International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235.2

(09/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Systems aspects

H.323 security: Signature security profile

ITU-T Recommendation H.235.2

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	**H.230–H.239**
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.235.2

H.323 security: Signature security profile

Summary

This Recommendation describes an optional security profile for deploying digital signatures to secure the H.225.0 signalling.

In earlier versions of the H.235 subseries, this profile was contained in Annex E/H.235. Appendices IV, V, VI to H.235.0 show the complete clause, figure, and table mapping between H.235 versions 3 and 4.

Source

ITU-T Recommendation H.235.2 was approved on 13 September 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

Authentication, certificate, digital signature, encryption, integrity, key management, multimedia security, security profile.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2006

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

ITU-T Recommendation H.235.2

H.323 security: Signature security profile

1 Scope

This Recommendation describes an optional security profile for deploying digital signatures to secure the H.225.0 signalling.

2 References

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

– ITU-T Recommendation H.235 (1998), *Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals*.

– ITU-T Recommendation H.235.0 (2005), *H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems*.

– ITU-T Recommendation H.235.1 (2005), *H.323 security: Baseline security profile*.

– ITU-T Recommendation H.235.6 (2005), *H.323 security: Voice encryption profile with native H.235/H.245 key management*.

– ITU-T Recommendation H.245 (2005), *Control protocol for multimedia communication*.

– ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems*.

– ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.

– ITU-T Recommendation X.509 (2005) | ISO/IEC 9594-8:2005, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

– ITU-T Recommendation X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

– ITU-T Recommendation X.803 (1994) | ISO/IEC 10745:1995, *Information technology – Open Systems Interconnection – Upper layers security model*.

– ITU-T Recommendation X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview*.

– ITU-T Recommendation X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework*.

– ISO/IEC 9798-3:1998, *Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques*.

– IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

2.2 Informative references

[ISO/IEC 14888-3]	ISO/IEC 14888-3:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms*.
[PKCS]	PKCS #1 v2.0: *RSA Cryptography Standard;* RSA Laboratories; October 1, 1998; http://www.rsa.com/rsalabs/pubs/PKCS/index.html.
[PKCS]	PKCS #7: *Cryptographic Message Syntax Standard*, An RSA Laboratories Technical Note, version 1.5, Revised November 1, 1993; http://www.rsa.com/rsalabs/pubs/PKCS/index.html
[RFC1321]	IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm.*
[RFC3447]	IETF RFC 3447 (2003), *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*.

3 Terms and definitions

For the purposes of this Recommendation, the definitions given in clauses 3/H.323, 3/H.225.0 and 3/H.245 apply along with those in this clause. Some of the terms used in this Recommendation are also defined in ITU-T Recs X.800 | ISO 7498-2, X.803 | ISO/IEC 10745, X.810 | ISO/IEC 10181-1 and X.811 | ISO/IEC 10181-2.

3.1 certification authorities: Certification Authorities (CAs), when used in the context of electronic signature, certify public verification keys by issuing "Certificates".

3.2 certificate repositories: Certificate repositories (e.g., an X.500 Directory) hold user certificates and Certificate Revocation Lists (CRLs). They are trusted to make that information accessible but are not responsible for the content or accuracy of the information they receive from the CAs or the RAs.

3.3 digital signature: Is a cryptographic transformation (using an asymmetric cryptographic technique) of the numerical representation of a data message, such that any person having the signed message and the relevant public key can determine that:

i) the transformation was created using the private key corresponding to the relevant public key; and

ii) the signed message has not been altered since the cryptographic transformation.

3.4 on-line certificate status providers: The On-line Certificate Status Protocol (OCSP) enables applications to determine the revocation state of an identified certificate. OCSP may be used to satisfy some of the operational requirements of providing revocation information in a more timely way than is possible with CRLs. On-line certificate status providers can be seen as an alternative to the use of off-line CRLs.

3.5 proxy: The proxy is an intermediate H.323 entity similar to a gatekeeper. The proxy may be a separate network node or may be collocated with the functionality of an H.323 entity such as of the gatekeeper. The proxy may perform security tasks such as signature and certificate verification and access control.

3.6 registration authorities: Registration authorities act as intermediaries between users and CAs. They receive requests from users and transmit them to the CAs in an appropriate form.

3.7 time stamping authorities: Time stamping authorities are mandatory for non-repudiation in case of key loss or key compromise. In practice, they provide a counter-signature to anyone, including a reliable time, over a hash and a hash identifier.

3.8 trust service provider: An entity, which can be used by other entities as a trusted intermediary in a communication or verification process, or as a trusted information service provider.

This Recommendation uses the following terms for provisioning the security services.

3.9 authentication-only: This security service of the signature security profile supports user authentication where the user authenticates when correctly digitally signing some piece of data by the private key. Note that this security service does not provide countermeasures against arbitrary cut and paste, message manipulation or tampering attacks. Authentication-only may be useful for security proxies that verify authenticity of the message (data origin authentication) when forwarding the message to another destination (e.g., Gatekeeper).

NOTE – The forwarding usually changes certain parts of the message; thus end-to-end integrity cannot be realized.

Nevertheless, authentication-only can be applied on a hop-by-hop basis as well. Procedure III specifies this security service for an end-to-end scenario while procedure II specifies this security service for the hop-by-hop case.

3.10 authentication and integrity: This is a combined security service that supports message integrity in conjunction with user authentication. The user authenticates when correctly digitally signing some piece of data by the private key. In addition to that, the message is protected against tampering. Both security services are provided by the same security mechanism. Combined authentication and integrity is possible only on a hop-to-hop basis. Procedure II specifies this security service.

NOTE – When digital signatures are applied, a non-repudiation security service may be supported; this depends also on the settings of the key usage bits of the signing key in the certificate (see also RFC 3280).

4 Symbols and abbreviations

This Recommendation uses the following abbreviations:

ARQ	Admission Request
ASN.1	Abstract Syntax Notation One
CA	Certification Authority
CRL	Certificate Revocation List
DH	Diffie-Hellman
DNS	Domain Name Service
EP	Endpoint
EPID	Endpoint Identifier
GK	Gatekeeper
GKID	Gatekeeper Identifier
GRQ	Gatekeeper Request
ICV	Integrity Check Value
IP	Internet Protocol
ITU	International Telecommunication Union

LDAP	Light-weight Directory Access Protocol
LRQ	Location Request
MCU	Multipoint Control Unit
MD5	Message Digest 5
NAT	Network Address Translation
OID	Object Identifier
OCSP	Online Certificate Status Protocol
PKCS	Public-Key Crypto System
RA	Registration Authority
RAS	Registration, Admission and Status
RSA	Rivest, Shamir, Adleman
RTP	Real-Time Protocol
SHA	Secure Hash Algorithm
URL	Uniform Resource Locator

5 Conventions

In this Recommendation the following conventions are used:

– "shall" indicates a mandatory requirement.

– "should" indicates a suggested but optional course of action.

– "may" indicates an optional course of action rather than a recommendation that something take place.

The signature security profile may use the **voice encryption security profile** of H.235.1 for achieving voice confidentiality if necessary.

Procedures II and III specify how to implement the security services for different scenarios as hop-by-hop and end-to-end with different security mechanisms such as asymmetric cryptographic (digital signature) techniques.

While the message integrity service always provides message authentication, the reverse is not always true. For the authentication-only mode, the integrity assured spans only a certain subset of message fields. This applies to integrity services realized by asymmetric means (e.g., digital signatures). Thus, in practice, a combined authentication and integrity service uses the same key material without introducing a security weakness.

Moreover, all hop-by-hop security information is put into the **CryptoSignedToken** element. This information is recomputed at every hop according to procedure II.

End-to-end security information on the other hand (only possible when using the H.323 proxy and procedure III), basically computes similar information as put in the **CryptoSignedToken**, but stores that information in a separate **CryptoToken** of the message. This information is not changed in transit. A separate object identifier allows distinguishing between hop-by-hop and end-to-end **CryptoTokens**.

Asymmetric techniques using digital signatures may apply on a hop-by-hop and/or also on an end-to-end basis.

6 Overview

This Recommendation describes an optional security profile for deploying digital signatures to secure the H.225.0 signalling. H.323 security entities (terminals, gatekeepers, gateways, MCUs, etc.) may implement this signature security profile for improved security or whenever required.

The signature security profile mandates the GK-routed model and is based upon the H.245 tunnelling techniques; support for non GK-routed models is for further study.

The signature security profile is applicable for scaleable "global" IP telephony; this security profile overcomes the limitations of the simple, baseline security profile of H.235.1. For example, the signature security profile does not depend on the administration of mutual shared secrets of the hops in different domains. It provides tunnelling of H.245 messages for H.245 message integrity and also provisions for non-repudiation of messages. The signature security profile supports hop-by-hop security as well as true end-to-end authentication with simultaneous use of H.235 proxies or intermediate gatekeepers.

The features provided by these profiles include, for RAS, H.225.0 and H.245 messages:

- User authentication to a desired entity irrespective of the number of application level hops that the message traverses.

  NOTE 1 – "Hop" is understood here in the sense of a trusted H.235 network element (e.g., GK, GW, MCU, proxy, firewall). Thus, application level hop-by-hop security when used with symmetric techniques does not provide true end-to-end security between terminals.
- Integrity of all or critical portions (fields) of messages arriving at an entity irrespective of the number of application level hops that the message traverses. Integrity of the message itself using a strongly generated random number is also optional.
- Application level hop-by-hop message authentication, integrity and non-repudiation provide these security services for the entire message.
- Non-repudiation of messages exchanged between two entities irrespective of the number of application level hops that the message traverses can also be provided. Specifically, the non-repudiation is provided for critical portions (fields) of the message. For instance, this may be the case when an EP sends a SETUP message to its GK and the two (EP and GK) are separated by one or more proxies.

Several attacks are thwarted by providing the above security services in a suitable fashion. These include:

- Denial-of-service attacks: Rapid checking of digital signatures can prevent such attacks.
- Man-in-the-middle attacks: Application level hop-by-hop message authentication and integrity prevents against such attacks when the man in the middle is between an application level hop, say, a hostile router. When the man in the middle is an application level entity, such attacks are prevented by the presence of end-to-end user authentication and integrity for selected portions of the message.
- Replay attacks: Use of timestamps and sequence numbers prevent such attacks.
- Spoofing: User authentication prevents such attacks.
- Connection hijacking: Use of authentication/integrity for each signalling message prevents such attacks.

This security profile is applicable in environments with potentially many terminals where password/symmetric key assignment is not feasible, e.g., in large-scale or global-scale scenarios. The signature security profile provides additional security services for non-repudiation using digital signatures and certificates. The digital signatures could use SHA1 or MD5 hashing and provides authentication and/or integrity (see procedures II and III).

H.323 entities using authentication and integrity, or authentication-only on a hop-by-hop basis, shall use procedure II. H.323 entities using just authentication-only would not implement integrity. The authentication-only H.323 entities shall use procedure III for true end-to-end authentication.

This Recommendation may apply message integrity protection that spans the entire message. For H.225.0 RAS the integrity protection covers the entire RAS message; for call signalling this covers the entire H.225.0 call signalling message including the Q.931 headers.

The signature security profile allows to securely tunnel H.245 call control PDUs within H.225.0 facility messages. The H.245 key update and synchronization mechanisms require tunnelling, e.g., useful for very long duration calls.

NOTE 2 – Key-update for secure G.711 speech coding should occur latest after transmission of 2^{30} 64-bit blocks, i.e., more than 12 days of ongoing conversation.

The vertically shaded area (blue in the electronic copy) in Table 1 represents the scope of the signature security profile. When omitting the integrity indicated by the horizontally shaded area (green in the electronic copy), the authentication-only security profile results. An option within the signature security profile is to choose between RSA-SHA1 or RSA-MD5 digital signatures. The voice encryption security profile of H.235.6 (see 6.1/H.235.6) could be optionally used in conjunction with the signature security profile.

Table 1/H.235.2 – Signature security profile

Security services	Call functions			
	RAS	H.225.0	H.245 (Note)	RTP
Authentication	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	digital signature	digital signature	digital signature	
Non-repudiation	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	digital signature	digital signature	digital signature	
Integrity	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	digital signature	digital signature	digital signature	
Confidentiality				
Access control				
Key management	certificate allocation	certificate allocation		
NOTE – Tunnelled H.245 or embedded H.245 inside H.225.0 fast connect.				

NOTE 3 – The signature security profile has to be supported also by other H.235 entities (e.g., gatekeepers, gateways and H.235 proxies).

NOTE 4 – Available key usage bits in the certificate could also determine the security service provided by a terminal (e.g., non-repudiation asserted).

For authentication, the user should use a public/private key signature scheme. Such a scheme usually provides for better integrity and non-repudiation of the call.

This Recommendation does **not** describe procedures for:

- Registration, certification and certificate allocation from a trust centre and private/public key assignment, directory services, specific CA parameters, certificate revocation, key pair update/recovery and other certificate operational or management procedures such as certificate or public/private key and certificate delivery and installation in terminals.

Such procedures may happen by means that are not part of this Recommendation.

The communication entities involved are able to implicitly determine usage of either the H.235.1 baseline security profiles or this signature security profile by evaluating the signalled security object identifiers in the messages (**tokenOID**, and **algorithmOID**; see also clause 20).

The following procedures are described for use in this profile:

Procedure II is based on digital signatures using a private/public key pair for providing authentication, integrity and non-repudiation of RAS, Q.931 and H.245 messages. Terminals may use this method if non-repudiation and sophisticated integrity is required.

Depending on the security policy, authentication may be unilateral or mutual applying the authentication/integrity in the reverse direction as well and providing higher security thereby. The security policy of a terminal may allow "authentication-only" without computing cryptographic integrity (see clause 9).

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS/call signalling message received from a terminal/peer gatekeeper respond with a corresponding reject message indicating security failure by setting the reject reason to **securityDenial** or other appropriate security error code according to 11.1/H.235.0. Depending on the ability to recognize an attack, and the most appropriate way to react to it, a gatekeeper receiving a secured **xRQ** with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured **xRJ**, or may discard that message. The encountered security event should be logged. On the other hand, the endpoint shall discard the received unsecured message, time out and may retry once again by considering to choose different OIDs. Likewise, a gatekeeper receiving a secured H.225.0 SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured RELEASE COMPLETE and reason set to **securityDenied** or may discard that message. Similarly, the encountered security event should be logged.

There is implicit H.235 signalling for indicating use of procedure II and the applied security mechanism based upon the value of the object identifiers (see also clause 20) and the message fields filled in. Object identifiers are referenced symbolically through letters (e.g., "A") in this text.

This profile does not use the H.235 ICV fields; rather cryptographic integrity check values are put into the **signature** field of the **token** in the **cryptoSignedToken**.

6.1 H.323 requirements

H.323 entities that implement this signature profile are assumed to support the following H.323 features:

- Fast connect;
- GK-routed model.

7 Digital signatures with public/private key pairs details (procedure II)

The following procedures shall be adhered to if procedure II is employed for hop-by-hop security:

- SHA1 or MD5 along with the RSA algorithm should be used to generate the digital signature. Adherence to PKCS #1 and PKCS #7 facilitates interoperability in this regard.

  The **CryptoH323Token** field in each RAS/H.225.0 message shall contain the following fields:

  – **nestedCryptoToken** containing a **CryptoToken** which itself contains the **cryptoSignedToken** containing the following fields:

    - **tokenOID** set to:

      – "A", indicating that the authentication/integrity computation includes all fields in the H.225.0 RAS or call signalling message (see clause 11);

      – "B", indicating that the authentication/integrity computation includes only a subset of fields (see clause 10) in the RAS/H.225.0 message for authentication-only.

    - **token** containing the fields:

      – **toBeSigned** containing the **EncodedGeneralToken** which actually is a **ClearToken** with the following fields set:

        - **tokenOID** set to "S", indicating that **ClearToken** is being used for message authentication/integrity/non-repudiation;
        - **timeStamp** contains the time stamp;
        - **random** contains a monotonically increasing sequence number;
        - **generalID** contains the identifier of the recipient (only in case of unicast messages);
        - **sendersID** contains the identifier of the sender;
        - **dhkey**, used to pass the Diffie-Hellman parameters as specified in this Recommendation during **Setup** to **Connect**:

          – **halfkey** contains the random public key of one party;

          – **modsize** contains the DH-**prime** (see Table 4/H.235.6);

          – **generator** contains the DH-group (see Table 4/H.235.6).

          NOTE 1 – When the signature security profile is used without the voice encryption security profile then no Diffie-Hellman parameters should be sent and **dhkey** should be absent; **halfkey**, **modsize** and **generator** may be set to {'0'B,'0'B,'0'B}.

        - **certificate** contains the digital certificate of the sender where type indicates the certificate type ("V" for MD5-RSA certificates or "W" for SHA1-RSA certificates) and **certificate** carries the actual certificate (see clause 14).
        - **algorithmOID** set to:

          – "V" indicating the use of MD5-RSA signature;

          – "W" indicating the use of SHA1-RSA signature.

        - **params** set to NULL.
        - **signature** containing the signature computed using SHA1 or MD5 RSA on all the fields (if **tokenOID** is "A", see clause 11) or certain critical fields (if **tokenOID** is "B", see clause 10) of the H.225.0 RAS or call signalling message.

When **tokenOID** "A" is used for protection of tunnelled H323-UU-PDUs including all H.245 message contents, then the signature computation shall be done over the entire H.225.0 call signalling message with all fields according to the procedure described in clause 11. In case **tokenOID** "B" is used, authentication-only of the **CryptoToken** is achieved when applying the procedure III (see clause 10).

- An entity (which may be one or more application hops away) for whom the signature is meant, verifies the signature.

NOTE 2 – The recipient is able to detect usage of procedure II by evaluating the **algorithmOID** within the token of the **cryptoSignedToken** (detecting presence of "V" or "W").

8 Multipoint conferencing procedures

MCUs shall support secured distribution of certificates upon request from terminals by the tunnelled H.245 **ConferenceRequest** and **ConferenceResponse** commands as described in 8.8.1/H.235.6. This allows terminals to request certificates from other terminals in a multipoint conference environment and thereby obtain certainty about the other participants' identity in the conference.

ConferenceRequest conveys **requestTerminalCertificate** of which the following fields are set:

- **terminalLabel**: used as addressing means of the remote terminal through the MCU;
- **certSelectionCriteria**: the sender may request certificates only of specific types;
- **sRandom**: a random challenge generated by the requesting sender.

ConferenceResponse conveys **terminalCertificateResponse** of which the following fields are set:

- **terminalLabel**: allows association of the returned certificate to the terminal.
- **CertificateResponse**: conveys the response from the MCU with fields set to:
    - **terminalLabel**: identification of the remote terminal;
    - **certificateResponse**: this is actually an octet string ASN.1 encoded from the **EncodedReturnSig** as:
        - **generalID**: identification of the destination terminal;
        - **responseRandom**: random challenge value generated by the MCU;
        - **requestRandom**: **sRandom** played back;
        - **certificate**: conveys the returned certificate where **type** indicates the certificate type as OID and **certificate** carries the digital certificate (see clause 14).

9 End-to-end authentication (procedure III)

Figure 1 shows a scenario with proxies separating GKs and EPs where two different CryptoTokens are used for hop-by-hop as well as end-to-end authentication and/or hop-by-hop integrity. The **CryptoToken** for hop-by-hop authentication applies only to the leg between two entities and has to be recomputed on every other leg. On the other hand, the **CryptoToken** for end-to-end authentication is generated just once by the sending endpoint and is not changed in transit by intermediate nodes. Intermediate nodes may validate signatures and certificates conveyed in end-to-end **CryptoTokens** and should forward the **CryptoToken** in transit.

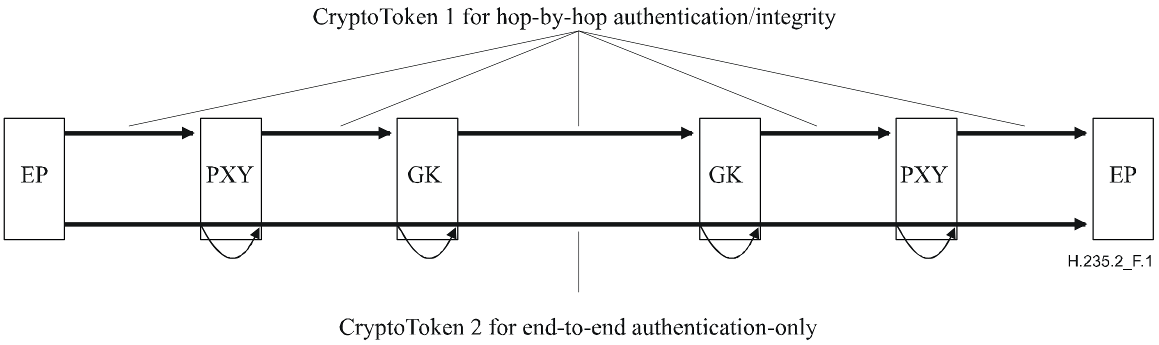
CryptoToken 1 for hop-by-hop authentication/integrity
EP
PXY
GK
GK
PXY
EP
H.235.2_F.1
CryptoToken 2 for end-to-end authentication-only

Figure 1/H.235.2 – Simultaneous use of hop-by-hop security and end-to-end authentication

NOTE 1 – The proxy may be a separate network node as shown in Figure 1 or may be collocated with the functionality of an H.323 entity, e.g., as part of the GK.

NOTE 2 – Depending on the signalled **tokenOID**, the proxy is able to determine whether the received **CryptoToken** is destined for the proxy ("S") or some other recipient ("R").

NOTE 3 – Due to the fact that intermediate entities change signalling message contents on every leg, end-to-end integrity is not possible.

For true end-to-end authentication across H.323 proxies or intermediate network elements, the sending endpoint/terminal shall compute a digital signature as follows.

The **CryptoH323Token** field in each RAS/H.225.0 message shall contain the following fields:

- **nestedCryptoToken** containing a **CryptoToken** which itself contains the **cryptoSignedToken** containing the following fields:
  - **tokenOID** set to:
    - "A", indicating that the hop-by-hop authentication/integrity computation includes all fields in the RAS/H.225.0 message (see clause 11);
    - "B", indicating that the authentication computation includes only a subset of fields (see clause 10) in the H.225.0 RAS or call signalling message for authentication only.
- **token** containing the fields:
  - **toBeSigned** containing the **ClearToken** field used with the following fields:
    - **tokenOID** set to "R" indicating that **ClearToken** is being used for authentication-only/non-repudiation on an end-to-end basis;

      NOTE 4 – Which security service is actually being applied depends also on the key usage bits in the certificate.
    - **random** contains a monotonically increasing sequence number;
    - **timeStamp** optionally for enhanced security only when the terminating end entities are time synchronized;
    - **generalID** contains the endpoint identifier of the recipient (only in case of unicast). In case of hop-by-hop this is the identifier of the next hop; in case of end-to-end this is the far-end endpoint identifier;
    - **sendersID** contains the endpoint sender;
    - **certificate** contains the digital certificate of the sender where **type** indicates the certificate type ("V" for MD5-RSA certificates or "W" for SHA1-RSA certificates) and **certificate** carries the actual certificate (see clause 14);

  - **dhkey**, used to pass the Diffie-Hellman parameters as specified in this Recommendation during **Setup** to **Connect**:
    - – **halfkey** contains the random public key of one party;
    - – **modsize** contains the DH-prime (see Table 4/H.235.6);
    - – **generator** contains the DH-group (see Table 4/H.235.6).

NOTE 5 – When the signature security profile is used without the voice encryption security profile, then no Diffie-Hellman parameters should be sent and **dhkey** should be absent; **halfkey**, **modsize** and **generator** may be set to {'0'B,'0'B,'0'B}.

- – **algorithmOID** set to:
  - "V", indicating the use of MD5-RSA signature;
  - "W", indicating the use of SHA1-RSA signature.
- – **params** set to NULL.
- – **signature** containing the signature computed using SHA1-RSA or MD5-RSA on all the fields (if **tokenOID** is "A") or certain critical fields (if **tokenOID** is "B") of the H.225.0 RAS or call signalling message.

The proxy may verify any obtained digital signature and/or certificate and may discard the message if not considered appropriate according to the local policy or the proxy shall forward the received **CryptoToken** further on. The proxy has to generate new H.235 signalling information elements for the hop-by-hop security according to procedures II or III.

The entity terminating the leg (this could be a terminal), should verify received security information in the **CryptoToken** and depending on the presence of end-to-end security elements, may additionally evaluate the end-to-end **CryptoToken** information. The exact verification procedures in a terminal or an intermediate H.323 entity may vary according to local policy.

10 Authentication-only

Terminals may choose to implement authentication-only (using OID "B"). In this case, the authenticator is computed just over a subset (**ClearToken** inside **CryptoToken**) of the RAS/H.225.0 message. Authentication-only may be useful for true end-to-end authentication (see clause 9). The following fields in the **ClearToken** structure are used as the subset:

- **tokenOID**: There is a separate token object identifier (tokenOID "B") for authentication-only implementation.
- **random**: The monotonically increasing sequence number.
- **timeStamp**: The time stamp.
- **generalID**: The identifier of the recipient (only in case of unicast messages). In case of hop-by-hop, this is the identifier of the next hop; in case of end-to-end, this is the far-end endpoint identifier.
- **sendersID**: The identifier of the sender.
- **dhkey**: The Diffie-Hellman parameters. This field and subfields are used during **Setup** to **Connect** messages.

The authenticator is computed over the **ClearToken** inside the **EncodedGeneralToken** (i.e., **ClearToken**) of the **token** of the **cryptoSignedToken**. The digital signature shall be computed over the ASN.1-encoded bitstring of **ClearToken**. Before computing the digital signature, the **tokenOID** in the **ClearToken** shall be set to {0 0}.

11 Authentication and integrity

For authentication and message integrity over all the ASN.1-coded message fields (using OID "A"), the procedure is the following.

The sender of a message shall compute the signature as follows:

1) Set the signature value to a specific default pattern with a fixed length (e.g., 1024 bits). This step shall reserve space for the maximum length of a digital signature, which is possible due to a given certificate. The exact bit pattern here does not matter, but a good choice is a unique bit pattern that does not occur in the remaining message.
2) ASN.1 encodes the entire message; for RAS this shall include the entire H.225.0 RAS message; for call signalling this shall include the entire H.225.0 call signalling message.
3) Locate the default pattern in the encoded message; overwrite the found bit pattern all with zero bits.

   NOTE 1 – This may involve some trial-and-error steps in the rare case when the default pattern occurs more than once in the message.
4) Compute the digital signature upon the ASN.1-encoded message using the method indicated by the **algorithmOID** "V" or "W" (see clause 12).
5) Substitute the default pattern in the encoded message with the computed digital signature value. In case the digital signature is shorter than the reserved space, leading zeros shall be put in front of the most significant bits of the signature value.

The recipient receives the message and then proceeds as follows:

1) ASN.1 decodes the message.
2) Extract the received digital signature value and keep it in a local variable SV.
3) Search and locate the signature value SV in the received encoded message.

   NOTE 2 – In rare circumstances where the signature value substring might occur several times in the entire message, steps 3-6 have to be iterated successively with a different starting search position.
4) Overwrite the bit pattern in the encoded message all with zeros.
5) Compute the digital signature upon the encoded message using the method indicated by the **algorithmOID** "V" or "W" (see clause 12).
6) Compare SV with the computed signature value. The message is considered uncorrupted and authentic only if both signature values are equal; in this case, the authentication is successful and the procedure stops.
7) Otherwise, repeat steps 3-7 by restoring SV to the previous location and search for another match. If none of the matches yield a correct signature value comparison, then the authentication has failed and the message has been altered (accidentally or intentionally) during transit, or for some other reason.

12 Computation of the digital signature

The input to the digital signature generation process is an ASN.1-encoded bit string and includes the result of the message digest calculation process and the signer's private key. The details of the digital signature generation depend on the signature algorithm employed; the certificate determines the signature algorithm to be applied; when the key usage extension in the certificate is present, the **digitalSignature** bit must be set for the key to be eligible for signing. The signature value generated by the signer is encoded as a bit string and carried in the **signature** field.

The method described in [PKCS #1, section E.8.1.1] for computing an RSA-based digital signature with appendix (RSASSA-PKCS1-v1_5-SIGN) along with the procedures OS2IP, RSASP1, I2OSP and the EMSA-PKCS1-v1_5-ENCODE method shall be used.

13 Verification of the digital signature

The input to the signature verification process includes the result of the message digest calculation process and the signer's public key. The recipient may obtain the correct public key for the signer by any means, but the preferred method is from a certificate obtained from the **certificate** field and then validated using the hash of the signer's certificate. The validation of the signer's public key may be based on the certification path processing (RFC 3280). The details of the signature verification depend on the signature algorithm employed.

The method described in [PKCS #1, section E.8.1.2] for verifying an RSA-based digital signature with appendix (RSASSA-PKCS1-v1_5-VERIFY) along with the procedures OS2IP, RSAVP1, I2OSP and the EMSA-PKCS1-v1_5-ENCODE method shall be used.

14 Handling of certificates

For verification of digital signatures, the receiving entity must have access to the sender's certificate that is signed by a recognized certification authority (CA). There are several possibilities as to how the recipient can access the sender's certificate:

- The certificate is included in the message exchange as described by procedures II and III; in this case, **certificate** holds the actual certificate and **type** holds OID "V" or OID "W".
- The recipient knows the certificate, possibly stored locally from an earlier exchange.
- Instead of including the certificate itself, the sender provides a URL where the certificate can be found. For this, **certificate** contains the URL and **type** is set to OID "P".
- The recipient obtains the certificate through some other means outside the scope of this Recommendation (e.g., LDAP directory lookup).

Whenever a digital certificate is conveyed in a message, the receiving entity (gatekeeper, endpoint) shall check the identity of the sender (gatekeeper, endpoint) against the identity of the certificate in order to prevent man-in-the-middle attacks.

For digitally signed messages sent from the gatekeeper to the endpoint, different possibilities exist for an endpoint to check the gatekeeper identity:

– If the hostname is available, for example, in the common name attribute of the subject field or of the subjectAltName field in the certificate, the endpoint may check this hostname against the gatekeeper identifier. Additionally, the endpoint may use DNS to query the associated IP address and check it against the gatekeeper's IP address as presented in the gatekeeper's signed response message.

– For example, the gatekeeper identifier may be constructed by the IP address (represented as a 4 byte value in network byte order) concatenated with other identifying information of the gatekeeper identifier, truncated to the maximum length of senders ID field, which carries the gatekeeper’s identity. The endpoint may additionally check the IP address belonging to the hostname against the IP address presented in the IP header of the response of the gatekeeper.

  NOTE – This method would not work as expected when Network address translation (NAT) devices are involved.

– If the hostname is not available in the certificate, the IP address, which would be part of the certificate (*iPAddress subjectAltName*), shall be taken directly to perform the checks stated above.

Users should carefully examine the certificate presented by the gatekeeper to determine if it meets their expectations. If the endpoint has external information as to the expected identity of the gatekeeper, the hostname check may be omitted. For instance, an endpoint may be connecting to a gatekeeper whose address and hostname are dynamic but the endpoint knows the certificate that the gatekeeper will present. In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man-in-the-middle attacks. In special cases, it may be appropriate for the endpoint to simply ignore the gatekeeper's identity, but it must be understood that this leaves the connection open to active attacks.

If the hostname does not match the identity in the certificate, user oriented endpoints shall either notify the user (endpoints may give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error. Automated endpoints shall log the error to an appropriate audit log (if available) and should terminate the connection (with a bad certificate error).

Automated endpoints may provide a configuration setting that disables this check, but shall provide a setting, which enables it.

Likewise, it is recommended that the gatekeeper perform an identity check for any digitally signed messages sent from the endpoint to the gatekeeper. How exactly the gatekeeper would implement such a checking is considered as a local matter and should be subject to implementation of the gatekeeper's security policy. As an example, one may imagine that the user name conveyed within the certificate may also be part of the H.323 identifier. Further on, the gatekeeper may crosscheck such identity information against locally administered/configured user data if available and may base a policy decision upon that.

If the gatekeeper has external information as to the expected identity of the endpoint, the hostname check may be omitted. For instance, a gatekeeper may be connecting to an endpoint whose address and hostname are dynamic, but the gatekeeper knows the certificate that the endpoint will present. In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man-in-the-middle attacks. In special cases, it may be appropriate for the gatekeeper to simply ignore the endpoint identity, but it must be understood that this leaves the connection open to active attack.

If the hostname does not match the identity in the certificate, the gatekeeper shall log the error to an appropriate audit log (if available) and should terminate the connection (with a bad certificate error).

If a subjectAltName extension of type dNSName is present, that shall be used as the identity. Otherwise, the (most specific) Common Name field in the Subject field of the certificate shall be used. Although the use of the Common Name is existing practice, it is deprecated and Certification Authorities are encouraged to use the dNSName instead.

Matching shall be performed using the matching rules specified by RFC 3280. If more than one identity of a given type is present in the certificate (e.g., more than one dNSName name), a match in any one of the set is considered acceptable. Names may contain the wildcard character * which is considered to match any single domain name component or component fragment. For example, *.a.com matches foo.a.com but not bar.foo.a.com. f*.com matches foo.com but not bar.com.

Procedures II and III provide means to carry a digital certificate. For efficiency, the digital certificates of the entities need to be transmitted at most only once if they are not already available in the entities through other means outside of this Recommendation. The certificate exchange thus should occur only at the beginning of a communication establishment: for RAS, this occurs either during gatekeeper discovery or, if this phase is omitted, then during gatekeeper registration. Similarly, for fast connect, where the certificate may be included in the initial call signalling messages but can safely be omitted in later call signalling messages.

For this security profile, an X.509v3 (1997) certificate shall be used. Other certificate formats are for further study.

15 Usage illustration for procedure II

Consider the case in Figure 2 where each entity has its own private-public key pair/certificate. An entity may also possess multiple key pairs. In the figure, an H.323 proxy separates EP1 from GK1.

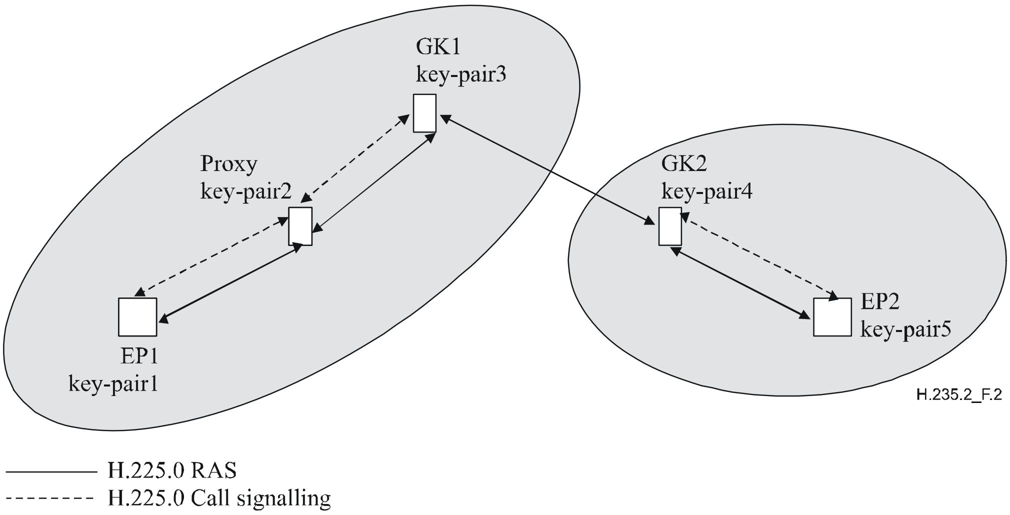

Figure 2/H.235.2 – Illustrating public-key usage in a GK-GK routed model

The H.323 proxy acts in a dual behaviour. On the one hand, the proxy terminates the authentication and integrity on each of its legs. The proxy actively includes the freshly computed authentication/integrity information in the outgoing RAS messages in a similar manner as described in procedure I of H.235.1. On the other hand, the proxy lets the end-to-end security information pass unmodified. The proxy may, however, verify received certificates and/or digital signatures in transit.

Below, we illustrate the procedure details for RAS, H.225.0 call signalling and H.245 message authentication, integrity and non-repudiation.

15.1 RAS message authentication, integrity and non-repudiation

Consider the case for a hop-to-hop communication where EP1 wishes to send a RAS message, say an **ARQ** message, to GK1. EP1 generates a timestamp and a sequence number and includes it in the **timeStamp** and **random** fields respectively, along with the proxy's alias in the **generalID** field and the **sendersID** of EP1. These fields are present in the **ClearToken** field of the **EncodedGeneralTokens** present in the **token** of the **cryptoSignedToken** of the **CryptoToken** field of the **cryptoH323Token** of the **ARQ** message. This **cryptoH323Token** is one of at least several tokens in the **cryptoTokens** sequence. The **tokenOID** within the **cryptoSignedToken** is set to "A", indicating that all the fields in the **ARQ** message are signed. The **token** in **cryptoSignedToken** has **algorithmOID** set to "V", indicating the use of MD5-RSA or **algorithmOID** set to "W", indicating the use of SHA1-RSA and **params** set to NULL. EP1 then computes the signature based on the given signature algorithm using its private key. The signature is computed over all the fields of the **ARQ** message when **tokenOID** is set to "A". EP1 includes the computed signature within **signature** in the **token** field of the **cryptoSignedToken** field of the **CryptoToken** present in the **cryptoH323Token** of the **ARQ** message and includes its certificate in the **certificate** field.

Similarly, for the end-to-end communication through a proxy, EP1 generates another **CryptoToken** containing a digital signature that covers certain critical fields (see clause 9) in the **ClearToken** of the **ARQ** message. The **tokenOID** in the **CryptoSignedToken** is set to "B", indicating authentication-only of that **ClearToken**; sets **tokenOID** in the **ClearToken** to "R", indicating end-to-end authentication, also **timeStamp**, **random**, **sendersID**, **generalID** and in case it is a **SETUP/CONNECT** also **dhkey**, sets in **token** the following fields: **algorithmOID** to "V" or "W", indicating the signature algorithm, **params** to NULL, and **signature** to the computed digital signature over the **ClearToken** fields. The **certificate** carries the digital certificate of EP1. The **ARQ** message is then sent to the proxy.

Upon receiving the **ARQ** message, the proxy verifies the signature of those tokens that are addressed to it (in this case, say, that with **tokenOID** "A"). This is based on several criteria that include:

- liveness of the timestamp, uniqueness of the **random**;
- identity of the **generalID** and own identifier;
- access permissions for the **sendersID**;
- matching of signature in **ARQ** message with that computed by GK1;
- verification of Diffie-Hellman parameters, e.g., testing whether the 1024-bit prime and generator are correct. Testing of whether the DH-parameters are secure is a time-consuming process and may be done only when local policy requires it;
- verification of the received certificate.

If the signature is successfully verified, the proxy computes a new signature to insert (replace) in the **ARQ** message before forwarding it to GK1 as follows. The proxy replaces the **timeStamp**, **random, sendersID** and **generalID** fields in the **ClearToken** (**toBeSigned**) field using values relevant to the proxy-GK1 leg. The **timestamp** field contains the current timestamp, the **random** field contains the next monotonically increasing sequence number for the proxy-GK1 leg, the **sendersID** of the proxy and the **generalID** field contains the alias of GK1. The proxy then computes a new signature for this **ARQ** message using its private key and signature algorithm, inserts it in **signature** within **token** and adds its **certificate.** The proxy also includes the received end-to-end **CryptoToken** with its **ClearToken** in the new outgoing message and passes the **ARQ** message on to GK1. The signature, computed by EP1 based on selected fields of the **ARQ** message (**tokenOID** of "B") and which was not meant for the proxy, is also passed untouched in the **ARQ** message to GK1.

Upon receiving the **ARQ** message, GK1 verifies the signatures, computes a new signature after modifying the **ClearToken** fields in **toBeSigned** suitably, inserts it in the **signature** field, adds its **certificate** and passes the **Setup** message on to EP2. Again, GK1 should forward any end-to-end information received in the separate **CryptoTokens** to the peer GK2 by including that information into a separate **CryptoToken** unmodified.

15.2 RAS authentication only

Consider the case for a hop-to-hop communication where EP1 wishes to send a RAS message, say an **ARQ** message, to GK1. EP1 generates a timestamp and a sequence number and includes it in the **timeStamp** and **random** fields respectively, along with the proxy's alias in the **generalID** field and the EP's id in the **sendersID**. These fields are present in the **ClearToken** field of **toBeSigned** present in the **token** in **cryptoSignedToken** of the **CryptoToken** field of the **cryptoH323Token** of the **ARQ** message. The **tokenOID** within the **cryptoSignedToken** is set to "B" indicating that only the specified subset fields in the **ClearToken** are signed. The **token** in **cryptoSignedToken** has **algorithmOID** set to "V" indicating use of MD5-RSA or "W" indicating use of the SHA1-RSA signature algorithm and **params** set to NULL. EP1 then computes the signature based on the signature algorithm using its private key. The signature is computed over the specified **ClearToken** fields of the **ARQ** message. EP1 includes the computed signature within **signature** in the **token** field of the **cryptoSignedToken** field of the **CryptoToken** present in the **cryptoH323Token** of the **ARQ** message and adds its **certificate**.

Similarly, EP1 generates another digital signature for end-to-end authentication that covers certain **ClearToken** fields in a separate **CryptoToken** in the **ARQ** message. This digital signature (identified by **tokenOID** of "V" or "W") is included. The **ARQ** message is then sent to the proxy.

Upon receiving the **ARQ** message, the proxy verifies the signature of those tokens that are addressed to it (in this case, say, that with **tokenOID** "B"). This is based on several criteria that include:

- liveness of the timestamp, uniqueness of the **random**;
- identity of the **generalID** and own identifier;
- access permissions for the **sendersID**;
- matching of signature in **ARQ** message with that computed by GK1;
- verification of the received certificate.

If the signature is successfully verified, the proxy computes a new signature to insert (replace) in the **ARQ** message before forwarding it to GK1 as follows. The proxy replaces the **timeStamp**, **random, sendersID** and **generalID** fields in the **ClearToken** field of **toBeSigned** using values relevant to the proxy-GK1 leg. The **timestamp** field contains the current timestamp, the **random** field contains the next monotonically increasing sequence number for the proxy-GK1 leg, and the **generalID** field contains the alias of GK1. The proxy then computes a new signature for this **ClearToken** using its private key and signature algorithm MD5-RSA or SHA1-RSA (**algorithmOID** ="V" or "W"), inserts it in **signature** within **token** of **cryptoSignedToken,** adds its **certificate** and passes the **ARQ** message on to GK1. The signature computed by EP1 based on selected **ClearToken** fields of the **ARQ** message (**tokenOID** of "B") and which was not meant for the proxy is also passed untouched in the **ARQ** message to GK1.

Upon receiving the **ARQ** message, GK1 verifies the signature, computes a new signature after modifying the **ClearToken** fields in **toBeSigned** suitably, inserts it in the **signature** field and passes the **Setup** message on to EP2. The end-to-end signature information from EP1 is included untouched in the **Setup** message.

15.3 H.225.0 message authentication, integrity and non-repudiation

The procedure for H.225.0 messages is identical to that for RAS messages. The only difference is that the set of fields that need to be signed has to be identified for each H.225.0 call signalling message when the **tokenOID** is set to "B".

15.4 H.245 message authentication and integrity

Consider the case where EP1 wishes to send an H.245 message, say a **TerminalCapabilitySet** message, to EP2. EP1 checks to see if an H.225.0 message needs to be sent to the proxy. If so, then the H.245 message is tunnelled within that H.225.0 message. The fields within the H.225.0 message are set as described earlier for the transmission of a H.225.0 message. Since the H.245 message is tunnelled, the **h323-uu-pdu** in the **h323-UserInformation** message has its fields set as follows:

- **h323-message-body** field is set to the H.225.0 message type that is being transmitted.
- **h245Tunnelling** set to TRUE.
- **h245Control** contains the H.245 PDU octet string.

However, if no H.225.0 message transmission is pending, then the H.245 message is tunnelled within an ad hoc H.225.0 **facility** message. The **h323-uu-pdu** in the **h323-UserInformation** message has its fields set as follows:

- **h323-message-body** field is set to **facility** which contains:
  - **reason** set to **undefinedReason**;
  - **tokens** and **cryptoTokens** set as for any H.225.0 message.
- **h245Tunnelling** set to TRUE.
- **h245Control** contains the H.245 PDU octet string.

The **facility** message is then transmitted by EP1 to the proxy.

In either case (whether a H.225.0 message transmission is pending or an ad hoc H.225.0 **facility** message is used), the proxy verifies the signature which is meant for it (in this case, depicted by **tokenOID** of "A") upon receiving the message. Then, if a H.225.0 message transmission is pending for the proxy-GK1 leg, the H.245 message is tunnelled within that message; otherwise, it is tunnelled within an ad hoc H.225.0 **facility** message. As in the case of transmission of any H.225.0 call signalling message, a new signature is computed for the H.225.0 message prior to its transmission from the proxy to GK1. The signature that was sent from EP1 to the proxy and that was not meant for the proxy is passed untouched by the proxy onto GK1.

This clause provides a summary of how, and by which means, the signature profile secures the various H.323 signalling messages.

16 H.235 version 1 compatibility

While these security profiles are developed with H.235 version 2 (ITU-T Rec. H.235v2) in mind, it is also possible to apply the security profiles for H.235 version 1 (ITU-T Rec. H.235v1) with some minor modifications. A recipient is able to detect presence of the sender's H.235 protocol version by evaluating the security profile object identifiers (see clause 20).

H.235 version 1 (ITU-T Rec. H.235v1) implementations:

- do not set or evaluate the **sendersID** in the **ClearToken**.

17 Multicast behaviour

H.225.0 multicast messages such as **GRQ** or **LRQ** shall include a **CryptoToken** according to the procedures II and III where the **generalID** is not set. When such messages are sent unicast, then the message shall include a **CryptoToken**.

18 List of secure signalling messages

18.1 H.225.0 RAS

H.225.0 RAS message	H.235 signalling fields	Authentication-only	Authentication and integrity	Non-repudiation
Any	cryptoTokens	Procedure II/III	Procedure II/III	Procedure II/III

NOTE – For unicast messages, procedures II or III shall be applied with the security fields in the **CryptoToken** used.

18.2 H.225.0 call signalling

H.225.0 call signalling message	H.235 signalling fields	Authentication-only	Authentication and integrity	Non-repudiation
Alerting-UUIE, CallProceeding-UUIE, Connect-UUIE, Setup-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	cryptoTokens	Procedure II/III	Procedure II/III	Procedure II/III

19 Usage of sendersID and generalID

The **ClearToken** holds **sendersID** and **generalID** fields. When identification information is available, the **sendersID** shall be set to the gatekeeper identifier (GKID) for the gatekeeper-initiated message and to the endpoint identifier (EPID) for the endpoint-initiated messages. When identification information is available, the **generalID** shall be set to the GKID for endpoint-initiated messages and to EPID for the gatekeeper-initiated messages. When the identification information is not available, or in case of broadcast/multicast is ambiguous, the field is missing or shall contain a null string. Table 2 summarizes the situation:

Table 2/H.235.2 – Usage of sendersID and GeneralID

Message	sendersID	generalID
Unicast **GRQ**	**EPID** if available, otherwise **NULL**	**GKID**
Multicast **GRQ**	**EPID** if available, otherwise **NULL**	
GCF, GRJ	**GKID**	**EPID** if available, otherwise **NULL**
Initial **RRQ**		**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ, UCF, URJ, BRQ, BCF, BRJ, DRQ, DCF, DRJ, NSM, RIP, SCI, SCR, XRS (EP-to-GK)	**EPID**	**GKID**
URQ, UCF, URJ, BRQ, BCF, BRJ, DRQ, DCF, DRJ, NSM, RIP, SCI, SCR, XRS (GK-to-EP)	**GKID**	**EPID**
ARQ, IRQ, RAI	**EPID**	**GKID**
ACF, ARJ, BCF, LCF, LRJ, IRR, IRQ, RAC, LCF, LRJ, IACK, INAK	**GKID**	**EPID**

Table 2/H.235.2 – Usage of sendersID and GeneralID

Message	sendersID	generalID
Unicast **LRQ** (EP-to-GK)	**EPID**	**GKID**
Unicast **LRQ** (GK-to-GK)	**GKID**	**GKID**
Multicast **LRQ**	**EPID**	
NOTE – GKID stands for gatekeeper identifier, EPID stands for endpoint identifier. Blank indicates a missing or null identification string.		

20 List of object identifiers

Table 3 lists all the referenced OIDs (see also [OIW] and [WEBOIDs]). There are object identifiers for H.235v1 and for H.235v2.

Table 3/H.235.2 – Object identifiers

Object identifier reference	Object identifier value(s)	Description

"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}
		Used in procedure II for the CryptoToken-tokenOID indicating that the signature includes **all** fields in the H.225.0 RAS or call signalling message (authentication and integrity).

"B"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 2}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 2}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 1 2}
		Used in procedure II for the CryptoToken-tokenOID indicating that the signature includes a **subset** of fields in the RAS/H.225.0 message (ClearToken) for authentication-only terminals without integrity.
		Used in H.235.1 procedure IA for the CryptoToken-tokenOID indicating that the hash includes a subset of fields in the RAS/H.225.0 message (ClearToken) for authentication-only terminals without integrity

"P"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 4}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 1 4}
		Used in procedures II or III to indicate that **certificate** carries a URL.

"R"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 3}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 1 3}
		Used in procedure II for the ClearToken-tokenOID indicating that the ClearToken is being used for end-to-end authentication/ integrity.

"S"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 7}
	{itu-t (0) recommendation (0) h (8) 235 version (0) 1 7}
		Used in procedure II this token OID indicates message authentication, integrity and non-repudiation.

Table 3/H.235.2 – Object identifiers

Object identifier reference	Object identifier value(s)	Description
"V"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4}	Used in procedure II or in procedure III as algorithm OID indicating use of MD5 RSA digital signature.
"W"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}	Used in procedure II or in procedure III as algorithm OID indicating use of SHA1 RSA digital signature.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2006