:schema>
```

A.11 Esquema del contexto de autenticación del SAML relativo a NomadTelephony (Telefonía nómada)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a NomadTelephony. Nomad telephony indica que el principal se encuentra "itinerando" (roaming) (utilizando probablemente una tarjeta telefónica) y realizando actividades de autenticación mediante el número de línea, un sufijo de usuario y un elemento contraseña.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
```

```
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.12 Esquema del contexto de autenticación del SAML relativo a PersonalizedTelephony (Telefonía personal)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la telefonía personal.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.13 Esquema del contexto de autenticación del SAML relativo a la Privacidad bastante aceptable (PGP)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la PGP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
```

```
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.14 Esquema del contexto de autenticación del SAML relativo al transporte protegido mediante contraseña (PPT)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo al PPT.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTranspor
t"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
```

```
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.15 Esquema del contexto de autenticación del SAML relativo a la contraseña

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la contraseña.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
```

```
        <xs:sequence>
          <xs:element ref="Identification" minOccurs="0"/>
          <xs:element ref="TechnicalProtection" minOccurs="0"/>
          <xs:element ref="OperationalProtection" minOccurs="0"/>
          <xs:element ref="AuthnMethod"/>
          <xs:element ref="GoverningAgreements" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:element ref="RestrictedPassword"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

</xs:redefine>

</xs:schema>
```

A.16 Esquema del contexto de autenticación del SAML relativo a PreviousSession (Sesión anterior)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a PreviousSession. La clase PreviousSession se puede aplicar cuando un principal se ha autenticado ante una autoridad de autenticación en algún momento anterior utilizando un contexto de autenticación aceptable para dicha autoridad.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.17 Esquema del contexto de autenticación del SAML relativo a la Smartcard (Tarjeta inteligente)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la tarjeta inteligente.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
```

```
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.18 Esquema del contexto de autenticación del SAML relativo a la SmartcardPKI (Infraestructura de clave pública para la tarjeta inteligente)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a la SmartcardPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
```

```
      V2.0 (March, 2005):
        New authentication_u99 context class schema for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:complexType name="AuthnContextDeclarationBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnContextDeclarationBaseType">
        <xs:sequence>
          <xs:element ref="Identification" minOccurs="0"/>
          <xs:element ref="TechnicalProtection"/>
          <xs:element ref="OperationalProtection" minOccurs="0"/>
          <xs:element ref="AuthnMethod"/>
          <xs:element ref="GoverningAgreements" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
          </xs:choice>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="Smartcard"/>
          <xs:element ref="ActivationPin"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.19 Esquema del contexto de autenticación del SAML relativo a SoftwarePKI (Infraestructura de clave pública para el software)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a software PKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
```

```
  </xs:annotation>

  <xs:complexType name="AuthnContextDeclarationBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnContextDeclarationBaseType">
        <xs:sequence>
          <xs:element ref="Identification" minOccurs="0"/>
          <xs:element ref="TechnicalProtection"/>
          <xs:element ref="OperationalProtection" minOccurs="0"/>
          <xs:element ref="AuthnMethod"/>
          <xs:element ref="GoverningAgreements" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
          </xs:choice>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="ActivationPin"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
```

```
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.20 Esquema del contexto de autenticación del SAML relativo a la infraestructura de clave pública única (SPKI)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a SPKI. La clase de contexto SPKI indica que la autenticación del principal se llevó a cabo mediante una firma digital y que la clave fue validada con una infraestructura de SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.21 Esquema del contexto de autenticación del SAML relativo a SRP

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a SRP [véase RFC 2945 del IETF].

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.22 Esquema del contexto de autenticación del SAML relativo a telefonía

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a telefonía. Se emplea cuando la autenticación del principal se lleva a cabo a través de un número telefónico de línea fija proporcionado para tal efecto y se transporta mediante un protocolo de telefonía.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
```

```
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:element ref="SubscriberLineNumber"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PSTN"/>
            <xs:element ref="ISDN"/>
            <xs:element ref="ADSL"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

</xs:redefine>

</xs:schema>
```

A.23 Esquema del contexto de autenticación del SAML relativo a la sincronización del tiempo (TimeSync)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a TimeSyncToken. Éste es aplicable cuando la autenticación de un principal se lleva a cabo a través de un testigo de sincronización de tiempo.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
```

```
        <xs:element ref="AuthnMethod"/>
        <xs:element ref="GoverningAgreements" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:ID" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthnMethodBaseType">
      <xs:sequence>
        <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:complexContent>
    <xs:restriction base="PrincipalAuthenticationMechanismType">
      <xs:sequence>
        <xs:element ref="Token"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:complexContent>
    <xs:restriction base="TokenType">
      <xs:sequence>
        <xs:element ref="TimeSyncToken"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TimeSyncTokenType">
  <xs:complexContent>
    <xs:restriction base="TimeSyncTokenType">
      <xs:attribute name="DeviceType" use="required">
        <xs:simpleType>
          <xs:restriction base="DeviceTypeType">
            <xs:enumeration value="hardware"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>

      <xs:attribute name="SeedLength" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="64"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>

      <xs:attribute name="DeviceInHand" use="required">
        <xs:simpleType>
          <xs:restriction base="booleanType">
            <xs:enumeration value="true"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.24 Esquema del contexto de autenticación del SAML relativo a los tipos

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a los tipos.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion_u111 on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between_u97 ? Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that identification has been
        performed in a physical
        face-to-face meeting with the principal and not in an
        online manner.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="credentialLevel">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="primary"/>
            <xs:enumeration value="secondary"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
```

```
  </xs:complexType>
</xs:element>

<xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

<xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characterstics that describe how the
      'secret' (the knowledge or possession
      of which allows the Principal to authenticate to the
      Authentication Authority) is kept secure
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the types and strengths of
      facilities
      of a UA used to protect a shared secret key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the types and strengths of
      facilities
      of a UA used to protect a private key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>The actions that must be performed
      before the private key_u99 can be used. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeySharing" type="KeySharingType">
  <xs:annotation>
    <xs:documentation>Whether or not the private key_u105 is shared
      with the certificate authority.</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyStorage" type="KeyStorageType">
  <xs:annotation>
    <xs:documentation>
      In which medium is the_u107 key stored.
      memory - the key is stored in memory.
      smartcard - the key is_u115 stored in a smartcard.
      token - the key is stored in a hardware token.
      MobileDevice - the key_u105 is stored in a mobile device.
      MobileAuthCard - the key is stored in a mobile
      authentication card.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
<xs:element name="UserSuffix" type="ExtensionOnlyType"/>

<xs:element name="Password" type="PasswordType">
```

```
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a password (or passphrase)
      has been used to
      authenticate the Principal to a remote system.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationPin" type="ActivationPinType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a Pin (Personal
      Identification Number)_u104 has been used to authenticate the Principal
      to some local system in order to activate a key.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Token" type="TokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a hardware or software
      token is used
      as a method of identifying the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TimeSyncToken" type="TimeSyncTokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a time synchronization
      token is used to identify the Principal. hardware -
      the time synchonization
      token has been implemented in hardware. software - the
      time synchronization
      token has been implemented in software. SeedLength -
      the length, in bits, of the
      random seed used in the time synchronization token.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Smartcard" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a smartcard is used to
      identify the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Length" type="LengthType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the minimum and/or maximum
      ASCII length of the password which is enforced (by the UA or the
      IdP). In other words, this is the minimum and/or maximum number of
      ASCII characters required to represent a valid password.
      min - the minimum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
      max - the maximum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimit" type="ActivationLimitType">
  <xs:annotation>
```

```
      <xs:documentation>
        This element indicates_u116 the length of time for which an
        PIN-based authentication is valid.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the
        Principal or auto-supplied by the Authentication Authority.
        principal chosen - the Principal is allowed to choose
        the value of the password. This is true even if
        the initial password is chosen at random by the UA or
        the IdP and the Principal is then free to change
        the password.
        automatic - the password is chosen by the UA or the
        IdP to be cryptographically strong in some sense,
        or to satisfy certain password rules, and that the
        Principal is not free to change it or to choose a new password.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the Authentication
        Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Authenticator" type="AuthenticatorBaseType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
```

```
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PreviousSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system and
        is now re-used (e.g. a_u77 master Secret is used to derive new session
        keys in TLS, SSL, WTLS).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a challenge-response protocol utilizing shared secret
        keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a mechanism which involves the Principal computing a
        digital signature over_u97 at least challenge data provided by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using the
```

```
        local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key and uses it for
        shared secret key agreement with the Authentication Authority (e.g.
        via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted using bare_u72 HTTP utilizing no additional security
        protocols.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="IPSec" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted using a transport mechanism protected by an IPSEC session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
```

```
<xs:element name="WTLS" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted using a transport mechanism protected by a WTLS session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted solely across a mobile network using no additional
      security mechanism.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
<xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

<xs:element name="SSL" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted using a transport mechanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (e.g. liability constraints, obligations, etc.) can be placed.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

<xs:simpleType name="nymType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:complexType name="AuthnContextDeclarationBaseType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0"/>
    <xs:element ref="AuthnMethod" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:sequence>
    <xs:element ref="PhysicalVerification" minOccurs="0"/>
    <xs:element ref="WrittenConsent" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="nym" type="nymType">
    <xs:annotation>
      <xs:documentation>
        This attribute indicates whether or not the
        Identification mechanisms allow the actions of the Principal to be
        linked to an actual end user.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="PrivateKeyProtection"/>
      <xs:element ref="SecretKeyProtection"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit" minOccurs="0"/>
    <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
    <xs:element ref="Authenticator" minOccurs="0"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
```

```
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
  <xs:sequence>
    <xs:element ref="PreviousSession" minOccurs="0"/>
    <xs:element ref="ResumeSession" minOccurs="0"/>
    <xs:element ref="DigSig" minOccurs="0"/>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="ZeroKnowledge" minOccurs="0"/>
    <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
    <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
    <xs:element ref="IPAddress" minOccurs="0"/>
    <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
    <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
    <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
    <xs:element ref="UserSuffix" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="AuthenticatorBaseType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="HTTP"/>
      <xs:element ref="SSL"/>
      <xs:element ref="MobileNetworkNoEncryption"/>
      <xs:element ref="MobileNetworkRadioEncryption"/>
      <xs:element ref="MobileNetworkEndToEndEncryption"/>
      <xs:element ref="WTLS"/>
```

```
        <xs:element ref="IPSec"/>
        <xs:element ref="PSTN"/>
        <xs:element ref="ISDN"/>
        <xs:element ref="ADSL"/>
      </xs:choice>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
        <xs:attribute name="min" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="3"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="max" type="xs:integer" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
```

```
<xs:complexType name="ActivationPinType">
  <xs:sequence>
    <xs:element ref="Length" minOccurs="0"/>
    <xs:element ref="Alphabet" minOccurs="0"/>
    <xs:element ref="Generation" minOccurs="0"/>
    <xs:element ref="ActivationLimit" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
  <xs:attribute name="requiredChars" type="xs:string" use="required"/>
  <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
  <xs:attribute name="case" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DeviceTypeType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="hardware"/>
    <xs:enumeration value="software"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Key Activation Limit is
      defined as a number of_u117 usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

```
  <xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Key Activation Limit is
        the session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="ActivationLimitDurationType">
    <xs:attribute name="duration" type="xs:duration" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitUsagesType">
    <xs:attribute name="number" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitSessionType"/>

  <xs:complexType name="LengthType">
    <xs:attribute name="min" type="xs:integer" use="required"/>
    <xs:attribute name="max" type="xs:integer" use="optional"/>
  </xs:complexType>

  <xs:simpleType name="mediumType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="memory"/>
      <xs:enumeration value="smartcard"/>
      <xs:enumeration value="token"/>
      <xs:enumeration value="MobileDevice"/>
      <xs:enumeration value="MobileAuthCard"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" type="mediumType" use="required"/>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:sequence>
      <xs:element ref="SwitchAudit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExtensionOnlyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Extension" type="ExtensionType"/>

  <xs:complexType name="ExtensionType">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

A.25 Esquema del contexto de autenticación del SAML relativo a X509

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a X.509.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
```

```
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.26 Esquema del contexto de autenticación del SAML relativo a la firma digital XML (XMLDSig)

A continuación se presenta el esquema del contexto de autenticación del SAML relativo a XMLDSig.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="RestrictedPassword"/>
        </xs:sequence>
        <xs:attribute name="preauth" type="xs:integer" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:element ref="DigSig"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PublicKeyType">
    <xs:complexContent>
      <xs:restriction base="PublicKeyType">
        <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

 </xs:redefine>

</xs:schema>
```

A.27 Esquema del SAML relativo al cliente/mandatario mejorado (ECP)

A continuación se presenta el esquema del SAML relativo al perfil de ECP.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
```

```
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>

    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

A.28 Esquema del SAML relativo a los metadatos

A continuación se presenta el esquema del SAML relativo a los metadatos.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for SAML metadata, first published in SAML 2.0.
        </documentation>
    </annotation>
```

```
    <simpleType name="entityIDType">
        <restriction base="anyURI">
            <maxLength value="1024"/>
        </restriction>
    </simpleType>
    <complexType name="localizedNameType">
        <simpleContent>
            <extension base="string">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="localizedURIType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

    <element name="Extensions" type="md:ExtensionsType"/>
    <complexType final="#all" name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="EndpointType">
        <sequence>
            <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Binding" type="anyURI" use="required"/>
        <attribute name="Location" type="anyURI" use="required"/>
        <attribute name="ResponseLocation" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <complexType name="IndexedEndpointType">
        <complexContent>
            <extension base="md:EndpointType">
                <attribute name="index" type="unsignedShort" use="required"/>
                <attribute name="isDefault" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
    <complexType name="EntitiesDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice minOccurs="1" maxOccurs="unbounded">
                <element ref="md:EntityDescriptor"/>
                <element ref="md:EntitiesDescriptor"/>
            </choice>
        </sequence>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="Name" type="string" use="optional"/>
    </complexType>

    <element name="EntityDescriptor" type="md:EntityDescriptorType"/>
    <complexType name="EntityDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice>
```

```
                <choice maxOccurs="unbounded">
                    <element ref="md:RoleDescriptor"/>
                    <element ref="md:IDPSSODescriptor"/>
                    <element ref="md:SPSSODescriptor"/>
                    <element ref="md:AuthnAuthorityDescriptor"/>
                    <element ref="md:AttributeAuthorityDescriptor"/>
                    <element ref="md:PDPDescriptor"/>
                </choice>
                <element ref="md:AffiliationDescriptor"/>
            </choice>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="entityID" type="md:entityIDType" use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <element name="Organization" type="md:OrganizationType"/>
    <complexType name="OrganizationType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:OrganizationName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationDisplayName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
        </sequence>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="OrganizationName" type="md:localizedNameType"/>
    <element name="OrganizationDisplayName" type="md:localizedNameType"/>
    <element name="OrganizationURL" type="md:localizedURIType"/>
    <element name="ContactPerson" type="md:ContactType"/>
    <complexType name="ContactType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:Company" minOccurs="0"/>
            <element ref="md:GivenName" minOccurs="0"/>
            <element ref="md:SurName" minOccurs="0"/>
            <element ref="md:EmailAddress" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:TelephoneNumber" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="contactType" type="md:ContactTypeType" use="required"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="Company" type="string"/>
    <element name="GivenName" type="string"/>
    <element name="SurName" type="string"/>
    <element name="EmailAddress" type="anyURI"/>
    <element name="TelephoneNumber" type="string"/>
    <simpleType name="ContactTypeType">
        <restriction base="string">
            <enumeration value="technical"/>
            <enumeration value="support"/>
            <enumeration value="administrative"/>
            <enumeration value="billing"/>
            <enumeration value="other"/>
        </restriction>
    </simpleType>

    <element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
    <complexType name="AdditionalMetadataLocationType">
        <simpleContent>
            <extension base="anyURI">
                <attribute name="namespace" type="anyURI" use="required"/>
```

```
            </extension>
        </simpleContent>
    </complexType>

    <element name="RoleDescriptor" type="md:RoleDescriptorType"/>
    <complexType name="RoleDescriptorType" abstract="true">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
        <attribute name="errorURL" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <simpleType name="anyURIListType">
        <list itemType="anyURI"/>
    </simpleType>

    <element name="KeyDescriptor" type="md:KeyDescriptorType"/>
    <complexType name="KeyDescriptorType">
        <sequence>
            <element ref="ds:KeyInfo"/>
            <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="use" type="md:KeyTypes" use="optional"/>
    </complexType>
    <simpleType name="KeyTypes">
        <restriction base="string">
            <enumeration value="encryption"/>
            <enumeration value="signing"/>
        </restriction>
    </simpleType>
    <element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>

    <complexType name="SSODescriptorType" abstract="true">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:ArtifactResolutionService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:SingleLogoutService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:ManageNameIDService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
    <element name="SingleLogoutService" type="md:EndpointType"/>
    <element name="ManageNameIDService" type="md:EndpointType"/>
    <element name="NameIDFormat" type="anyURI"/>

    <element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
    <complexType name="IDPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:SingleSignOnService" maxOccurs="unbounded"/>
```

```
                    <element ref="md:NameIDMappingService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="WantAuthnRequestsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SingleSignOnService" type="md:EndpointType"/>
    <element name="NameIDMappingService" type="md:EndpointType"/>
    <element name="AssertionIDRequestService" type="md:EndpointType"/>
    <element name="AttributeProfile" type="anyURI"/>

    <element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
    <complexType name="SPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:AssertionConsumerService"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeConsumingService" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="AuthnRequestsSigned" type="boolean"
use="optional"/>
                <attribute name="WantAssertionsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
    <element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
    <complexType name="AttributeConsumingServiceType">
        <sequence>
            <element ref="md:ServiceName" maxOccurs="unbounded"/>
            <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="index" type="unsignedShort" use="required"/>
        <attribute name="isDefault" type="boolean" use="optional"/>
    </complexType>
    <element name="ServiceName" type="md:localizedNameType"/>
    <element name="ServiceDescription" type="md:localizedNameType"/>
    <element name="RequestedAttribute" type="md:RequestedAttributeType"/>
    <complexType name="RequestedAttributeType">
        <complexContent>
            <extension base="saml:AttributeType">
                <attribute name="isRequired" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
    <complexType name="AuthnAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AuthnQueryService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnQueryService" type="md:EndpointType"/>

    <element name="PDPDescriptor" type="md:PDPDescriptorType"/>
    <complexType name="PDPDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AuthzService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzService" type="md:EndpointType"/>

    <element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
    <complexType name="AttributeAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AttributeService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AttributeService" type="md:EndpointType"/>

    <element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
    <complexType name="AffiliationDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AffiliateMember" type="md:entityIDType"/>
</schema>
```

A.29 Esquema del SAML relativo al protocolo

A continuación se presenta el esquema del SAML relativo al protocolo.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-protocol-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard_u83 schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New protocol schema based in a SAML V2.0 namespace.
     </documentation>
    </annotation>
    <complexType name="RequestAbstractType" abstract="true">
        <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
        </sequence>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
        <attribute name="Destination" type="anyURI" use="optional"/>
         <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
    <element name="Extensions" type="samlp:ExtensionsType"/>
    <complexType name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="StatusResponseType">
         <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
            <element ref="samlp:Status"/>
         </sequence>
         <attribute name="ID" type="ID" use="required"/>
         <attribute name="InResponseTo" type="NCName" use="optional"/>
         <attribute name="Version" type="string" use="required"/>
         <attribute name="IssueInstant" type="dateTime" use="required"/>
         <attribute name="Destination" type="anyURI" use="optional"/>
         <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
    <element name="Status" type="samlp:StatusType"/>
    <complexType name="StatusType">
        <sequence>
            <element ref="samlp:StatusCode"/>
```

```
            <element ref="samlp:StatusMessage" minOccurs="0"/>
            <element ref="samlp:StatusDetail" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="StatusCode" type="samlp:StatusCodeType"/>
    <complexType name="StatusCodeType">
        <sequence>
            <element ref="samlp:StatusCode" minOccurs="0"/>
        </sequence>
        <attribute name="Value" type="anyURI" use="required"/>
    </complexType>
    <element name="StatusMessage" type="string"/>
    <element name="StatusDetail" type="samlp:StatusDetailType"/>
    <complexType name="StatusDetailType">
        <sequence>
            <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>
    <complexType name="AssertionIDRequestType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:AssertionIDRef" maxOccurs="unbounded"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>
    <complexType name="SubjectQueryAbstractType" abstract="true">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="AuthnQuery" type="samlp:AuthnQueryType"/>
    <complexType name="AuthnQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                </sequence>
                <attribute name="SessionIndex" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="RequestedAuthnContext"
type="samlp:RequestedAuthnContextType"/>
    <complexType name="RequestedAuthnContextType">
        <choice>
            <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
            <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
        </choice>
        <attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
    </complexType>
    <simpleType name="AuthnContextComparisonType">
        <restriction base="string">
            <enumeration value="exact"/>
            <enumeration value="minimum"/>
            <enumeration value="maximum"/>
            <enumeration value="better"/>
        </restriction>
    </simpleType>
    <element name="AttributeQuery" type="samlp:AttributeQueryType"/>
    <complexType name="AttributeQueryType">
```

```
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>
    <complexType name="AuthzDecisionQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnRequest" type="samlp:AuthnRequestType"/>
    <complexType name="AuthnRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject" minOccurs="0"/>
                    <element ref="samlp:NameIDPolicy" minOccurs="0"/>
                    <element ref="saml:Conditions" minOccurs="0"/>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                    <element ref="samlp:Scoping" minOccurs="0"/>
                </sequence>
                <attribute name="ForceAuthn" type="boolean" use="optional"/>
                <attribute name="IsPassive" type="boolean" use="optional"/>
                <attribute name="ProtocolBinding" type="anyURI" use="optional"/>
                <attribute name="AssertionConsumerServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="optional"/>
                <attribute name="AttributeConsumingServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="ProviderName" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
    <complexType name="NameIDPolicyType">
        <attribute name="Format" type="anyURI" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
        <attribute name="AllowCreate" type="boolean" use="optional"/>
    </complexType>
    <element name="Scoping" type="samlp:ScopingType"/>
    <complexType name="ScopingType">
        <sequence>
            <element ref="samlp:IDPList" minOccurs="0"/>
            <element ref="samlp:RequesterID" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ProxyCount" type="nonNegativeInteger" use="optional"/>
    </complexType>
    <element name="RequesterID" type="anyURI"/>
    <element name="IDPList" type="samlp:IDPListType"/>
    <complexType name="IDPListType">
        <sequence>
            <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
            <element ref="samlp:GetComplete" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="IDPEntry" type="samlp:IDPEntryType"/>
    <complexType name="IDPEntryType">
```

```
        <attribute name="ProviderID" type="anyURI" use="required"/>
        <attribute name="Name" type="string" use="optional"/>
        <attribute name="Loc" type="anyURI" use="optional"/>
    </complexType>
    <element name="GetComplete" type="anyURI"/>
    <element name="Response" type="samlp:ResponseType"/>
    <complexType name="ResponseType">
         <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element ref="saml:Assertion"/>
                    <element ref="saml:EncryptedAssertion"/>
                </choice>
            </extension>
         </complexContent>
    </complexType>
    <element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>
    <complexType name="ArtifactResolveType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="samlp:Artifact"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="Artifact" type="string"/>
    <element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>
    <complexType name="ArtifactResponseType">
         <complexContent>
            <extension base="samlp:StatusResponseType">
                <sequence>
                    <any namespace="##any" processContents="lax" minOccurs="0"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="ManageNameIDRequest" type="samlp:ManageNameIDRequestType"/>
    <complexType name="ManageNameIDRequestType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <choice>
                        <element ref="samlp:NewID"/>
                        <element ref="samlp:NewEncryptedID"/>
                        <element ref="samlp:Terminate"/>
                    </choice>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="NewID" type="string"/>
    <element name="NewEncryptedID" type="saml:EncryptedElementType"/>
    <element name="Terminate" type="samlp:TerminateType"/>
    <complexType name="TerminateType"/>
    <element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
    <element name="LogoutRequest" type="samlp:LogoutRequestType"/>
    <complexType name="LogoutRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
```

```
                    <element ref="samlp:SessionIndex" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="Reason" type="string" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SessionIndex" type="string"/>
    <element name="LogoutResponse" type="samlp:StatusResponseType"/>
    <element name="NameIDMappingRequest" type="samlp:NameIDMappingRequestType"/>
    <complexType name="NameIDMappingRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:NameIDPolicy"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDMappingResponse"
type="samlp:NameIDMappingResponseType"/>
    <complexType name="NameIDMappingResponseType">
        <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

A.30 Esquema del SAML relativo a X500

A continuación se presenta el esquema del SAML relativo a X.500.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

A.31 Esquema del SAML relativo a XACML

A continuación se presenta la lista del esquema del SAML relativo a XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
              Custom schema for XACML attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

Apéndice I

Consideraciones relativas a la seguridad y la privacidad

La seguridad y la privacidad deben abordarse de una manera sistemática, considerando las cuestiones humanas tales como los ataques por manipulación social, las cuestiones de política, la gestión de claves y de confianza, la implementación de la seguridad y otros factores que están fuera del alcance de este apéndice. Las soluciones técnicas de la seguridad implican un costo y por lo tanto también deben tenerse en cuenta las alternativas en cuanto a los requisitos y la política, así como los requisitos jurídicos y de reglamentación.

En este apéndice se resumen las cuestiones y los enfoques de seguridad generales así como las amenazas y las medidas preventivas para la utilización de aserciones, protocolos, vinculaciones y perfiles del SAML de una manera segura que permita mantener la privacidad. Asimismo, se describen y analizan las propiedades de la seguridad y la privacidad del SAML. El objetivo es proporcionar información a los arquitectos e implementadores de sistemas basados en el SAML acerca de lo siguiente:

- Las cuestiones sobre privacidad que han de tenerse en cuenta y la manera en que la arquitectura del SAML las aborda.
- Las amenazas y por consecuencia los riesgos de la seguridad, a los que está supeditado un sistema basado en el SAML.
- Los riesgos de la seguridad que aborda la arquitectura del SAML, y cómo lo lleva a cabo.
- Los riesgos de la seguridad que no aborda la arquitectura del SAML.
- Recomendaciones acerca de las medidas preventivas que pueden aplicarse para mitigar esos riesgos de la seguridad.

I.1 Privacidad

El SAML incluye la capacidad de expedir enunciados acerca de los atributos y autorizaciones de las entidades autenticadas. Hay muchas situaciones comunes en las que una o más partes en una comunicación desearían mantener la información que se transporta en estos enunciados accesible a un grupo de entidades tan restringido como sea posible. Los enunciados relativos a atributos médicos o financieros representan ejemplos simples de estos casos.

Cuando se despliega un sistema basado en el SAML deben tenerse en cuenta las leyes y reglamentos en materia de privacidad que existen en muchos países y jurisdicciones. Las partes que crean enunciados y que expiden, transportan y consumen aserciones, tienen que estar conscientes de estas preocupaciones potenciales acerca de la privacidad y deberían tratar de considerarlas en sus implementaciones de sistemas con capacidad de manejo del SAML.

I.2 Confidencialidad

El aspecto más importante que se ha de considerar al asegurar la privacidad de las partes en una transacción habilitada para SAML es probablemente la capacidad de transportar la transacción con una garantía de confidencialidad. En otras palabras, ¿es posible transportar la información contenida en una aserción del emisor a la audiencia objetivo, y sólo a ella, sin que otras partes tengan acceso a dicha información?

Desde el punto de vista técnico es posible transportar la información confidencialmente. Todas las partes en las transacciones habilitadas para el SAML deberían analizar cada uno de sus pasos en la interacción (y en cualquier utilización subsiguiente de los datos obtenidos de las transacciones) a fin de asegurar que se mantiene la confidencialidad de la información que así debería mantenerse.

Cabe señalar también que el simple oscurecimiento del contenido de las aserciones puede no resultar en una protección adecuada de la privacidad. Hay muchos casos en los que la simple disponibilidad de la información de que un determinado usuario (o dirección IP) accede a un servicio específico puede constituir una violación de la privacidad (por ejemplo, la información de que un usuario tuvo acceso a una instalación de prueba médica para obtener una aserción puede bastar para infringir la privacidad, aun sin conocer el contenido de la aserción). Diversas técnicas relativas a la interacción anónima están disponibles para solucionar parcialmente estos problemas, como se describe en las cláusulas siguientes.

I.3 Seudoanonimidad y anonimidad

No existen definiciones de la anonimidad que puedan satisfacer todos los casos. Muchas de ellas tratan el caso simple de un emisor y un mensaje, y examinan la "anonimidad" en virtud de la imposibilidad de establecer un vínculo entre un emisor determinado y un mensaje enviado, o un mensaje devuelto a un emisor. Aunque esa definición es adecuada para el caso "único en su género" (*one off*), ésta ignora la agregación de información que puede estar disponible con el tiempo basada en el comportamiento y no en un identificador.

En el sistema SAML resulta útil reflexionar acerca de la anonimidad como si estuviera "dentro de un conjunto". Esta noción es importante para el SAML debido a la utilización de autoridades. Aunque se trate de un sujeto "anónimo", es posible identificarlo como miembro del conjunto de sujetos del dominio de la autoridad pertinente. Los sistemas habilitados para el SAML están limitados, en el mejor de los casos, a la "anonimidad parcial" debido a la intervención de autoridades. Una entidad para la que se expide una aserción ya puede ser identificada como parte del grupo de entidades en una relación con la autoridad expedidora.

Las limitaciones relativas a la anonimidad pueden ser mucho más estrictas que la simple asociación de autoridad, dependiendo de cómo se emplean los identificadores, ya que la reutilización de los identificadores seudónimos facilita la acumulación de información de identificación potencial. Además, los usuarios de los sistemas con capacidad para el SAML también pueden empeorar la infracción de la anonimidad a través de sus acciones.

Aparte de la identidad legal, cualquier identificador de un sujeto puede considerarse como un seudónimo. Incluso las nociones como "titular de clave" pueden considerarse equivalentes a un seudónimo al vincular una acción (o un conjunto de acciones) a un sujeto. Incluso una descripción como "el usuario que acaba de solicitar acceso al objeto XYZ en el instante 23:34" puede servir como equivalente de un seudónimo.

Por consiguiente, con relación a la "aptitud para provocar un daño" no hay diferencia si el usuario se describe mediante un identificador o un comportamiento (por ejemplo, el uso de una clave o el desempeño de una acción).

Lo que establece una diferencia es la frecuencia con la que se emplea el equivalente particular de un seudónimo. La anonimidad ofrece una clasificación de los seudónimos comenzando por los seudónimos personales (como los sobrenombres) que se emplean continuamente, pasando por los diversos tipos de seudónimos de cometido (como el Secretario de la Defensa), hasta llegar a los seudónimos "que se emplean una sola vez".

Únicamente los seudónimos que se emplean una sola vez pueden ofrecer anonimidad (dentro del SAML, debe considerarse como "anonimidad dentro de un conjunto"). Sin embargo, mientras más se emplee un seudónimo determinado, mayor será el riesgo para la anonimidad. En otras palabras, la reutilización de un seudónimo facilita asociar la información de identificación potencial adicional con el seudónimo. Al paso del tiempo, esto conduce a una acumulación de información que permite identificar de manera inequívoca la identidad asociada con un seudónimo.

Las autoridades del sitio de origen (como las autoridades de autenticación y de atributo) pueden ofrecer un grado de "anonimidad parcial" mediante el empleo de identificadores o claves que se emplean una sola vez (para el caso "titular de clave"). Esta anonimidad es "parcial" en el mejor de los casos, porque el usuario está confinado necesariamente al conjunto de sujetos en una relación con la autoridad. Este conjunto puede reducirse aun más (reduciendo más la anonimidad) cuando se emplean atributos de agregación que crean subconjuntos de la comunidad de usuarios en el sitio de origen. Los usuarios que se preocupan realmente por la anonimidad deben encargarse de disfrazar o evitar patrones de comportamiento poco usuales que pudieran servir para "anular su anonimidad" con el paso del tiempo.

I.4 Seguridad

En las siguientes cláusulas se examinan las consideraciones relativas a la seguridad.

I.4.1 Contexto

Las comunicaciones entre los sistemas basados en ordenadores están supeditadas a una diversidad de amenazas que conllevan algún nivel de riesgo asociado. La naturaleza del riesgo depende de una gran cantidad de factores, incluyendo la naturaleza de las comunicaciones y de los sistemas de comunicación, los medios de comunicaciones, el entorno de comunicación, los entornos del sistema de extremo y otros.

El SAML tiene por objetivo apoyar a los instaladores a establecer contextos de seguridad para las comunicaciones basadas en ordenadores en el nivel de aplicación, dentro de los dominios de seguridad o entre éstos. Con este cometido, el SAML transfiere datos de autenticación para soportar la capacidad del sistema de extremo a protegerse contra la utilización no autorizada de la información. La seguridad de las comunicaciones puede aplicarse directamente al diseño del SAML. La seguridad de los sistemas es importante principalmente en el contexto de los modelos de amenazas del SAML.

I.4.2 Alcance

Algunos campos que repercuten ampliamente en la seguridad general de un sistema que emplea el SAML quedan explícitamente fuera del alcance del SAML. Si bien es cierto que esta Recomendación no aborda esos campos, éstos deberían tenerse en cuenta cuando se examine la seguridad de un sistema. En particular, estas cuestiones son importantes, pero hoy en día están fuera del alcance del SAML:

- Autenticación inicial: El SAML facilita la creación de enunciados acerca de los actos de autenticación que han tenido lugar, pero no incluye los requisitos o especificaciones correspondientes a tales actos. Los consumidores de las aserciones de seguridad deberían ser muy precavidos y no confiar a ciegas en estas aserciones a menos que, o hasta que conozcan las condiciones en las que fueron creadas. La confianza en las aserciones no debe sobrepasar la confianza a la que ha llegado la parte afirmadora en cuanto a las conclusiones afirmadas.
- Modelo de confianza: En muchos casos, la seguridad de una conversación en el SAML dependerá del modelo de confianza subyacente, que por lo general se basa en una infraestructura de gestión de claves (por ejemplo, PKI o clave secreta). Por ejemplo, los mensajes SOAP asegurados mediante una firma XML se consideran verdaderamente asegurados en la medida en que se pueda confiar en las claves utilizadas durante el intercambio. Las claves violadas o los certificados revocados no detectados podrían propiciar, por ejemplo, una infracción de la seguridad. Incluso cuando no se exige un certificado se da la oportunidad para cometer ataques por suplantación de persona. La configuración de una PKI no es trivial y debe implementarse correctamente de manera que las capas construidas por encima de ella (como las partes del SAML) estén aseguradas.

Es necesario disponer de implementaciones de protocolos de seguridad adecuadas, incluyendo la generación de números aleatoria o seudoaleatoria y el almacenamiento seguro de las claves.

I.4.3 Modelo de amenazas del SAML

El modelo general de amenazas de Internet que se describe en las directrices relativas a las consideraciones de seguridad del IETF constituye la base para el modelo de amenazas del SAML. Supongamos que dos o más puntos extremo de una transacción del SAML no se encuentran en peligro, pero que el atacante tiene el control total del canal de comunicaciones.

Además, debido a la naturaleza del SAML como un protocolo de enunciados de autenticación y autorización con múltiples partes, tendrán que considerarse los casos en los que una o más partes en una transacción del SAML legítima – que funciona legítimamente en su cometido para esa transacción – tratan de utilizar malintencionadamente información obtenida de una transacción anterior en una transacción subsiguiente.

En los siguientes casos se describen posibles ataques:

- **Colusión (*collusion*)**: Colaboración en secreto entre dos o más entidades del sistema para lanzar un ataque, por ejemplo:
  - colusión entre el principal y el proveedor de servicio;
  - colusión entre el principal y el proveedor de identidad;
  - colusión entre el proveedor de identidad y el proveedor de servicio;
  - colusión entre dos o más principales;
  - colusión entre dos o más proveedores de servicio;
  - colusión entre dos o más proveedores de identidad.
- **Ataques por denegación de servicio (*denial-of-service attacks*)**: Impedimento del acceso autorizado a un recurso del sistema o retraso de las operaciones y funciones del sistema.
- **Intromisión (*man-in-the-middle attacks*)**: Ataque por intercepción activa en el que el atacante intercepta y modifica selectivamente los datos comunicados para hacerse pasar por una o varias de las entidades que participan en una asociación de comunicación.
- **Ataques por repetición (*replay attacks*)**: Ataque en el que se repite una transmisión de datos válida de modo malintencionado o fraudulento, bien sea por el originador o por un adversario que intercepta los datos y los retransmite, posiblemente como parte de un ataque por impostura.
- **Secuestro de sesión (*session hijacking*)**: Ataque por intercepción en el que el atacante toma el control de una asociación de comunicación establecida con anterioridad.

En todos los casos, los mecanismos locales que aplicarán los sistemas para decidir si generan aserciones o no, quedan fuera del alcance de esta Recomendación. Por lo tanto, las amenazas que surgen de los detalles del inicio de sesión original en una autoridad de autenticación, por ejemplo, también quedan fuera del alcance de este documento. Si una

autoridad expide una aserción falsa, las amenazas que surgen del aprovechamiento de esa aserción por parte de los sistemas en sentido descendente quedan explícitamente fuera del alcance de este documento.

La consecuencia directa de lo anterior es que la seguridad de un sistema basado en aserciones como entradas será sólo tan buena como la seguridad del sistema empleado para generar esas aserciones y la exactitud de los datos y el procesamiento en los que se basaron las aserciones generadas. Cuando se determina cuáles son los emisores en los que se puede confiar, particularmente en los casos donde las aserciones se utilizarán como entradas a las decisiones de autenticación o autorización, el riesgo de los compromisos de seguridad que surgen del consumo de aserciones falsas pero expedidas con legitimidad, es muy grande. Las políticas de confianza entre las partes afirmante y confiante deberían redactarse siempre incluyendo una consideración significativa de responsabilidad, y las implementaciones deberían ofrecer una vía de auditoría apropiada.

I.5 Técnicas de seguridad

En las siguientes cláusulas se describen las técnicas de seguridad y diversas tecnologías normalizadas que están disponibles para su implementación en las instalaciones del SAML.

I.5.1 Autenticación

En esta Recomendación, autenticación significa la capacidad de una parte en una transacción para determinar la identidad de la otra parte. La autenticación puede ser en un solo sentido o bidireccional.

- **Sesión activa (*active session*)**: El canal de comunicaciones que se utiliza para transportar un mensaje del SAML proporciona autenticación no persistente. La autenticación puede ser unilateral – del iniciador de la sesión al receptor – o bilateral. El protocolo de comunicaciones que se emplee determinará el método específico. Por ejemplo, la aplicación de un protocolo de red seguro como el protocolo TLS o el de seguridad IP, habilita al emisor del mensaje SAML con la capacidad de autenticar el destino del entorno TCP/IP.
- **Nivel del mensaje (*message-level*)**: En los documentos W3C XML Signature y OASIS WSS se proponen métodos para una "autenticación" persistente que está firmemente acoplada a un documento. Este método no garantiza de modo independiente que el emisor del mensaje es en realidad ese firmante (y por cierto, en muchos casos en los que intervienen intermediarios, explícitamente este no es el caso). Para satisfacer este requisito basta cualquier método que permita la confirmación persistente de la participación de una entidad resoluble única con un subconjunto de un mensaje XML determinado.

I.5.2 Confidencialidad

Significa que únicamente los destinatarios deseados podrán leer el contenido del mensaje y no cualquiera que encuentre el mensaje.

- **En tránsito (*in transit*)**: La aplicación de un protocolo de red seguro como el protocolo TLS o el de seguridad IP proporciona la confidencialidad de un mensaje cuando se transfiere entre dos nodos.
- **Nivel del mensaje (*message-level*)**: La criptación XML permite criptar selectivamente documentos XML. Este método proporciona una confidencialidad permanente y selectiva de los elementos de un mensaje XML.

I.5.3 Integridad de los datos

La integridad de los datos es la capacidad de confirmar que un determinado mensaje tal como ha sido recibido es una versión sin alterar del que fue enviado.

- **En tránsito**: La aplicación de un protocolo de red seguro como el protocolo TLS o el de seguridad IP se puede configurar para proporcionar la protección de la integridad de los paquetes transmitidos a través de una conexión de red.
- **Nivel del mensaje**: La firma XML proporciona un método para crear una garantía persistente de la naturaleza inalterada de un mensaje que está íntimamente ligada a dicho mensaje. Un subconjunto persistente determinado de un mensaje XML es suficiente para cumplir este requisito.

I.5.4 Observaciones acerca de la gestión de claves

En muchas partes de este apéndice se hará referencia a la capacidad de los sistemas para la autenticación, la integridad de los datos y la confidencialidad, a través de diversos esquemas basados en la firma y la criptación. Para todos estos esquemas la seguridad suministrada por cada uno de ellos es limitada, sobre la base de los sistemas de gestión de claves con que se cuenta. Algunas de las limitaciones particulares son:

1) **Acceso a la clave**: Se supone que, si se van a emplear sistemas basados en claves para la autenticación, la integridad de los datos y el no repudio, la seguridad permitirá garantizar que terceras partes inadecuadas no puedan acceder a claves privadas o secretas de un principal. Por ejemplo, una firma digital creada con la clave privada de Bob sólo puede probar la participación de éste en la medida en que él es el único que puede acceder a dicha clave. En general, se debería limitar el acceso a las claves al mínimo conjunto de entidades posible (algo especialmente importante en el caso de claves empresariales o de organizaciones) y conviene protegerlo utilizando frases de contraseña y otros medios. Se aplican las precauciones normales de seguridad (no se debe guardar por escrito la contraseña, si se aleja del computador no deje una ventana abierta con la clave de acceso, etc.).

2) **Vinculación de la identidad a la clave**: Con el fin de poder utilizar para la autenticación un sistema basado en claves, ha de existir algún vínculo fiable entre la identidad y la clave. Si bien la verificación de la firma digital de un documento permite establecer si el documento ha sido alterado desde la firma de dicha clave y si fue firmado realmente por cierta clave, no confirma que la clave empleada sea realmente la de un individuo determinado, ni que éste sea el apropiado para el momento y el propósito dados. La verificación de la vinculación entre la identidad y la clave requiere una validación adicional.

   Es necesario establecer el vínculo entre la clave y el individuo. Algunas de las soluciones más empleadas incluyen directorios locales en los que se almacenan tanto los identificadores como las claves – algo fácil de entender pero difícil de mantener – o la utilización de certificados. Esta última proporciona un método escalable para hacer corresponder una clave con una identidad, pero requiere mecanismos para gestionar la el periodo de validez de certificados y los cambios de estado de la vinculación (por ejemplo, un empleado que deja de trabajar para una empresa ya no dispone de la identidad empresarial). Se suele utilizar una infraestructura de clave pública (PKI, *public key infrastructure*).

   En este caso, se identifica un conjunto de Autoridades de certificación raíz (CA) para cada consumidor de firmas – con el fin de responder a la pregunta "¿En quién confiar cuando se hacen declaraciones acerca de la vinculación identidad-clave?". La verificación de una firma se convierte entonces en un proceso en el que primero se verifica dicha firma (estableciendo de esta manera si la firma fue hecha con la clave en cuestión y si el mensaje no ha sido modificado), luego se valida la cadena de certificados (para determinar que la clave está vinculada a la identidad correcta) y entonces se confirma que el vínculo sigue siendo el adecuado. Esta última parte requiere varias etapas, necesarias para garantizar que un vínculo está en vigor – aunque los certificados suelen tener una "vida útil" incorporada, si una clave se ve amenazada durante dicho periodo de validez del certificado la vinculación clave-identidad contenida en el certificado pierde validez, aunque el certificado siga estando marcado como en vigor. Del mismo modo, con frecuencia los certificados dependen de asociaciones que terminan antes que expire su vida útil (por ejemplo, algunos certificados pierden su validez cuando alguien cambia de empleo, etc.). Como resultado, un buen sistema de gestión de claves es bastante sólido pero muy complejo. En el fondo, un proceso de verificación de firma se resume en la verificación de los vínculos documento-clave y clave-identidad, así como en el control de la validez de la clave y del certificado.

I.5.5 Códigos de cifrado TLS

En muchas partes de esta Recomendación se recomienda enfáticamente la utilización del HTTP sobre SSL 3.0 (véase el apéndice IV) o sobre el TLS 1.0, o de los URL con el esquema URL HTTPS.

Salvo si se indica lo contrario, cuando las vinculaciones de SAML utilizan SSL 3.0 o TLS 1.0 los servidores deben autenticarse con los clientes a través de certificados X.509 v3. El cliente ha de establecer la identidad del servidor sobre la base del contenido del certificado (normalmente a través del análisis de su campo DN de sujeto).

Es posible configurar el SSL/TLS para que emplee varios protocolos de cifrado diferentes, no todos apropiados para garantizar una seguridad conforme a las "prácticas idóneas". Un protocolo de cifrado combina cuatro tipos de características de seguridad y recibe un nombre en [SSL]. Antes de que se transmitan datos a través de una conexión SSL, ambos extremos tratan de negociar un protocolo de cifrado, con lo cual pueden determinar una calidad de protección apropiada para sus comunicaciones, teniendo en cuenta las restricciones de las combinaciones de mecanismos de que se dispone. Las características de seguridad correspondientes a un protocolo de cifrado son:

En SSL se definen varios algoritmos de intercambio de claves. En algunos mecanismos se cuenta con la autenticación de servidor. Sin embargo, también se soportan mecanismos de intercambio anónimo de claves (los cuales corren el riesgo de ataques del tipo "intromisión", por lo que no se recomiendan en el contexto del SAML). El algoritmo más utilizado actualmente es el de intercambio de clave autenticada "RSA" (la patente del algoritmo RSA ya no es válida).

Otro algoritmo importante de intercambio de claves es el Diffie-Hellman autenticado, "DHE_DSS", cuya utilización no tiene restricciones relacionadas con patentes.

Si bien el algoritmo de intercambio de claves se puede exportar libremente desde los EE.UU., los algoritmos exportables deben emplear claves públicas cortas (512 bits) para el intercambio de claves y claves simétricas cortas (40 bits) para la criptación. Ya se han presentado ataques exitosos contra claves de dichas longitudes por lo cual no se recomienda su utilización.

El algoritmo de criptación más rápido es el de cifrado de tren RC4; en el modo de "encadenamiento de bloques cifrados" (CBC, *cipher block chaining*) se soportan también el DES y sus variantes (DES40, 3DES-EDE), al igual que el AES. También se soportan otros modos, véanse las referencias relativas al TLS.

En algunos protocolos de cifrado se cuenta con la opción criptación nula, en la que no se efectúa ninguna criptación, y en cuyo caso se utiliza el SSL/TLS solamente para autenticar y a los efectos de protección de la integridad. Los protocolos de cifrado que tienen criptación nula no proporcionan confidencialidad y no deben ser utilizados en casos en los que ésta sea un requisito y no se obtenga a través de medios diferentes al SSL/TLS.

El algoritmo de resumen de mensaje que se utiliza para el código de autenticación de mensaje. Recientemente la FCC recomendó emplear el SHA-256 y el IETF decidió seguir dicha recomendación.

I.6 Consideraciones generales de seguridad en el SAML

En las cláusulas siguientes se analizan los riesgos que atañen a la seguridad cuando se utiliza y se implementa el SAML, y se describen las medidas paliativas del caso.

I.6.1 Aserciones SAML

En el nivel de la aserción SAML propiamente dicho, es poco lo que hay que decir acerca de problemas de seguridad – la mayoría de ellos aparecen durante las comunicaciones en el marco del protocolo de petición/respuesta o cuando se intenta utilizar el SAML a través de alguna de las vinculaciones. Siempre se espera, desde luego, que el cliente respete el intervalo de validez de la aserción y todos los elementos `<OneTimeUse>` presentes en ella.

Con todo, cabe analizar un aspecto del nivel de aserción, a saber, una de éstas, una vez emitida, queda fuera del control de quien la produce, lo cual tiene varias implicaciones. Por ejemplo, quien la emite no tiene control sobre cuánto tiempo persista la aserción en los sistemas del cliente, ni sobre las partes con las que el cliente comparta la información de aserción. La principal preocupación se centra en que un posible atacante malintencionado pueda ver los contenidos de las aserciones que pasan sin encriptar (o insuficientemente criptadas) por la línea.

Si bien se han realizado esfuerzos considerables en la Recomendación sobre el SAML para resolver estos problemas, nada de lo que aquí se diga reemplaza la necesidad de prestar mucha atención a lo que se ponga en una aserción. Conviene que quienes producen aserciones consideren siempre las posibles consecuencias que podrían derivarse de que la información, almacenada en un sitio distante, sea directamente utilizada de una manera inconveniente, expuesta a posibles piratas informáticos o almacenada a fines de utilizaciones fraudulentas más creativas. Asimismo, deben tener en cuenta la posibilidad de que la información contenida en la aserción sea compartida con otras partes e inclusive hecha pública, bien sea intencionalmente o sin pretenderlo.

I.6.2 Protocolo SAML

En esta cláusula se describen aspectos de seguridad relativos al protocolo de petición-respuesta SAML, aparte de algunas amenazas resultantes de la utilización de ciertas vinculaciones de protocolo.

– **Denegación de servicio**

El protocolo SAML es susceptible de sufrir un ataque del tipo denegación de servicio (DoS, *denial of service*). El procesamiento de una petición SAML puede ser bastante costoso, pues incluye el análisis sintáctico del mensaje de petición (algo que suele implicar la construcción de un árbol DOM), la búsqueda de almacenamiento de aserción o base de datos (probablemente sobre una clave sin indexar), la construcción de un mensaje de respuesta y tal vez una o varias operaciones de firma digital. En otras palabras, el esfuerzo que requiere un atacante que genera peticiones es mucho menor que el necesario para procesar dichas peticiones.

1) Requisito de autenticación de cliente a un nivel inferior

El hacer que los clientes tengan que autenticarse en algún nivel inferior al del protocolo SAML (por ejemplo, mediante el SOAP sobre la vinculación HTTP, con HTTP sobre el TLS/SSL, y requiriendo que los certificados en el lado del cliente provengan de una autoridad de certificación fiable en su raíz) proporcionará la rastreabilidad en el caso de ataques DoS.

La autenticación, cuando se emplea solamente para garantizar la rastreabilidad, no sirve para evitar el ataque propiamente dicho, pero sí permite lograr un efecto disuasivo.

Si se utiliza la autenticación junto con algún sistema de control de acceso, se bloquean realmente los ataques DoS desde fuera (puede ocurrir, no obstante, que una sobrecarga del esquema de autenticación de cliente siga siendo equivalente a un ataque de denegación de servicio contra el servicio SAML, aunque este ataque se debe enfrentar en el contexto del esquema de autenticación de cliente que se haya escogido).

No importa cuál sistema de autenticación de cliente se utilice, éste debería proporcionar la capacidad de establecer un sólo remitente por cada petición, y no debería estar expuesto a ningún fraude (por ejemplo, en el caso en que exista solamente la función de rastreabilidad, no basta con introducir la dirección IP ya que es muy fácil suplantar dicha información).

2) Requisito de firma de peticiones

Al exigir que las peticiones vayan firmadas también se está ayudando a reducir la asimetría entre la cantidad de trabajo que debe llevar a cabo quien efectúa la petición y la que debe hacer quien responde a ella. El trabajo adicional que implica la verificación de la firma por parte de quien responde representa un porcentaje relativamente bajo del trabajo total de éste, mientras que el proceso de cálculo de la firma digital corresponde a una cantidad relativamente alta del trabajo total efectuado por quien emite la petición. La disminución de esta asimetría mengua el riesgo de un ataque DoS.

Ahora bien, en teoría es posible que un atacante intercepte un mensaje firmado y lo reproduzca continuamente, con lo cual logra eludir el cumplimiento de este requisito. Lo anterior puede remediarse si se exige la utilización del elemento Signature XML `<ds:SignatureProperties>` que contenga una indicación de tiempo, que a su vez puede servir para determinar si la firma es reciente. En este caso, cuanto menos tiempo haya transcurrido desde que la firma se considera válida, más seguridad existe contra ataques de denegación de servicio.

3) Restricción de acceso al URL de interacción

Al limitar a un conjunto de partes conocidas la capacidad de producir una petición a un servicio SAML a muy bajo nivel, se reduce dramáticamente el riesgo de ataques DoS. En este caso, sólo puede haber ataques que se originen dentro de dicho conjunto de partes conocidas, con lo cual disminuye bastante la exposición tanto a posibles clientes malintencionados como a ataques DoS perpetrados a través de máquinas alteradas, denominadas zombis.

Hay muchos métodos para limitar el acceso, como el que consiste en poner el respondedor SAML dentro de un intranet seguro y el que requiere el establecimiento de reglas de acceso al nivel del encaminador.

I.7 Consideraciones de seguridad relativas a las vinculaciones SAML

Las consideraciones de seguridad que se han de tener en cuenta en el diseño del protocolo de petición-respuesta SAML dependen en gran medida del tipo específico de vinculación de protocolo que se está utilizando. Se soportan las siguientes vinculaciones: la SOAP, la Reverse SOAP (PAOS), la Redirect HTTP, la Redirect/POST HTTP, la Artifact HTTP y la URI SAML.

I.7.1 Vinculación SOAP SAML

Al no tener requisitos de autenticación ni de confidencialidad o integridad de mensaje en tránsito, está expuesta a una gran variedad de ataques comunes. Se discuten aparte las consideraciones generales de las relacionadas con el caso SOAP-sobre-HTTP.

1) Escucha clandestina (eavesdropping)

Amenaza: Al no requerirse la confidencialidad en tránsito, es posible que una parte que efectúa una escucha clandestina adquiera tanto el mensaje SOAP que contiene una petición como el que contiene la respuesta correspondiente. Siendo así, se pone en riesgo la naturaleza de la petición y los detalles de la respuesta, que tal vez incluyan una o varias aserciones.

La exposición de los detalles de la petición debilitará en algunos casos la seguridad de la parte que la produce, al revelar detalles sobre los tipos de aserciones que ésta requiere o de quién los requiere. Por ejemplo, cuando alguien se entera, gracias a una escucha clandestina, de que el sitio *X* solicita a menudo al sitio *Y* aserciones de autenticación con determinado método de confirmación, puede utilizar esta información para amenazar el sitio *X*.

Del mismo modo, la escucha clandestina de una serie de peticiones de autenticación puede servir para crear un "mapa" de los recursos controlados por cierta autoridad de autorización.

Además, en ciertos casos la publicación de la petición podría constituir una violación de la privacidad. Por ejemplo, la escucha clandestina de una petición y de su respuesta puede revelar que determinado usuario está activo en el sitio donde se origina la petición, algo que no tiene por qué ser público, como en el caso de los sitios de información médica, los sitios políticos, etc. Asimismo, los detalles de las

aserciones transportadas en la respuesta pueden ser confidenciales. Lo anterior es especialmente cierto en el caso de respuestas que contengan aserciones de atributos: si éstos contienen información que no debería revelarse a terceras partes que no participan en la transacción (índices de solvencia, atributos médicos, etc.), la escucha clandestina es muy peligrosa.

Medidas preventivas: Cuando exista cualquiera de estos riesgos, la escucha clandestina se puede prevenir utilizando algún tipo de confidencialidad de mensaje en tránsito. Si se trata de mensajes SOAP, esta confidencialidad se puede implementar en el nivel SOAP o en el de transporte SOAP (o cualquier otro nivel inferior a éste).

Al decir que se añade confidencialidad en tránsito en el nivel SOAP, lo que se quiere indicar es que se construyen los mensajes SOAP de tal manera que, sin importar cuál sea el transporte SOAP, sólo la parte a la que está destinado un mensaje podrá acceder a él. En general, este problema se resuelve utilizando criptación XML. En la presente Recomendación se permite la criptación del mensaje SOAP propiamente dicho, con lo cual se elimina el riesgo de escucha clandestina, salvo si la clave empleada en la criptación ha podido ser descubierta por terceros. También es posible que quienes desarrollan aplicaciones se apoyen en la capa de transporte SOAP, o en una inferior, para proporcionar la confidencialidad en tránsito.

Los detalles acerca de cómo obtener esta confidencialidad dependen del tipo específico de transporte SOAP que se haya seleccionado. Un método consiste en utilizar HTTP sobre TLS/SSL, mientras que otros transportes requieren otras técnicas de confidencialidad en tránsito, por ejemplo un transporte SMTP debe emplear S/MIME.

En algunos casos, es posible que un nivel por debajo del de transporte SOAP proporcione la confidencialidad en tránsito requerida. Por ejemplo, si se transporta la interacción petición-respuesta a través de un túnel IPsec, tal vez el túnel garantice dicha confidencialidad.

2) Reproducción

Amenaza: Hay poca vulnerabilidad a este tipo de ataques en el nivel de la vinculación SOAP. Los ataques por reproducción son algo que concierne más a los distintos perfiles. Básicamente, lo que ha de tenerse en cuenta al respecto de la reproducción al nivel de la vinculación SOAP es que se puede utilizar como método de ataque por denegación de servicio.

Medidas preventivas: En general, la mejor manera de prevenir los ataques por reproducción consiste en evitar por encima de todo la captura de los mensajes, algo que es posible en algunos de los esquemas del nivel de transporte que sirven para proporcionar la confidencialidad en tránsito. Por ejemplo, si la conversación petición-respuesta SAML se basa en el SOAP en HTTP/TLS, no es posible que terceras partes intercepten los mensajes.

Puesto que no es necesario que un atacante por reproducción entienda el mensaje para poder reproducirlo, los esquemas como la criptación XML no protegen contra dichos ataques. Si un atacante intercepta una petición SAML que ha sido firmada por quien la produjo y criptada para ser enviada al respondedor, puede reproducirla en cualquier momento sin necesidad de descriptarla. Una petición SAML contiene información acerca del momento de expedición, con lo cual es posible establecer si se está presentando una reproducción. De otra parte, se puede utilizar la clave única de la petición (su `ID`) para determinar si la petición en cuestión es una reproducción o no.

Otras amenazas provenientes de este tipo de ataques incluyen los casos en los que existe un modelo de "cobro por petición". La reproducción podría servir para efectuar cobros importantes en una determinada cuenta.

Del mismo modo, en los modelos en los que se atribuye al cliente (o éste compra) una cantidad fija de interacciones con un sistema, un ataque por reproducción puede agotar dicho crédito, a menos que quien emite la petición ponga especial cuidado en rastrear la clave única de cada petición.

3) Inserción de mensaje

Amenaza: Se introduce en el tren de mensajes una petición o una respuesta fabricadas. Una respuesta falsa, como un "sí" fraudulento o el retorno de falsa información de atributo en respuesta a una petición de autorización, puede inducir una acción inapropiada del receptor.

Medidas preventivas: La capacidad de insertar una petición no es amenaza en el nivel de vinculación SOAP. La amenaza de inserción de una respuesta falsa puede ser un ataque del tipo denegación de servicio, por ejemplo al devolver Fallos SOAP como respuestas, pero este tipo de ataque se descubre rápidamente. El ataque más sutil consistente en la devolución de respuestas fabricadas se tiene en cuenta en el protocolo SAML, que es el adecuado puesto que, conforme a la definición de vinculación SOAP, cada respuesta SOAP ha de contener una sola respuesta de protocolo SAML, salvo si contiene un fallo. El protocolo SAML lo hace mediante dos mecanismos, a saber la correlación entre las respuestas y las peticiones gracias al atributo requerido `InResponseTo`, que hace más difícil efectuar un ataque pues hay

que interceptar las peticiones para poder generar respuestas, y a través del soporte de la autenticación de origen, bien sea basándose en respuestas SAML firmadas o bien mediante una conexión de transporte asegurada, como la SSL/TLS.

4) Supresión de mensaje

Amenaza: Este ataque impediría la llegada de una petición al respondedor o de la respuesta al solicitante.

Medidas preventivas: En ambos casos, la vinculación SOAP no se ocupa de esta amenaza. En general, la correlación entre los mensajes de petición y los de respuesta, por ejemplo con el atributo `InResponseTo` en el **StatusResponseType**, bastaría para desalentar un ataque de supresión de mensaje.

5) Modificación de mensaje

Amenaza: La modificación de mensajes es una amenaza para la vinculación SOAP en ambos sentidos.

Si se modifica la petición alterando sus detalles, puede ocurrir que se obtengan resultados muy diferentes, los cuales a su vez pueden ser empleados por un atacante inteligente para poner en peligro sistemas que dependan de la aserción devuelta. Por ejemplo, la alteración de la lista de atributos requeridos en los elementos `<Attribute>` podría amenazar la petición o provocar que el respondedor la rechace.

La modificación de la petición con el fin de cambiar el emisor aparente de la petición podría implicar una denegación de servicio o un encaminamiento incorrecto de la respuesta. Esta modificación tendría que hacerse en un nivel por debajo del SAML, por lo que está fuera del alcance de esta Recomendación.

La modificación de la respuesta, con el fin de alterar los detalles de las aserciones subsecuentes, podría provocar toda una gama de amenazas. Un ejemplo simple de ello, la alteración de los detalles de una autenticación o de una decisión de autorización, puede generar brechas importantes en la seguridad.

Medidas preventivas: Para hacer frente a estas amenazas potenciales, se ha de utilizar un sistema que garantice la integridad de los mensajes en tránsito. Aunque el protocolo SAML y la vinculación SOAP no requieren ni prohíben la utilización de sistemas que garanticen la integridad de los mensajes en tránsito, debido a la intensidad de la amenaza se recomienda enfáticamente emplear uno de dichos sistemas. Esto se puede lograr, al nivel de la vinculación SOAP, a través de la firma digital de las peticiones y de las respuestas con un sistema como Signature XML.

Si los mensajes van firmados digitalmente, se garantiza al recipiente que el mensaje no ha sido alterado en tránsito, salvo si se ha violado la clave utilizada.

También es posible preservar la integridad del mensaje en tránsito, en un nivel inferior, utilizando un transporte SOAP que proporciona la propiedad de integridad garantizada, o se basa en un protocolo que suministre dicha propiedad. Un transporte que garantiza lo anterior es el SOAP sobre HTTP con TLS/SSL.

No basta con la criptación para contar con esta protección, pues aunque no se logre alterar el mensaje interceptado *per se*, es posible reemplazarlo por uno nuevo fabricado a este fin.

6) Intromisión

Amenaza: La vinculación SOAP puede sufrir ataques por intromisión (MITM, *man-in-the-middle*). Para evitar que entidades malintencionadas se entrometan (con todos los riesgos que ya se discutieron en las secciones relativas a la escucha clandestina y la modificación de mensajes), se requiere algún tipo de autenticación bilateral.

Medidas preventivas: Un sistema de autenticación bilateral permitiría a ambas partes asegurarse de que lo que están recibiendo en una conversación proviene realmente de la otra.

En el nivel de vinculación SOAP también se podría lograr este objetivo firmando digitalmente peticiones y respuestas. Si bien de esta manera no se evitaría que un atacante que utilice escuchas clandestinas se siente en el medio y reenvíe en ambos sentidos, sí se lograría que no altere la conversación sin ser detectado.

Puesto que en muchas aplicaciones del SOAP no se utilizan sesiones, tal vez sea necesario combinar este tipo de autenticación de autor (a diferencia de la autenticación de remitente) con información de la capa de transporte, para confirmar que el remitente y el autor son la misma entidad, con lo cual se evita una cierta forma débil de escucha clandestina por parte de un "MITM entrometido".

Otra implementación podría depender del transporte SOAP que proporciona autenticación bilateral, o está configurado en una capa inferior que la suministra. Como ejemplo, considérese de nuevo el SOAP sobre HTTP con TLS/SSL, en el que se requieren certificados en el lado del cliente y en el del servidor.

Además, el intervalo de validez de las aserciones devueltas sirve para ajustar el grado de riesgo ante ataques MITM. Cuanto más corto sea dicho intervalo, menos daño se puede generar en una intromisión.

7) Utilización del SOAP con el HTTP

La vinculación SOAP requiere que las aplicaciones conformes a ella soporten HTTP sobre TLS/SSL con varios métodos de autenticación bilateral, como el Básico sobre el SSL en el lado servidor y la autenticación basada en certificado sobre el SSL en el lado servidor, por lo que siempre se dispone de estos métodos para reducir las amenazas en casos en que no haya otros sistemas de nivel inferior y las amenazas enumeradas anteriormente se consideren significativas.

Lo anterior no implica que sea obligatorio emplear el HTTP con alguna forma de autenticación bilateral. Si se obtiene, por otros medios (por ejemplo, un túnel IPsec), un nivel aceptable de protección contra los diversos riesgos, no es necesario tener un TLS completo con certificados. No obstante, en la mayoría de los casos en que se cuenta con el SOAP sobre HTTP, lo más conveniente será utilizar HTTP sobre TLS con autenticación bilateral.

En RFC que aborda el tema de la autenticación HTTP (RFC 2617 del IETF) se describen posibles ataques en el entorno HTTP cuando se trata de esquemas de autenticación básicos o de resumen de mensaje.

Ahora bien, la utilización de seguridad en el nivel de transporte (por ejemplo los protocolos SSL o TLS con el HTTP) sólo proporciona confidencialidad y/o integridad y/o autenticación para "un salto". En los modelos en los que pueda haber intermediarios o las aserciones del caso tengan que sobrevivir durante más de un salto, el HTTP con TLS/SSL no garantiza la seguridad adecuada.

I.7.2 Perfiles de inscripción única (SSO, *single sign-on*) a navegador Web

La autenticación de usuario en la fuente está explícitamente fuera del alcance, tal como ocurre con los aspectos relativos a esta autenticación de sitio fuente. Lo fundamental radica en que la entidad de sistema fuente ha de poder afirmar que el sistema de cliente autenticado con el que interactúa es el mismo en la próxima etapa de interacción. Una forma de lograrlo consiste en efectuar las primeras etapas utilizando TLS como capa de sesión por debajo del protocolo que se esté empleando para esta interacción inicial (probablemente el HTTP).

I.7.2.1 Perfil SSO

1) Escucha clandestina

Amenaza: La posibilidad de escucha clandestina existe en todos los casos basados en navegador web.

Medidas preventivas: Cuando se requiera confidencialidad (teniendo en mente que toda aserción que no se envíe con seguridad, junto con la petición correspondiente, está a merced de un oyente clandestino malintencionado), el tráfico HTTP debe cursarse a través de un transporte que garantice la confidencialidad. El HTTP sobre TLS/SSL y el Protocolo de seguridad IP satisfacen este requisito.

2) Robo de información de autenticación de usuario

Amenaza: Cuando el sujeto se autentique ante el sitio fuente a través de información de autenticación reutilizable, por ejemplo una contraseña, el robo de dicha información permitirá a quien la obtenga suplantarlo.

Medidas preventivas: Para evitarlo, debe haber una salvaguardia de confidencialidad en la conexión entre el navegador del sujeto y el sitio fuente. Así mismo, bien sea el sujeto o el sitio de destino han de tomar medidas para garantizar que el sitio origen es realmente el genuino y fiable, antes de revelar la información de autenticación. La utilización del HTTP sobre TLS permite evitar este problema.

3) Robo de testigo de portador

Amenaza: Cuando la aserción de autenticación contenga el identificador de protocolo de autenticación del portador de la aserción, el robo del artefacto permitirá a un adversario suplantar al sujeto.

Medidas preventivas: Cada uno de los siguientes métodos reduce la probabilidad de que esto ocurra:

El sitio de destino implementa una salvaguardia de confidencialidad en su conexión con el navegador del sujeto.

El sujeto o el sitio de destino se aseguran (fuera de banda) de que el sitio fuente implemente una salvaguardia de confidencialidad en su conexión con el navegador del sujeto.

El sitio de destino verifica que se haya llegado al navegador del sujeto directamente desde un sitio fuente que autenticó al sujeto él mismo.

El sitio fuente se niega a responder a más de una petición por cada aserción que corresponda al mismo ID de aserción.

Si la aserción contiene un elemento condición del tipo **AudienceRestrictionType** que identifica un dominio específico, el sitio de destino verifica que éste sea miembro de dicho dominio.

La conexión entre el sitio de destino y el sitio fuente, a través de la cual se hace pasar el ID de aserción, se implementa con una salvaguarda de confidencialidad.

El sitio de destino, en su comunicación con el sitio fuente, a través de la cual se hace pasar el ID de aserción, debe verificar que el sitio fuente es genuinamente el sitio fuente esperado y fiable.

4) Reproducción

Para este tipo de perfiles, existe la posibilidad de un ataque por reproducción de mensajes. Un ataque por reproducción puede ser utilizado para tratar de denegar el servicio o para obtener información fraudulentamente. Las medidas preventivas que se han de tomar dependen del tipo específico de vinculación y han sido descritas en las subcláusulas anteriores.

5) Inserción de mensajes

Este tipo de ataques se trata en I.7.1.

6) Supresión de mensajes

Amenaza: La supresión de un mensaje durante cualquier etapa de las comunicación entre el navegador, el que produce aserciones SAML y quien las utiliza, provocará el fallo de la interacción. Si bien el resultado será una denegación de algún servicio, no se incrementa el peligro de revelar alguna información.

Medidas preventivas: Basta con emplear un canal de transporte cuya integridad esté protegida para evitar la amenaza de supresión de mensajes, cuando no hay intermediarios presentes.

7) Modificación de mensajes

Amenaza: Para este tipo de perfiles, existe la posibilidad de alteración de los mensajes en el tren. Algunos posibles resultados indeseables son:

La alteración de la petición inicial puede provocar el rechazo del emisor SAML o causar la creación de un artefacto destinado a un recurso diferente que el solicitado.

La alteración del artefacto puede provocar la denegación de servicio en el cliente SAML.

La alteración de las aserciones propiamente dichas, mientras están transitando, podría ocasionar todo tipo de resultados indeseables (si no están firmadas) o la denegación de servicio (si están firmadas y el cliente las rechaza).

Medidas preventivas: Con el fin de evitar la modificación de mensajes, se debe transportar el tráfico mediante un sistema que garantice su integridad de extremo a extremo.

En el caso de perfiles basados en el navegador, para proporcionar integridad de mensajes en tránsito se recomienda utilizar HTTP sobre TLS/SSL con un código de cifrado que permita verificar la integridad de la información.

8) Intromisión

Amenaza: Los ataques por intromisión son especialmente perniciosos para este conjunto de perfiles. El MITM puede retransmitir peticiones, interceptar la aserción (o el artefacto) devuelta, y retransmitir hacia el origen una falsa. En tal caso, aunque el usuario original no puede acceder al recurso en cuestión, el MITM sí puede hacerlo sirviéndose del recurso capturado.

Medidas preventivas: Se necesitan varias medidas. En primer lugar, la utilización de un sistema que suministre una autenticación bilateral fuerte hará mucho más difícil que un MITM se introduzca en la conversación.

No obstante, sigue existiendo la posibilidad de que haya un MITM que actúe simplemente como puerto bidireccional de reenvío y que esté escuchando clandestinamente la información con la intención de interceptar la aserción o el manejador devueltos (e incluso tal vez alterar lo que finalmente se devuelve a quien produjo la petición). La puesta en marcha de un sistema de confidencialidad evitará las escuchas clandestinas, mientras que la de un sistema de integridad de datos lo hará con la alteración del mensaje durante el reenvió en un puerto.

Para este conjunto de perfiles, se pueden satisfacer todos los requisitos de autenticación fuerte de sesión bilateral, confidencialidad e integridad de datos, mediante la utilización del HTTP sobre TLS/SSL, si la capa TLS/SSL emplea un código de cifrado adecuado (una encriptación lo suficientemente fuerte como para proporcionar la confidencialidad y soportar la integridad de datos) y requiere certificados X.509 v3 para la autenticación.

9) Suplantación de identidad sin reautenticación

Amenaza: Intentos innumerables de suplantar la identidad de un principal que está inscrito legítimamente y, por tanto, obtener acceso a recursos protegidos.

Cuando un principal se ha conectado con (*logged into*) un proveedor de identidad, no siempre se reautenticará dicho principal al recibirse después mensajes `<AuthnRequest>` de otros proveedores de servicios. Ahora bien, los Principales han de autenticarse, salvo si el proveedor de identidad logra establecer determine que un `<AuthnRequest>` está asociado no sólo a la identidad del Principal, sino también con una sesión de proveedor de identidad correctamente autenticada para dicho Principal.

Medidas preventivas: En las implementaciones en las que se deba tener en cuenta esta amenaza, los proveedores de identidad deben mantener información de estado relativa a las sesiones activas, y deben validar la correspondencia entre un `<AuthnRequest>` y una sesión activa antes de producir una `<Response>` sin autenticar antes al Principal. Se pueden emplear las cookies enviadas por los proveedores de identidad para soportar este proceso de validación, aunque Liberty no impone que se haga de esta manera.

I.7.2.2 Perfil de cliente mejorado y apoderado

1) Intromisión

Amenaza: Intercepción de mensajes SOAP `AuthnRequest` y `Response`, con lo cual es posible la suplantación de identidad del Principal.

Una entidad de sistema espuria puede interponerse ella misma, como intrusa (MITM), entre el cliente mejorado y un proveedor legítimo de servicio, y fungir como proveedor de servicio en interacciones con el cliente mejorado y como cliente mejorado en interacciones con el proveedor legitimo de servicio. De esta manera, en primera instancia, el MITM puede interceptar el `AuthnRequest` del proveedor de servicio y sustituir cualquier URL de su elección por el valor `responseConsumerServiceURL` en el bloque de encabezamiento PAOS, antes de reenviar el `AuthnRequest` al cliente mejorado. A menudo, el MITM inserta un valor de URL que apunta a él mismo, tras lo cual si el cliente mejorado recibe luego una Response del proveedor de identidad y envía entonces la Response contenida al `responseConsumerServiceURL` recibido del MITM, el intruso podrá suplantar al Principal ante el proveedor legitimo de servicio.

Medidas preventivas: El proveedor de identidad especifica al cliente mejorado la dirección a la cual éste ha de enviar la Response. El `responseConsumerServiceURL` en el encabezamiento PAOS se emplea solamente para las respuestas de error provenientes del cliente mejorado – conforme a lo que se especifica en el perfil.

2) Denegación de servicio

Amenaza: Modificación de la petición SOAP `AuthnRequest` de tal manera que no pueda ser procesada, cambiando por ejemplo el valor de atributo de servicio de bloque de encabezamiento PAOS por un valor desconocido, o cambiando el `ProviderID` o la `IDPList` del bloque de encabezamiento ECP, con el fin de hacer fracasar la petición.

Medidas preventivas: Proporcionar protección de integridad al mensaje SOAP, gracias a la Seguridad de mensaje SOAP o al SSL/TLS.

I.7.2.3 Perfil de descubrimiento de proveedor de identidad

Amenaza: Ataque de "envenenamiento" de cookie, en el que se modifican los parámetros al interior de esta última, con lo cual se descubre, por ejemplo, un proveedor de identidad fraudulento.

Medidas preventivas: El mecanismo específico que consiste en utilizar un dominio común limita la viabilidad de esta amenaza.

I.7.2.4 Perfil de inscripción única

Amenaza: Un atacante pasivo puede colectar un identificador de nombre del Principal

Al empezar, un atacante pasivo puede colectar la información `<LogoutRequest>` cuando ésta se produce en la redirección. La exposición de dicha información constituye una amenaza a la privacidad.

Medidas preventivas: Se deberían efectuar todos los intercambios a través de un transporte seguro, como SSL o TLS.

Amenaza: Mensaje `<LogoutRequest>` sin firmar.

Una entidad de sistema espuria puede insertar un `<LogoutRequest>` sin firma, provocando la denegación de servicio al Principal. Si se supone que es posible deducir o calcular el `NameID`, cabe entonces concebir que el agente de usuario pueda ser forzado a emitir un mensaje `<LogoutRequest>` fabricado.

Medidas preventivas: Firmar el mensaje `<LogoutRequest>`. El proveedor de identidad también puede verificar la identidad de un Principal de no haber una petición firmada.

I.7.2.5 Perfiles de gestión de identificador de nombre

Amenaza: Cuando entidades de sistema pueden correlacionar información o, de lo contrario, exponer inadecuadamente información de identidad, afectando la privacidad.

Medidas preventivas: El IDP ha de tener cuidado en utilizar varios identificadores de nombre con diferentes proveedores de servicios, para el mismo principal. El IDP debería criptar el identificador de nombre que devuelve al proveedor de servicio, con lo cual en las interacciones subsecuentes se empela un identificador opaco.

I.7.2.6 Perfiles de atributo

Las amenazas relativas a las vinculaciones asociadas con perfiles de atributo se examinaron antes. No se conocen otras amenazas específicas de perfil.

Apéndice II

Registro de aplicación del tipo de medios MIME application/samlassertion+xml

En este apéndice se presenta el registro del tipo de medios MIME Application/Assertion SAML.

Nombre de tipo de medios MIME

– `application`

Nombre de subtipo MIME

– `samlassertion+xml`

Parámetros requeridos

– Ninguno.

Parámetros facultativos

– `charset`

– Igual al parámetro `charset` de `application/xml` de RFC 3923 del IETF.

Consideraciones relativas a la codificación

– Igual a la de application/xml de RFC 3923 del IETF.

Consideraciones de seguridad

Los objetos de tipo `samlassertion+xml` no incluyen contenido ejecutable. Sin embargo, las aserciones SAML son objetos basados en XML. Siendo así, disponen de todas las consideraciones generales de seguridad indicadas en la cláusula 10 de RFC 3923 del IETF, así como otras, pues se trata de objetos de seguridad explícitos. Por ejemplo, con frecuencia los objetos de tipo `samlassertion+xml` contendrán información que pueda identificar a una persona natural o se relacione con ella, y que puede ser utilizada como base para decisiones relativas a sesiones y al control de acceso.

A fin de prevenir posibles problemas, los objetos de tipo `samlassertion+xml` contienen información que el remitente debería firmar adecuadamente. Cualquiera de dichas firmas debe ser verificada por el recibiente de los datos – como firma válida y como firma del remitente. Quienes producen objetos del tipo `samlassertion+xml` que contengan aserciones SAML puedan encriptar todas las aserciones, o parte de ellas.

Además, los perfiles y las vinculaciones de protocolo SAML especifican el empleo de canales seguros, cuando corresponda.

La versión 2 del SAML (esta Recomendación) incorpora diversas técnicas de protección de privacidad en su diseño. Por ejemplo: se pueden atribuir a sujetos asideros (*handles)* opacos, propios de interacciones entre entidades específicas de sistema. Solamente las partes específicas pueden hacer corresponder dichos *handles* con identificadores de un contexto más amplio (por ejemplo direcciones de correo electrónico, identificadores de cuenta, etc.).

Consideraciones de interoperabilidad

Las aserciones SAML tienen explícitamente la versión correspondiente. Las partes que retransmiten deberían garantizar que observan la información de aserción y se comportan conforme a ella.

Especificación publicada

La versión 2 del SAML (esta Recomendación) especifica explícitamente la utilización de tipos de medios MIME `application/samlassertion+xml`. No obstante, se puede concebir que en la práctica se transporten aserciones que no son SAML (es decir, SAMLv1 y/o SAMLv1.1) mediante vinculaciones SAML.

Aplicaciones que utilizan este tipo de medios

En principio cualquier aplicación que implemente SAML, así como las aplicaciones que implementen especificaciones basadas en el SAML.

Información adicional

Número(s) mágico(s)

En general, el mismo que para la application/xml. En particular, el elemento raíz XML del objeto devuelto tendrá un espacio de nombres calificado con:

- un nombre local de: `Assertion`
- un espacio de nombres URI de: uno de los URI del espacio de nombres XML SAML específica de la versión de aserción, con arreglo a la definición dada por la Recomendación "básica" sobre el SAML específico de la versión

En particular con el SAML, los elementos raíz del objeto devuelto pueden ser `<saml:Assertion>` o `<saml:EncryptedAssertion>`, donde "saml" representa cualquier prefijo de espacio de nombres XML que hace corresponder con el URI del espacio de nombres de aserción SAML:

`urn:oasis:names:tc:SAML:2.0:assertion`

Extensión o extensiones de fichero

Ninguna.

Código(s) de tipo de fichero Macintosh

Ninguno.

Personas & direcciones de correo electrónico a quienes dirigirse para obtener más información

Este registro se hizo en nombre del Comité técnico de seguridad de servicios OASIS (SSTC, *Security Services Technical Committee*).

Destino

Común.

Apéndice III

Registro de tipos de medios MIME application/samlmetadata+xml

En este apéndice se define un tipo de medios MIME – application/samlmetadata+xml – que puede emplearse con la serialización XML de metadatos del lenguaje de marcaje de aserción de seguridad (SAML, *security assertion markup language*).

1) Nombre de tipo de medios MIME

– `application`

2) Nombre de subtipo MIME

– `samlmetadata+xml`

3) Parámetros requeridos

– Ninguno.

4) Parámetros facultativos

– `charset`

– Igual al parámetro `charset` de application/xml (véase RFC 3023 de IETF).

5) Consideraciones relativas a la codificación

– Las mismas de `application/xml` en RFC 3023 de IETF.

6) Consideraciones de seguridad

Los objetos de tipo `samlmetadata+xml` no incluyen contenido ejecutable. Sin embargo, estos objetos se basan en XML, y por ende disponen de todas las consideraciones generales de seguridad indicadas en la cláusula 10 de RFC 3023 del IETF.

A fin de resolver probables problemas que puedan presentarse, quien publica puede firmar objetos del tipo `samlmetadata+xml`. Cualquiera de estas firmas debería ser verificada por quien recibe los datos – certificando que es una firma válida y que es la de quien publica.

7) Consideraciones de interoperabilidad

Los metadatos SAML aceptan explícitamente la identificación de protocolos y versiones soportados por las entidades identificadas. Por ejemplo, una entidad proveedora de identidad puede señalarse como conforme a la versión v2.0 del SAML y otros protocolos si éstos son identificables sin ambigüedad vía el URI. Esta información de soporte de protocolo se transporta a través del atributo `protocolSupportEnumeration` de objetos de metadatos del **RoleDescriptorType**.

8) Recomendación publicada

Los metadatos SAML especifican explícitamente la utilización del tipo de medios MIME `application/samlmetadata+xml`.

Aplicaciones que utilizan este tipo de medios.

En principio cualquier aplicación que acepte la v2.0 del SAML, así como las que implementan especificaciones basadas en SAML.

9) Información adicional

1) Número(s) mágico(s)

En general, el mismo de la `application/xml` de RFC 3023 del IETF. En particular, el elemento raíz XML del objeto devuelto tendrá un espacio de nombres calificado con:

– un nombre local de: `EntityDescriptor` o `AffiliationDescriptor` o `EntitiesDescriptor`;

– un URI de espacio de nombres de: `urn:oasis:names:tc:SAML:2.0:metadata` (el mismo espacio de nombres de metadatos SAMLv2.0).

10) Extensión o extensiones de fichero

Ninguna.

11) Código(s) de tipo de fichero Macintosh

Ninguno.

12) Personas & direcciones de correo electrónico a quienes dirigirse para obtener más información

Este registro se hizo en nombre del Comité Técnico de Seguridad de Servicios OASIS (SSTC, *Security Services Technical Committee*).

13) Destino

Común.

Apéndice IV

Utilización de SSL

Es posible que alguna implementación del SAML soporte el SSL 3.0 junto con, o en lugar del TLS 1.0. Las implementaciones que utilicen el SSL 3.0 deberían garantizar que su seguridad general sea coherente con las restricciones que existen para los códigos de cifrado en el TLS. Por ejemplo, los requisitos que se imponen a la utilización de los códigos de cifrado TLS_RSA_WITH_3DES_EDE_CBC_SHA se traducen en el empleo del código de cifrado SSL_RSA_WITH_3DES_EDE_CBC_SHA. Las implementaciones compatibles con el SSL FIPS se sirven del código de cifrado FIPS que corresponde al código de cifrado SSL SSL_RSA_WITH_3DES_EDE_CBC_SHA.

Las implementaciones TLS del perfil SSO Web de SAML que soporta el código de cifrado TLS_RSA_WITH_3DES_EDE_CBC_SHA utilizarán el código de cifrado SSL_RSA_WITH_3DES_EDE_CBC_SHA.

Apéndice V

Contexto de autenticación de esquema SAML

Este apéndice contiene un esquema de contexto de aplicación SAML para certificado SSL (sslcert).

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
```

```
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Apéndice VI

Esquema XML de tipos de contexto de autenticación

En este apéndice se enumeran todos los esquemas XML de tipos de contexto de autenticación y el esquema XML de contexto de autenticación propiamente dicho, utilizados para la validación de declaraciones generalizadas individuales. El esquema de tipos no tiene un espacio de nombres objetivo en sí mismo, por lo que se incluye en el apéndice V.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between a Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
      <xs:documentation>
        This element indicates that identification has been
        performed in a physical
        face-to-face meeting with the principal and not in an
        online manner.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="credentialLevel">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="primary"/>
            <xs:enumeration value="secondary"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
```

```
<xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

<xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characterstics that describe how the
      'secret' (the knowledge or possession
      of which allows the Principal to authenticate to the
      Authentication Authority) is kept secure
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a shared secret key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a private key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>The actions that must be performed
      before the private key can be used. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeySharing" type="KeySharingType">
  <xs:annotation>
    <xs:documentation>Whether or not the private key is shared
      with the certificate authority.</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyStorage" type="KeyStorageType">
  <xs:annotation>
    <xs:documentation>
      In which medium is the key stored.
      memory - the key is stored in memory.
      smartcard - the key is stored in a smartcard.
      token - the key is stored in a hardware token.
      MobileDevice - the key is stored in a mobile device.
      MobileAuthCard - the key is stored in a mobile
      authentication card.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
<xs:element name="UserSuffix" type="ExtensionOnlyType"/>

<xs:element name="Password" type="PasswordType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a password (or passphrase)
```

```
      has been used to
      authenticate the Principal to a remote system.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationPin" type="ActivationPinType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a Pin (Personal
      Identification Number) has been used to authenticate the Principal
      to some local system in order to activate a key.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Token" type="TokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a hardware or software
      token is used
      as a method of identifying the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TimeSyncToken" type="TimeSyncTokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a time synchronization
      token is used to identify the Principal. hardware -
      the time synchronization
      token has been implemented in hardware. software - the
      time synchronization
      token has been implemented in software. SeedLength -
      the length, in bits, of the
      random seed used in the time synchronization token.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Smartcard" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a smartcard is used to
      identify the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Length" type="LengthType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the minimum and/or maximum
      ASCII length of the password which is enforced (by the UA or the
      IdP). In other words, this is the minimum and/or maximum number of
      ASCII characters required to represent a valid password.
      min - the minimum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
      max - the maximum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimit" type="ActivationLimitType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the length of time for which an
      PIN-based authentication is valid.
    </xs:documentation>
```

```
    </xs:annotation>
  </xs:element>

  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the
        Principal or auto-supplied by the Authentication Authority.
        principalchosen - the Principal is allowed to choose
        the value of the password. This is true even if
        the initial password is chosen at random by the UA or
        the IdP and the Principal is then free to change
        the password.
        automatic - the password is chosen by the UA or the
        IdP to be cryptographically strong in some sense,
        or to satisfy certain password rules, and that the
        Principal is not free to change it or to choose a new password.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the
        Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Authenticator" type="AuthenticatorBaseType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
```

```
  <xs:element name="PreviousSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system
        and is now re-used (e.g. a Master Secret is used to derive new
        session keys in TLS, WTLS).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a challenge-response protocol utilizing shared
        secret keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a mechanism which involves the Principal computing
        a digital signature over at least challenge data provided
        by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using
        the local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
```

```
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key and uses it for
        shared secret key agreement with the Authentication Authority
        (e.g., via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using bare HTTP utilizing no additional security
        protocols.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="IPSec" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using a transport mechanism protected by an IPSEC session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="WTLS" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
```

```
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by a WTLS session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted solely across a mobile network using no additional
      security mechanism.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
<xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

<xs:element name="SSL" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (e.g., liability constraints, obligations, etc.) can be placed.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

<xs:simpleType name="nymType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AuthnContextDeclarationBaseType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0"/>
```

```
    <xs:element ref="AuthnMethod" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:sequence>
    <xs:element ref="PhysicalVerification" minOccurs="0"/>
    <xs:element ref="WrittenConsent" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="nym" type="nymType">
    <xs:annotation>
      <xs:documentation>
        This attribute indicates whether or not the
        Identification mechanisms allow the actions of the Principal to
        be linked to an actual end user.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="PrivateKeyProtection"/>
      <xs:element ref="SecretKeyProtection"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit" minOccurs="0"/>
    <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
    <xs:element ref="Authenticator" minOccurs="0"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
```

```
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
  <xs:sequence>
    <xs:element ref="PreviousSession" minOccurs="0"/>
    <xs:element ref="ResumeSession" minOccurs="0"/>
    <xs:element ref="DigSig" minOccurs="0"/>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="ZeroKnowledge" minOccurs="0"/>
    <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
    <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
    <xs:element ref="IPAddress" minOccurs="0"/>
    <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
    <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
    <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
    <xs:element ref="UserSuffix" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="AuthenticatorBaseType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="HTTP"/>
      <xs:element ref="SSL"/>
      <xs:element ref="MobileNetworkNoEncryption"/>
      <xs:element ref="MobileNetworkRadioEncryption"/>
      <xs:element ref="MobileNetworkEndToEndEncryption"/>
      <xs:element ref="WTLS"/>
      <xs:element ref="IPSec"/>
      <xs:element ref="PSTN"/>
      <xs:element ref="ISDN"/>
      <xs:element ref="ADSL"/>
    </xs:choice>
```

```
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
        <xs:attribute name="min" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="3"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="max" type="xs:integer" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ActivationPinType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="ActivationLimit" minOccurs="0"/>
```

```
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
  <xs:attribute name="requiredChars" type="xs:string" use="required"/>
  <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
  <xs:attribute name="case" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DeviceTypeType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="hardware"/>
    <xs:enumeration value="software"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a number of usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      the session.
```

```
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="ActivationLimitDurationType">
    <xs:attribute name="duration" type="xs:duration" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitUsagesType">
    <xs:attribute name="number" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitSessionType"/>

  <xs:complexType name="LengthType">
    <xs:attribute name="min" type="xs:integer" use="required"/>
    <xs:attribute name="max" type="xs:integer" use="optional"/>
  </xs:complexType>

  <xs:simpleType name="mediumType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="memory"/>
      <xs:enumeration value="smartcard"/>
      <xs:enumeration value="token"/>
      <xs:enumeration value="MobileDevice"/>
      <xs:enumeration value="MobileAuthCard"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" type="mediumType" use="required"/>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:sequence>
      <xs:element ref="SwitchAudit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExtensionOnlyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Extension" type="ExtensionType"/>

  <xs:complexType name="ExtensionType">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>


<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0.
          This is just an include of all types from the schema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schema>
```

NOTA – En el apéndice IV se describe la utilización de SSL.

Apéndice VII

Perfil de atributo PAC DCE SAML

En este apéndice se trata el perfil de vinculación SAML en el caso de un entorno de computación distribuido (DCE, *distributed computing environment*), de certificados de atributo de privilegio (PAC, *privilege attribute certificates*) (véase el DCE *opensource*).

VII.1 Perfil de atributo PAC DCE

El perfil de atributo PAC DCE define la expresión de información PAC DCE en la forma de nombres y valores de atributos SAML, y sirve para normalizar una correspondencia entre la información primaria que proporciona una identidad del principal DCE y un conjunto de atributos SAML. Este perfil se basa en el perfil de atributo UUID definido en 11.4.9.3.

1) Información requerida

- **Identificación**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE` (este también es el espacio de nombres objetivo atribuido en el esquema de perfil de atributo PAC DCE correspondiente del anexo A).
- **Información de contacto**: security-services-comment@lists.oasis-open.org
- **Descripción**: Véase más adelante.
- **Actualizaciones**: Ninguna.

2) Descripción PAC

Un PAC DCE es una estructura ampliable que puede transportar atributos de registro DCE arbitrarios, aunque haya un conjunto común de información básica entre los principales, que constituya el grueso de la identidad DCE:

- el "sector" o la "célula" DCE del principal;
- el identificador único del principal;
- la membresía del grupo local DCE primario del principal;
- el conjunto de las membresías del grupo local DCE del principal (varios valores);
- el conjunto de las membresías del externo DCE del principal (varios valores).

El valor o los valores primarios de cada uno de estos atributos es un UUID.

3) Denominación de atributos SAML

Este perfil define una correspondencia de información específica DCE con atributos SAML y define, por consiguiente, nombres reales de atributos específicos, en lugar de una convención de denominación.

Para todo atributo definido por este perfil, el atributo XML `NameFormat` en los elementos `<Attribute>` ha de tener el valor `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

A los efectos de legibilidad para las personas, también puede haber un requisito que consiste en que algunas aplicaciones lleven un nombre de cadena facultativo junto con el URI. El atributo facultativo XML `FriendlyName` puede servir a estos fines.

4) Comparación de nombre de atributo

Dos elementos `<Attribute>` se refieren al mismo atributo SAML si, y sólo si, sus valores de atributo XML Name son iguales conforme a la Rec. UIT-T X.667. La comparación no depende del atributo `FriendlyName`.

5) Atributos XML específicos del perfil

No se definen otros atributos XML para ser empleados con el elemento `<Attribute>`.

6) Valores de atributo SAML

El valor o los valores primarios de cada uno de los atributos definidos por este perfil es un UUID. Se utiliza la sintaxis URN descrita en 11.4.9.3 para representar dichos valores.

No obstante, este perfil acepta información adicional asociada con el valor UUID, compuesta por una cadena fácilmente lisible para las personas, y otro UUID que representa un dominio o una célula DCE. Esta información adicional se lleva en el elemento `<AttributeValue>` de los atributos XML `FriendlyName` y Realm definidos en el espacio de nombres XML

urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE. No se tata del atributo XML FriendlyName definido en la cláusula 8, aunque tiene el mismo propósito básico.

En el esquema se muestra cómo se utilizan los atributos y el tipo complejo XML específicos del perfil en un xsi:type (anexo A):

```
<schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

7) Definiciones de atributo

A continuación se enumeran los atributos SAML definidos por este perfil. En cada caso, se puede incluir un atributo XML xsi:type en el elemento <AttributeValue>, pero éste debe tener el valor **dce:DCEValueType**, donde el prefijo dce es arbitrario y debe estar ligado al espacio de nombres XML urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE.

Este tipo de utilización del xsi:type hará necesario validar consumidores de atributo, con el fin de incluir el esquema de ampliación definido por este perfil.

a) Sector

Este atributo de valor único representa el sector o la célula DCE del sujeto de aserción SAML.

Nombre: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm

El elemento único <AttributeValue> contiene un UUID en forma URN, que identifica el sector o la célula DCE del sujeto de aserción SAML, con un atributo XML FriendlyName específico del perfil facultativo que contiene el nombre de cadena del sector.

b) Principal

Este atributo de valor único representa la identidad del principal DCE del sujeto de aserción SAML.

Nombre: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal

El elemento único <AttributeValue> contiene un UUID en forma URN, que identifica la identidad del principal DCE del sujeto de aserción SAML, con un atributo XML FriendlyName específico del perfil facultativo que contiene el nombre de la cadena del principal.

Se puede incluir el atributo XML Realm específico del perfil, y éste debe contener un UUID en forma URN, que identifica el dominio o la célula DCE del sujeto de aserción SAML.

c) Grupo primario

Este atributo de valor único representa la membresía del grupo DCE primario del sujeto de aserción SAML.

Nombre: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-group`

El elemento único `<AttributeValue>` contiene un UUID en forma URN, que identifica el grupo DCE primario del sujeto de aserción SAML, con un atributo XML `FriendlyName` específico del perfil facultativo que contiene el nombre de la cadena del grupo.

Se puede incluir el atributo XML Realm específico del perfil, y éste debe contener un UUID en forma URN, que identifica el sector o la célula DCE del sujeto de aserción SAML.

d) Grupos

Este atributo, que puede tener varios valores, representa la membresía del grupo local DCE del sujeto de aserción SAML.

Nombre: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups`

Cada elemento `<AttributeValue>` contiene un UUID en forma URN que identifica la membresía del grupo DCE del sujeto de aserción SAML, con un atributo XML `FriendlyName` específico del perfil facultativo que contiene el nombre de la cadena del grupo.

Se puede incluir el atributo XML Realm específico del perfil, y éste debe contener un UUID en forma URN, que identifica el sector o la célula DCE del sujeto de aserción SAML.

e) Grupos externos

Este atributo, que puede tener varios valores, representa la membresía del grupo externo DCE del sujeto de aserción SAML.

Nombre: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-groups`

Cada elemento `<AttributeValue>` contiene un UUID en forma URN que identifica la membresía del grupo externo DCE del sujeto de aserción SAML, con un atributo XML `FriendlyName` específico del perfil facultativo que contiene el nombre de la cadena del grupo.

Se puede incluir el atributo XML Realm específico del perfil, y éste debe contener un UUID en forma URN, que identifica el sector o la célula DCE del grupo externo.

VII.2 DCE de esquema SAML

Este es el esquema de contexto de autenticación SAML para un entorno de computación distribuida (DCE).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

VII.3 Ejemplo

A continuación se presenta un ejemplo de transformación de información PAC en atributos SAML que pertenecen a un principal DCE llamado "jdoe" en el dominio "example.com", miembro de los grupos locales "cubicle-dwellers" y "underpaid" y del grupo externo "engineers".

```
<saml:Assertion
xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE" ...>
  <saml:Issuer>...</saml:Issuer>
  <saml:Subject>...</saml:Subject>
  <saml:AttributeStatement>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="example.com">
    urn:uuid:003c6cc1-9ff8-10f9-990f-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="jdoe">
    urn:uuid:00305ed1-a1bd-10f9-a2d0-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-
group">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
       dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
       dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="underpaid">
    urn:uuid:006a5a91-a2b7-10f9-824d-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-
groups">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="engineers"
       dce:Realm="urn:uuid:00583221-a35f-10f9-8b6e-004005b13a2b">
    urn:uuid:00099cf1-a355-10f9-9e95-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Apéndice VIII

Aclaraciones del OASIS relativas al SAML

En este apéndice se presentan estudios adicionales acerca del SAML v2.0 efectuados en el marco del OASIS. El Grupo sobre el SAML del OASIS decidió publicar, como documento aparte, estas aclaraciones (véase OASIS:2006 PE). Dichas aclaraciones no tienen carácter normativo y no fueron incorporadas en la versión 2.0 del SAML del OASIS. En la presente Recomendación se enumeran dichos estudios en este apéndice para garantizar que quienes implementen el SAML estén al tanto de los debates que se dieron tras la publicación de la v2.0 SAML del OASIS como norma OASIS.

VIII.1 Posible error: PE14

Descripción: Es necesario definir mejor "Allowcreate"

Aplicabilidad en la Recomendación:

Véanse las notas del caso en 8.2.4.1 y 8.2.6. Además, a continuación se suministra la aclaración al segundo párrafo de 8.2.6.3:

Cuando se incluye en la petición el elemento `<Terminate>`, quiere decir que el proveedor que solicita está indicando que (en el caso de un proveedor de servicio) no aceptará más aserciones del proveedor de identidad o (en el caso de un proveedor de identidad) no enviará más aserciones al proveedor de servicio acerca del principal.

Si el proveedor que recibe está manteniendo un estado asociado con el identificador de nombre, por ejemplo el valor del identificador propiamente dicho (en el caso de identificador basado en pares), un valor `SPProvidedID`, el consentimiento del remitente a la creación o utilización del identificador, etc., podrá entonces efectuar cualquier mantenimiento sabiendo que ha terminado la relación representada por el identificador de nombre.

Toda otra operación que efectúe después el recibiente en nombre del remitente, relacionada con el principal (por ejemplo, una `<AuthnRequest>` enviada después), debería tener en cuenta que no hay ningún estado previo.

Es probable que la terminación elimine todo comportamiento de gestión de estado activado por la utilización del atributo `AllowCreate` en el protocolo de petición de autenticación de 8.2.4. En los casos en los que no se emplee dicho atributo, se evitará muy probablemente la utilización del elemento `<Terminate>` o se lo tratará como un simple consejo.

Obsérvese que casi nunca (una excepción digna de mencionarse se presenta en las reglas que rodean el atributo `SPProvidedID`) se imponen requisitos relativos al proveedor de identidad o al proveedor de servicios, sobre la creación o utilización de un estado persistente. Por consiguiente, no se impone ningún comportamiento explícito al recibir 450 el elemento `<Terminate>`. No obstante, de haber un estado persistente relativo a la utilización de un identificador (por ejemplo si hubiera un atributo `SPProvidedID` adjunto), el elemento `<Terminate>` proporciona una indicación clara de que se debería suprimir dicho estado (o marcarlo como obsoleto de alguna manera).

VIII.2 Posible error: PE26

Descripción: Es necesario aclarar más algunos aspectos del perfil SSO

Aplicabilidad en la Recomendación: Se aclaran las siguientes subcláusulas conforme al texto suministrado:

11.4.1.4.2 Utilización de <Response>

Si el proveedor de identidad desea devolver un error, no puede incluir ninguna aserción en el mensaje `<Response>`. De lo contrario, si la petición es exitosa (o si la respuesta no está asociada con una petición), el elemento `<Response>` ha de cumplir con lo siguiente:

- Si la respuesta no está firmada, se puede prescindir del elemento `<Issuer>`, pero si lo hubiere (o si la respuesta estuviera firmada), tendría que incluir el identificador único del proveedor que emite la identidad; se puede ignorar el atributo `Format` u otorgarle un valor igual a `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- Debe incluir por lo menos una `<Assertion>`. Cada elemento `<Issuer>` de la aserción debe contener el identificador único del proveedor que emite la identidad; se debe ignorar el atributo de `Format` u otorgarle un valor de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Obsérvese que en este perfil se supone que se cuenta con un sólo proveedor de identidad que responde y que todas las aserciones presentes en una respuesta deben provenir de la misma.

- Si se incluyen varias aserciones, cada elemento <Subject> de una aserción ha de referirse al mismo principal. Se puede permitir que el contenido de dichos elementos <Subject> difiera (por ejemplo, empleando diferentes elementos <NameID> o elementos alternativos <SubjectConfirmation>).
- Toda aserción que se produzca, a los efectos de ser utilizada con este perfil, debe contener un elemento <Subject> que tenga por lo menos un elemento <SubjectConfirmation> cuyo método sea urn:oasis:names:tc:SAML:2.0:cm:bearer. Dicha aserción se denomina de portador. Es posible que las aserciones de portador contengan otros elementos <SubjectConfirmation>.
- También se pueden incluir las aserciones que no dispongan de una <SubjectConfirmation> de portador; el tratamiento de aserciones adicionales o de elementos <SubjectConfirmation> queda fuera del alcance de este perfil.
- Por lo menos un elemento <SubjectConfirmation> de portador debe incluir un elemento <SubjectConfirmationData> que, a su vez, ha de contener un atributo Recipient que incluya el URL de servicio de consumidor de aserción del proveedor de servicio y un atributo NotOnOrAfter que limite el periodo durante el cual se puede entregar la aserción. También puede tener un atributo Address que delimite la dirección de cliente desde la cual se pueden entregar aserciones. No debe contener un atributo NotBefore. Si el mensaje que lo incluye es una respuesta a una <AuthnRequest>, el atributo InResponseTo debe corresponder con el ID de la petición.
- El conjunto de una o varias aserciones de portador debe incluir por lo menos una <AuthnStatement> que refleje la autenticación del principal con el proveedor de identidad. Si bien puede haber varios elementos <AuthnStatement>, en este perfil no se define la semántica de varias declaraciones.
- Si el proveedor de identidad soporta el perfil de desinscripción única, definido en 11.4.1.4.5, toda declaración de autenticación ha de incluir un atributo SessionIndex para permitir que el proveedor de servicio haga peticiones de desinscripción sesión por sesión.
- Es potestad del proveedor de identidad incluir otras declaraciones en la aserción o aserciones de portador. En particular, se pueden incluir elementos <AttributeStatement>. La <AuthnRequest> puede contener un atributo XML AttributeConsumingServiceIndex, que hace referencia en formación acerca de los atributos requeridos o deseados en la cláusula 9. El proveedor de identidad puede hacer caso omiso de ello, o enviar otros atributos.
- Cada aserción de portador debe contener una <AudienceRestriction> que incluya el identificador único del proveedor de servicio como una <Audience>.
- Se pueden tener en cuenta otras condiciones (y otros elementos <Audience>) conforme lo solicite el proveedor de servicio, o lo decida el proveedor de identidad. (Todas estas condiciones deben, desde luego, ser comprendidas y aceptadas por el proveedor de servicio, para que se considere válida la aserción).
- El proveedor de identidad no está obligado a satisfacer el conjunto de <Conditions> en el <AuthnRequest>, si lo hubiere.

11.4.1.4.3 Reglas de procesamiento de mensaje <Response>

Sin importar el tipo de vinculación SAML empleado, el proveedor de servicio debe:

- Verificar todas las firmas presentes en la aserción y aserciones o en la respuesta.
- Verificar que el atributo Recipient en cualquier <SubjectConfirmationData> portador corresponda con el URL del servicio de consumidor de aserción al cual se entregó la <Response> o el artefacto.
- Verificar que no ha pasado el atributo NotOnOrAfter en ningún <SubjectConfirmationData> portador, sujeto a que se permita un sesgo de temporización entre los proveedores.
- Verificar que el atributo InResponseTo en el <SubjectConfirmationData> portador sea igual al ID de su mensaje <AuthnRequest> original, a menos que la respuesta no haya sido solicitada en cuyo caso no debería estar presente dicho atributo.
- Verificar que todas las aserciones en las que se confíe sean válidas en otros casos. Obsérvese que aunque pude haber varios elementos <SubjectConfirmation> de portador, para confirmar una aserción basta con la evaluación exitosa de uno sólo de dichos elementos, conforme a este perfil. No obstante, se debe evaluar independientemente cada aserción, si hubiera más de una.
- Si cualquier <SubjectConfirmationData> portador incluye un atributo Address, es probable que el proveedor de servicio deba comparar la dirección del cliente del agente de usuario con él.

- Toda aserción que no sea válida, o cuyos requisitos de confirmación de sujeto no puedan cumplirse, debería ser descartada y no debería emplearse para establecer un contexto de seguridad para el principal.
- Si un `<AuthnStatement>` que se utilice para establecer un contexto de seguridad para el principal contiene un atributo `SessionNotOnOrAfter`, se debería descartar el contexto de seguridad una vez se cumpla este tiempo, a menos que el proveedor de servicio restablezca la identidad del principal utilizando de nuevo este perfil. Obsérvese que si hay varios elementos `<AuthnStatement>`, se debería cumplir con el valor `SessionNotOnOrAfter` más cercano a la hora actual.

11.4.1.4.4 Reglas de procesamiento específicas al POST

Si se utiliza la vinculación POST HTTP para entregar la `<Response>`, cada aserción debe estar protegida por una firma digital, lo cual se puede lograr firmando cada elemento `<Assertion>` o el elemento `<Response>`.

El proveedor de servicio tiene que garantizar que no se reproduzcan las aserciones de portador, a través del mantenimiento del conjunto de los valores de `ID` utilizados el periodo del tiempo durante el cual se pueda considerar que la aserción es válida basándose en el atributo `NotOnOrAfter` en el `<SubjectConfirmationData>`.

BIBLIOGRAFÍA

- **FIPS-197** (2001), *Advanced Encryption Standard* (AES).
- **IETF RFC 1738** (1994), *Uniform Resource Locators (URL).*
- **IETF RFC 2256** (1997), *A Summary of the X.500 (96) User Schema for use with LDAPv3*.
- **IETF RFC 2279** (1998), *UTF-8, a transformation format of ISO 10646*.
- **IETF RFC 2743** (2000), *Generic Security Service Application Program Interface Version 2, Update 1*.
- **DCE**, *Distributed Computing Environment (DCE)*, Open Source. See http://www.opengroup.org/dce.
- **OASIS Authentication Context 2.0**, *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, 5 de noviembre de 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 1.1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, 22 de septiembre de 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 2.0**, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Conformance 2.0**, *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.00*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Glossary 2.0**, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Metadata 2.0**, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Errata Document 24**, *Revision 24 draft of the non-normative SAML V2.0 Errata document*, 27 de febrero de 2006, http://www.oasis-open.org/committees/download.php/16935/sstc-saml-errata-2.0-draft-24.pdf.
- **OASIS Protocol 1.0**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*, 5 de noviembre de 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Protocol 1.1**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*, 22 de septiembre de 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Protocol 2.0**, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS SAML 1.0**, *Security Assertion Markup Language (SAML) Version 1.0 Specification Set*, 5 de noviembre de 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS SAML 1.1**, *Security Assertion Markup Language (SAML) Version 1.1 Specification Set*, 22 de septiembre de 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML)*, 5 de noviembre de 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 1.1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML)*, 22 de septiembre de 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 2.0**, *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 de marzo de 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.1*, 24 de julio de 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.0*, 18 de febrero de 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **OASIS XACML 2.0**, *eXtensible Access Control Markup Language (XACML) V2.0*, 1 de febrero de 2005, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **SSL3**, *The SSL Protocol Version 3.0*. See http://wp.netscape.com/eng/ssl3/draft302.txt.
- **W3C Character Model** (2004), Working draft, 27 de octubre de 2005, *Character Model for the World Wide Web 1.0: Normalization.*

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	**Redes de datos, comunicaciones de sistemas abiertos y seguridad**
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2007