International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235.3

(09/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Systems aspects

H.323 security: Hybrid security profile

ITU-T Recommendation H.235.3

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.235.3

H.323 security: Hybrid security profile

Summary

The purpose of this Recommendation is to describe an efficient and scalable, PKI-based hybrid security profile for version 2 or higher of ITU-T Rec. H.235.0. The hybrid security profile contained herein takes advantage of the security profiles in ITU-T Recs H.235.1 and H.235.2 by deploying digital signatures from ITU-T Rec. H.235.2 and deploying the baseline security profile from ITU-T Rec. H.235.1.

In earlier versions of the H.235 sub-series, this profile was contained in Annex F/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.235.3 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 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

ITU-T Recommendation H.235.3

H.323 security: Hybrid security profile

1 Scope

The purpose of this Recommendation is to describe an efficient and scalable, PKI-based hybrid security profile for version 2 and higher of ITU-T Rec. H.235.0. The hybrid security profile contained herein takes advantage of the security profiles in ITU-T Recs H.235.1 and H.235.2 by deploying digital signatures from ITU-T Rec. H.235.2 and deploying the baseline security profile from ITU-T Rec. H.235.1.

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, version 1 (1998), *Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals*.
- ITU-T Recommendation H.235, version 2 (2000), *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.235.2 (2005), *H.323 security: Signature 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*.

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

- 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*.
- 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*.

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. Some of the terms used in this Recommendation are also as defined in ITU-T Recs X.800 | ISO 7498-2, X.803 | ISO/IEC 10745, X.810 | ISO/IEC 10181-1, X.811 | ISO/IEC 10181-2 and H.235.0.

4 Symbols and abbreviations

This Recommendation uses the following abbreviations:

ALG	Application Level Gateway
ASN.1	Abstract Syntax Notation One
BRJ	Bandwidth Reject
BRQ	Bandwidth Request
CA	Certification Authority
CRL	Certificate Revocation List
DB	Database
DH	Diffie-Hellman
DN	Distinguished Name
EP	Endpoint
GCF	Gatekeeper Confirm
GK	Gatekeeper
GKID	Gatekeeper Identifier
GKSP	Gatekeeper Security Processor

GRJ	Gatekeeper Reject
GRQ	Gatekeeper Request
HMAC	Hashed Message Authentication Code
ICV	Integrity Check Value
ID	Identifier
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
LRQ	Location Request
MCU	Multipoint Control Unit
MD5	Message Digest 5
NAT	Network Address Translation
OID	Object Identifier
PDU	Protocol Data Unit
PKI	Public Key Infrastructure
RAS	Registration, Admission and Status
RCF	Registration Confirm
RRJ	Registration Reject
RRQ	Registration Request
RSA	Rivest, Shamir and Adleman encryption algorithm
RTP	Real-time Transport Protocol
SHA	Secure Hash Algorithm
UDP	User Datagram Protocol
URQ	Unregistration Request
VoIP	Voice-over-IP

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 hybrid security profile uses terms and definitions from ITU-T Recs H.235.1 and H.235.2.

While the message integrity service always provides message authentication, the reverse is not always true. For the authentication-only mode, the integrity assured only spans 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.

This security profile is applicable in environments with potentially many terminals, where static password/symmetric key assignment is not feasible, e.g., in large-scale or global-scale scenarios.

Instead, this security profile assumes availability of a public-key infrastructure with assigned certificates and private/public-keys, directories, etc. In addition, this security profile deploys symmetric crypto techniques where applicable.

This security profile introduces the terms "first" message and "last" message sent. Security protection of the first message (and probably also for the last message) is different from security protection of the remaining other messages.

The "first message" sent is understood as a message that flows between two H.323 entities and establishes a security context. It makes symmetric key material available to both entities and, for example, marks the beginning of a call. For H.225.0 RAS, the first message is the RRQ and the related response message. For H.225.0 call signalling using fast start, the first message is SETUP and CONNECT.

The "last message" terminates the established security context. The established key material shall be destroyed. For H.225.0 RAS, the last message is the URQ and related response message, while for H.225.0 call signalling the last message is RELEASE-COMPLETE.

6 Overview

This Recommendation describes an efficient and scalable, PKI-based hybrid security profile deploying digital signatures from H.235.2 and deploying the baseline security profile from H.235.1. This Recommendation is suggested as an option. H.323 security entities (terminals, gatekeepers, gateways, MCUs, etc.) may implement this hybrid security profile for improved security or whenever required.

The notion of "hybrid" in this text shall mean that security procedures from the signature profile in ITU-T Rec. H.235.2 are actually applied in a lightweight sense and the digital signatures still conform to the RSA procedures. However, digital signatures are deployed only where absolutely necessary while highly efficient symmetric security techniques from the baseline security profile in ITU-T Rec. H.235.1 are used otherwise.

The hybrid 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 when strictly applying it. Furthermore, this security profile overcomes certain drawbacks of H.235.2, such as the need for higher bandwidth and increased performance needs for processing, when strictly applying it. For example, the hybrid security profile does not depend on the (static) administration of mutual shared secrets of the hops in different domains. Thus, users can more easily choose their VoIP provider. This security profile thus supports a certain kind of user mobility as well. It applies asymmetric cryptography with signatures and certificates only where necessary and otherwise uses simpler and more efficient symmetric techniques. It provides tunnelling of H.245 messages for H.245 message integrity and also implements some provisions for non-repudiation of messages.

The hybrid 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 features provided by this profile 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, or 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 (some) non-repudiation provide these security services for the entire message.
- Using the available public-key infrastructure, users can choose their service provider. Key-management for session key distribution is well integrated in the hybrid security profile.

Suitable provision of the above-described security services thwarts several types of attacks, including:

- *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 in an application-level hop, say, a hostile router.
- *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 assumes the GK-routed call model, where the fast connect call signalling method is applied. H.245 call control messages are securely tunnelled in H.225.0 call signalling messages and inherit thereby the H.225.0 security protection scheme.

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 for key-update FACILITY message to be signalled and is useful, for example, for very long duration calls.

The diagonally shaded area in Table 1 represents the security mechanisms that are used by the hybrid security profile.

NOTE 2 – RSA certificates with MD5 ([RFC1321]) hashing are not part of this security profile.

The voice encryption security profile of H.235.6 (see 6.1/H.235.6) could be optionally used in conjunction with the hybrid security profile. Its use is negotiated as part of the call set-up signalling.

Table 1/H.235.3 – Overview of the hybrid security profile

Security services	Call functions			
	RAS	**H.225.0**	**H.245 (Note 3)**	**RTP**
Authentication	RSA digital signature (SHA1)	RSA digital signature (SHA1)	RSA digital signature (SHA1)	
	▨ HMAC-SHA1-96	▨ HMAC-SHA1-96	▨ HMAC-SHA1-96	
Non-repudiation	(possible only on first message)	(possible only on first message)		
Integrity	RSA digital signature (SHA1)	RSA digital signature (SHA1)	RSA digital signature (SHA1)	
	▨ HMAC-SHA1-96	▨ HMAC-SHA1-96	▨ HMAC-SHA1-96	
Confidentiality				
Access control				
Key management	certificate allocation	certificate allocation		
	▨ authenticated Diffie-Hellman key-exchange	▨ authenticated Diffie-Hellman key-exchange		

NOTE 1 – The hybrid security profile has to be also supported by other H.235 entities (e.g., gatekeepers, gateways and H.235 proxies).

NOTE 2 – Available key usage bits in the certificate could also determine the security service provided by a terminal (e.g., non-repudiation asserted).

NOTE 3 – Tunnelled H.245 or embedded H.245 inside H.225.0 fast connect.

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.

For authentication, the user should use a public/private key signature scheme. Such a scheme usually provides for better integrity.

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, H.235.2 signature profile, or this hybrid security profile by evaluating the signalled security object identifiers in the messages (**tokenOID**, and **algorithmOID**; see also clause 10/H.235.2).

6.1 H.323 requirements

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

– Fast connect;

– H.245 tunnelling; and

– GK-routed model.

6.2 Authentication and integrity

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

Authentication and integrity: This is a combined security service that supports message integrity in conjunction with user authentication. The user authenticates when either correctly digitally signing some piece of data with the private key or when correctly applying a related, shared secret. 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.

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

The following procedures are described for use in this profile.

Procedure IV is based on digital signatures using a private/public key pair and deploying symmetric crypto techniques for providing authentication and integrity of RAS, Q.931 and H.245 messages. Terminals may use this method if efficient, scalable security is required.

Depending on the security policy, authentication may be unilateral or mutual (i.e., applying the authentication/integrity in the reverse direction as well, thereby providing higher security). The preferred security mode is to have mutual authentication.

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS/call signalling message received from a terminal/peer gatekeeper will respond with a corresponding reject message indicating security failure. This is done 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** and reject reason set to **securityDenial** or may discard that message. 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 call signalling SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured RELEASE COMPLETE and reject reason set to **securityDenied,** or it may discard that message whereas a gatekeeper receiving a secured H.225.0 FACILITY with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured FACILITY and reason set to **undefinedReason**, or it may discard that message. Similarly, the encountered security event should be logged. As part of the returned response, the sender may provide a list of acceptable certificates in separate tokens, in order to facilitate selection of an appropriate one by the recipient.

There is implicit H.235 signalling for indicating use of procedure IV and the applied security mechanism based upon the value of the object identifiers (see also clause 13) and the message fields filled-in. In this Recommendation, object identifiers are referenced symbolically through letters (e.g., "A").

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** when referring to H.235.2, or the integrity check values are put in the hash fields of the **CryptoToken** when referring to H.235.1.

7 Procedure IV

The following procedures shall be adhered to if procedure IV is employed for hop-by-hop security. This procedure unites procedure I of clause 7/H.235.1 and procedure II of clause 7/H.235.2.

For the first message, including corresponding response sent in each direction, H.235.2 procedure II (hop-by-hop authentication and integrity, see clause 7/H.235.2) shall be used with the following settings:

– OID "A1" instead of OID "A" and OID "S1" instead of OID "S". Use of these OIDs allows identifying the hybrid security profile.

– **algorithmOID** in **tokenOID** shall be set to "W" indicating use of RSA-SHA1 signature.

– **signature** shall contain an ASN.1 encoded RSA signature (see clause 12/H.235.2).

– **certificate** should contain the sender's user certificate if not available otherwise to the receiver; **type** shall hold OID "W" indicating an included RSA-SHA1 certificate or OID "P" (see clause 20/H.235.2) indicating that **certificate** holds an URL.

In a single administrative domain scenario, "the first message/response" is defined to equal the initial H.225.0 RAS message/response; this is usually either GRQ/GCF or RRQ/RCF. In a multi-administrative domain scenario, the first message/response within each domain is defined as above; the first message between the domains is defined as SETUP.

Whenever a digital certificate is conveyed in a message, the receiving entity shall check the identity of the sender against the identity of the certificate according to the procedure in clause 14/H.235.2 in order to prevent man-in-the-middle attacks.

Sender and recipient exchange and compute an authenticated Diffie-Hellman secret bit string. Table 4/H.235.6 provides an example of Diffie-Hellman group parameters and recommends taking the 1024-bit prime whenever possible, for security reasons. The Diffie-Hellman secret shall be computed for each leg, regardless of whether the voice encryption profile is deployed or not.

From the common bit string that both parties compute, both parties derive a 160-bit secret by taking the least significant 160 bits. The resulting 160-bit secret acts as the password/shared secret that is used in ITU-T Rec. H.235.1.

In a scenario with gatekeepers in distinct administrative domains, sender and receiver shall use two tokens in each direction for H.225.0 call signalling:

– One **ClearToken** inside **CryptoToken**, which is used to compute the media key that is shared among the terminals (see 8.5/H.235.6). This is only necessary if voice encryption is to be deployed.

– A separate **ClearToken** is used to compute a link key that is shared among the sender and receiver entities for protection of the signalling link. This link key replaces the shared password among the gatekeepers in H.235.1. The **tokenOID** of that **ClearToken** shall be set to "Q", indicating use of Diffie-Hellman and hybrid security profile. Computation of the link key proceeds in the same manner as computation of the media key (see 8.5/H.235.6).

NOTE 1 – For direct-routed environments, sender/receiver entities and terminals correspond. For GK-routed environments, the link key is shared hop-by-hop between each pair of peer gatekeepers, while the media key is shared on an end-to-end basis.

In GK-routed environments, the GK shall forward the received Diffie-Hellman token from the endpoint to the next hop.

For all but the very first message/response sent in each direction, H.235.1 procedure I (see clause 7/H.235.1) shall be used. This applies also in a scenario where multiple gatekeepers are located within an administrative domain. In this case, there is no need for asymmetric key management; instead, H.235.1 is sufficient.

This Recommendation may be used with H.235 version 1 systems when taking care of restricted use of sendersID and generalID, as described in clause 19/H.235.2.

It is anticipated that a gatekeeper should receive only a single **RRQ** including a DH-token with a digital signature from a particular fixed endpoint. However, lost or delayed **RCF**/**RRJ** messages may lead to retransmission using another signed **RRQ**.

In the case of the untimely arrival of the corresponding registration response, at the endpoint, the endpoint may attempt another try. For this, the endpoint shall use the most recent DH token but use a new sequence number and a new timestamp.

For a particular fixed endpoint, the gatekeeper shall use the most recently received signed **RRQ** message and derive the shared secret from that DH-token, regardless of whether or not the GK already has a shared secret available. Thus, the GK shall overwrite any existing shared secret with the newly derived secret. The GK shall respond with a signed **RCF** that holds the response DH-token. Preferably, the response DH token should be generated anew.

NOTE 2 – The recommended and preferred method for key update is by using the FACILITY message as defined in clause 9. However, it is recognized that key update may be achieved using another additive signed RRQ with a new DH-token.

NOTE 3 – A gatekeeper in possession of a shared secret shall respond to an HMAC-protected **RRQ** (according to ITU-T Rec. H.235.1) with an HMAC-protected response message.

8 Security association for concurrent calls

An optimization is provided for the case that a fixed pair of entities would process several independent calls in parallel using a single call signalling channel. Instead of establishing several link keys with Diffie-Hellman for each call, a security association is defined which spans multiple concurrent calls.

More precisely, the security association spans all calls between a fixed pair of entities as long as the call signalling channel is alive. Entities use the **multipleCalls** flag within Setup to indicate the capability of signalling multiple calls over a single call signalling connection (see 7.3/H.323).

If the single call signalling connection is used, then only one common link key needs to be established, see Figure 1.

On the other hand, if the **multipleCalls** flag within SETUP is not set, then a link key shall be individually computed anew for each call.

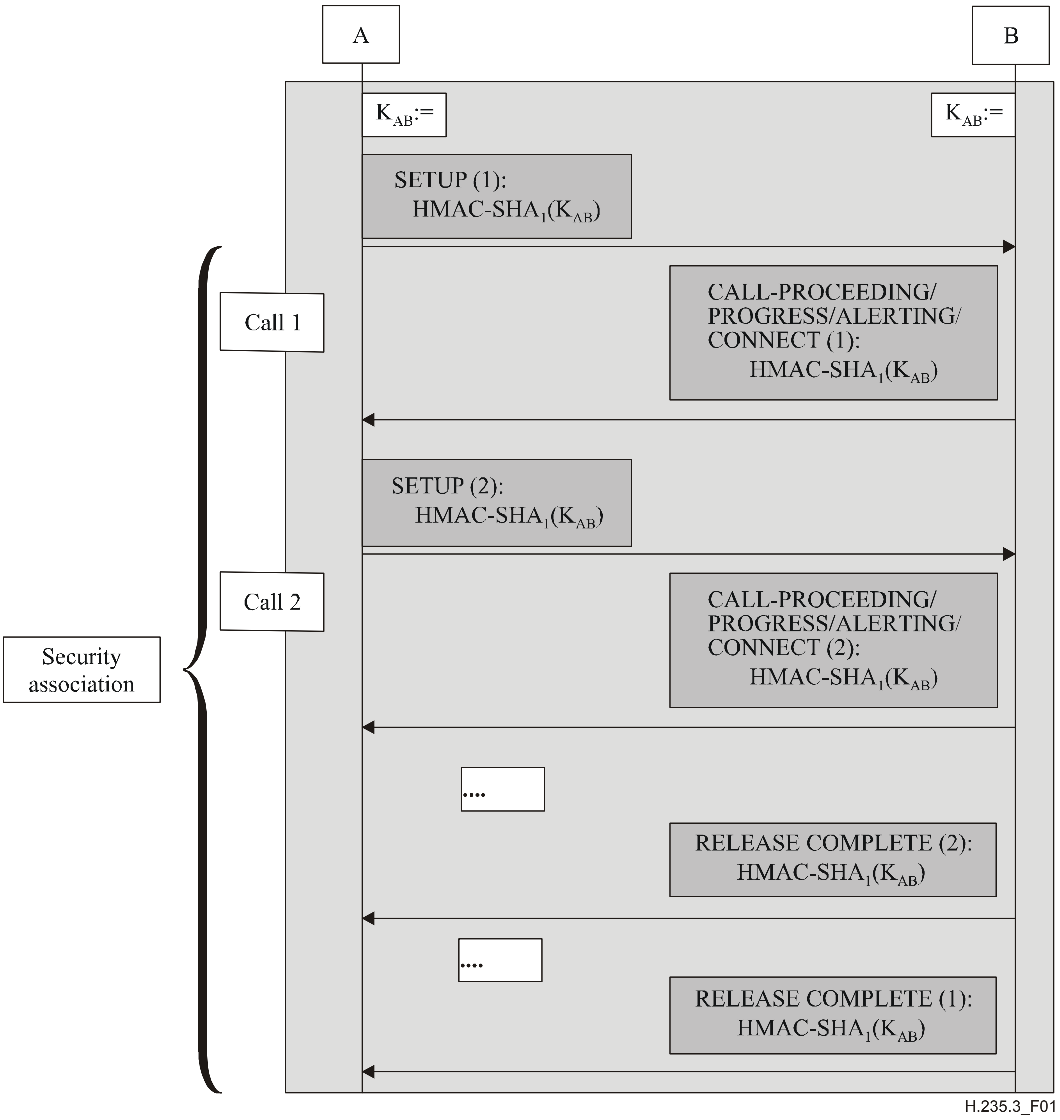


Figure 1/H.235.3 – Security association for concurrent calls

9 Key update

An optional key update procedure allows either communication entity (GK or terminal) to refresh the currently-used session key with a new one. Such a key update should be initiated by whichever entity feels a need for it. A key update may be motivated by a compromised session key, the perception that the session key has or will become insecure, or other security policy criteria. These aspects are all outside the scope of this Recommendation.

The initiator invokes the key update using the FACILITY message. The FACILITY message for key update conveys a new Diffie-Hellman token, an optional digital certificate, and a digital signature of the initiator. Upon reception of the FACILITY message, the recipient replies with a similar FACILITY message conveying his Diffie-Hellman token, an optional digital certificate, and a digital signature of the recipient. Upon completion of the key update procedure, initiator and responder shall use the computed new link key.

– **tokenOID** of the **ClearToken** within FACILITY shall be set to "Q" indicating use of Diffie-Hellman and hybrid security profile. Computation of the link key proceeds in the same manner as computation of the media session key (see 8.5/H.235.6).

The FACILITY message for key update purposes shall be protected according to H.235.2 procedure II. Any other FACILITY message without conveyed Diffie-Hellman token shall not be deployed for key update purposes and shall be protected according to clause 7/H.235.1 procedure I.

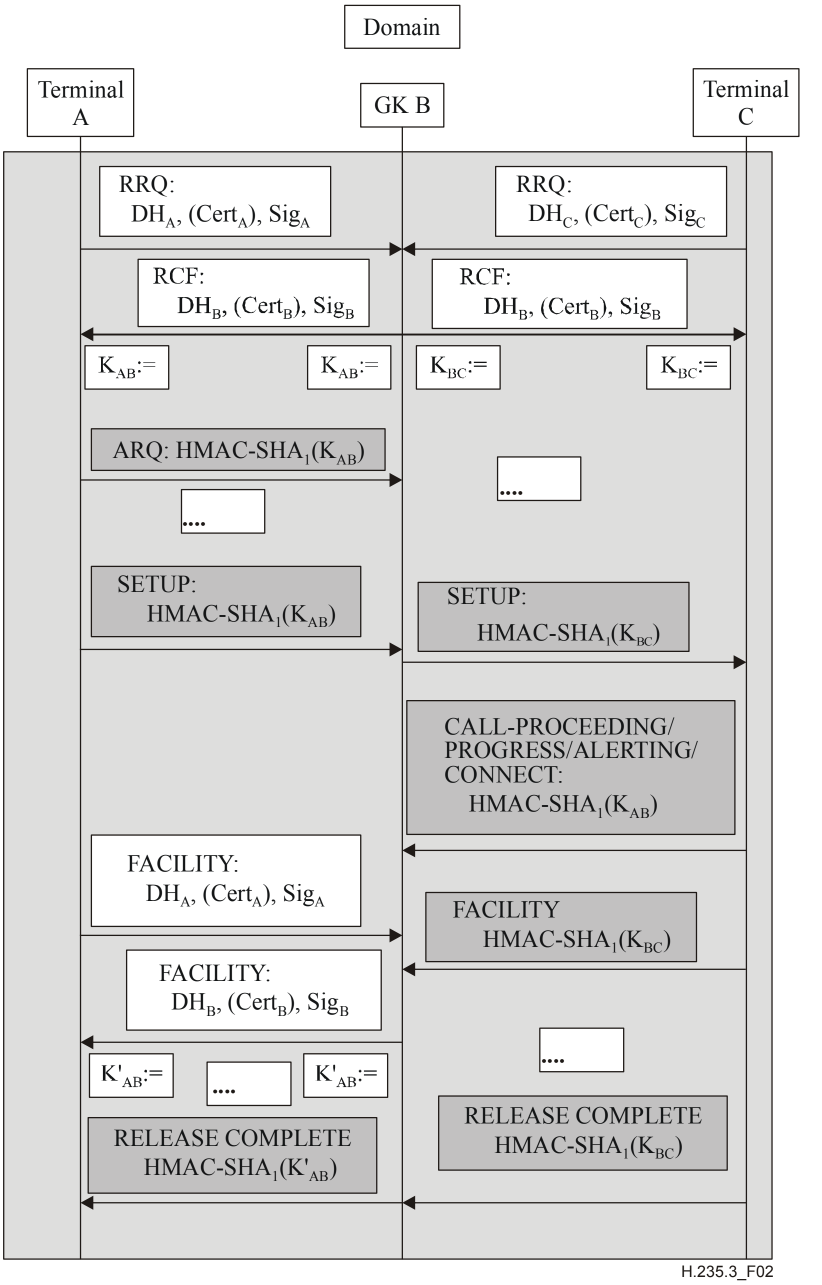
10 Usage of elliptic curve techniques

This is for further study.

11 Illustration examples

The flow diagrams in Figures 2 and 3 illustrate usage of this Recommendation in a basic message flow. Note that the diagrams do not show the complete message flow and that several messages are omitted for simplicity. Messages highlighted in light gray relate to the signature profile H.235.2, while dark gray messages relate to the baseline profile H.235.1. The figures emphasize the (most important) security parts of each message (H.235 CryptoTokens, Tokens) while omitting details.

The flow diagram in Figure 2 illustrates the basic message flow in a scenario with one gatekeeper within a single administrative domain. Assuming that the gatekeeper certificate is known to all the terminals involved, and that the terminals know the gatekeeper certificate likewise, there is no need to transmit the certificates in-band during the registration procedure.


Domain
Terminal A
GK B
Terminal C
RRQ: DH_A, $(Cert_A)$, Sig_A
RRQ: DH_C, $(Cert_C)$, Sig_C
RCF: DH_B, $(Cert_B)$, Sig_B
RCF: DH_B, $(Cert_B)$, Sig_B
K_{AB}:=
K_{AB}:=
K_{BC}:=
K_{BC}:=
ARQ: HMAC-SHA$_1$(K_{AB})
....
....
SETUP: HMAC-SHA$_1$(K_{AB})
SETUP: HMAC-SHA$_1$(K_{BC})
CALL-PROCEEDING/ PROGRESS/ALERTING/ CONNECT: HMAC-SHA$_1$(K_{AB})
FACILITY: DH_A, $(Cert_A)$, Sig_A
FACILITY HMAC-SHA$_1$(K_{BC})
FACILITY: DH_B, $(Cert_B)$, Sig_B
....
K'_{AB}:=
....
K'_{AB}:=
RELEASE COMPLETE HMAC-SHA$_1$(K'_{AB})
RELEASE COMPLETE HMAC-SHA$_1$(K_{BC})
H.235.3_F02

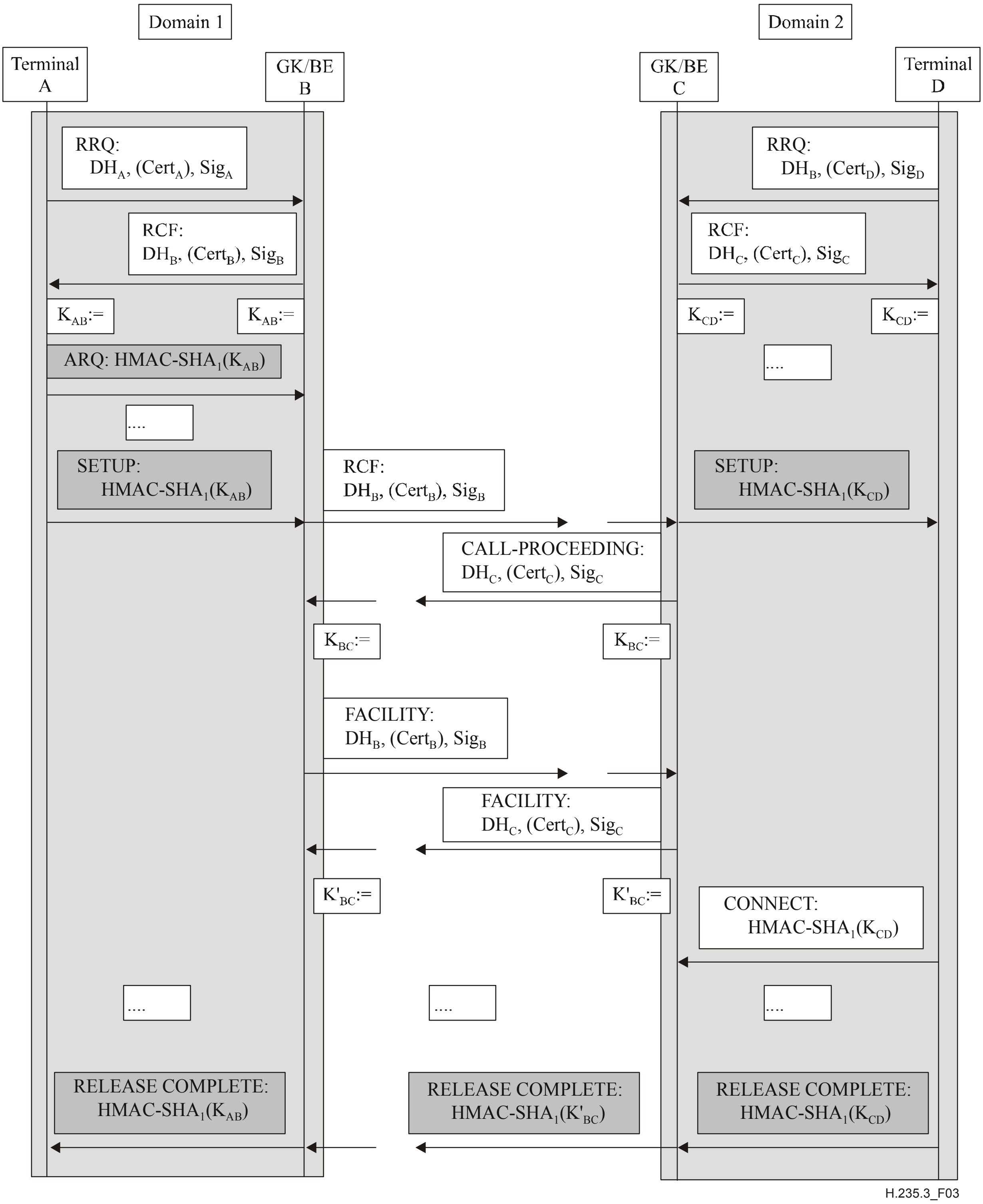
Cert	User certificate	K, K'	symmetric link key
DH_A	Diffie-Hellman Token g^a mod p	Sig	digital signature
DH_B	Diffie-Hellman Token g^b mod p		
EP	Endpoint (Terminal)		
GK	Gatekeeper		

Figure 2/H.235.3 – Flow diagram in a single administrative domain

NOTE 1 – Figures 2 and 3 also cover the fast start procedure when the call signalling messages SETUP and CALL PROCEEDING/PROGRESS/ALERTING/CONNECT include the faststart token (see 8.1.7/H.323). Otherwise, non-faststart mode is assumed according to 7.3.1/H.323. Figure 2 shows also the key update procedure between Terminal A and Gatekeeper B using FACILITY.

Figure 3 shows an example message flow in a scenario with different administrative domains. While the hybrid security profile is applied within each domain between terminal and gatekeeper as illustrated in Figure 2, the hybrid security profile may be applied also between both domains during the call establishment phase.

NOTE 2 – Figure 3 omits any communication among border elements (BE) and any communication between GK-to-BE. Figure 3 also shows the key update procedure between both domains using FACILITY.



Domain 1
Domain 2
Terminal A
GK/BE B
GK/BE C
Terminal D
RRQ: DH_A, $(Cert_A)$, Sig_A
RRQ: DH_B, $(Cert_D)$, Sig_D
RCF: DH_B, $(Cert_B)$, Sig_B
RCF: DH_C, $(Cert_C)$, Sig_C
K_{AB}:=
K_{AB}:=
K_{CD}:=
K_{CD}:=
ARQ: HMAC-$SHA_1(K_{AB})$
....
....
SETUP: HMAC-$SHA_1(K_{AB})$
RCF: DH_B, $(Cert_B)$, Sig_B
SETUP: HMAC-$SHA_1(K_{CD})$
CALL-PROCEEDING: DH_C, $(Cert_C)$, Sig_C
K_{BC}:=
K_{BC}:=
FACILITY: DH_B, $(Cert_B)$, Sig_B
FACILITY: DH_C, $(Cert_C)$, Sig_C
K'_{BC}:=
K'_{BC}:=
CONNECT: HMAC-$SHA_1(K_{CD})$
....
....
....
RELEASE COMPLETE: HMAC-$SHA_1(K_{AB})$
RELEASE COMPLETE: HMAC-$SHA_1(K'_{BC})$
RELEASE COMPLETE: HMAC-$SHA_1(K_{CD})$
H.235.3_F03

Figure 3/H.235.3 – Flow diagram in a multi-administrative domain

12 Multicast behaviour

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

13 List of secure signalling messages

Procedure IV deploys procedure I of H.235.1 or procedure II of H.235.2, depending on the scenario and on the actual message, as indicated below.

13.1 H.225.0 RAS

H.225.0 RAS message	H.235 signalling fields	Authentication and integrity	Non-repudiation
GatekeeperRequest, GatekeeperConfirm, GatekeeperReject if GK discovery is applied RegistrationRequest, RegistrationConfirm, RegistrationReject if GK discovery is not applied	CryptoToken, ClearToken	Procedure II	Procedure II
Any other RAS message (Note 2)	CryptoToken	Procedure I	
NOTE 1 – For unicast messages, procedure II shall be applied with the security fields in the CryptoToken used. NOTE 2 – GK discovery and multicast messages are not sent.			

13.2 H.225.0 call signalling (single administrative domain)

H.225.0 call signalling message	H.235 signalling fields	Authentication and integrity	Non-repudiation
Setup-UUIE, Connect-UUIE (Note 1), Facility-UUIE (Note 2), Alerting-UUIE, CallProceeding-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken, ClearToken	Procedure I	
Facility-UUIE (Note 3)	CryptoToken	Procedure II	Procedure II
NOTE 1 – Assuming that either message is the first in each direction. NOTE 2 – Not used for key update. NOTE 3 – Used for key update.			

13.3 H.225.0 call signalling (multi-administrative domain)

H.225.0 call signalling message	H.235 signalling fields	Authentication and integrity	Non-repudiation
Setup-UUIE, Connect-UUIE (Note 1), Alerting-UUIE (Note 2), CallProceeding-UUIE, Facility-UUIE (Note 3), Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE	CryptoToken, ClearToken	Procedure II	Procedure II
Alerting-UUIE (Note 4), CallProceeding-UUIE, Facility-UUIE (Note 5), Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken, ClearToken	Procedure I	Procedure I
NOTE 1 – Assuming that either message is the first in each direction. NOTE 2 – Any of those messages occurs as first message in either direction. NOTE 3 – Used for key update. NOTE 4 – Any of those messages does not occur as the first message in either direction. NOTE 5 – Not used for key update.			

14 List of object identifiers

Table 2 lists all the referenced OIDs.

Table 2/H.235.3 – Object identifiers

Object identifier reference	Object identifier value(s)	Description
"A1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 20}	Used as replacement for OID "A" in procedure II of ITU-T Rec. H.235.2 for the CryptoToken-tokenOID indicating that the RSA signature/hash includes all fields in the H.225.0 RAS or call signalling message (authentication and integrity).
"S1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 21}	Used as replacement for OID "S" in procedure II of ITU-T Rec. H.235.2 for the ClearToken-tokenOID indicating that the ClearToken is being used for message authentication and integrity. This OID in the end-to-end CryptoToken implicitly indicates also use of DH during fast start.
"Q"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 22}	Used in procedure IV indicating that the ClearToken on the hop-by-hop link carries a Diffie-Hellman token.
"W"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 23}	Used in procedure IV as algorithm OID indicating use of an RSA SHA1-based digital signature.

Appendix I

H.235.3 enabled Gatekeeper Security Processor

This informative appendix describes an implementation example for an H.235.3-enabled Gatekeeper Security Processor (GKSP) in conjunction with a gatekeeper. The purpose of the GKSP is to off-load certain H.235.3-specific security tasks such as execution of performance-expensive Diffie-Hellman operations, digital signature computations and verifications and X.509 certificate processing from a monolithic GK into a new and separate Gatekeeper Security Processor (GKSP) functional entity. There is at least one GKSP entity for each GK, yet one GK may serve also multiple GKSPs to increase scalability in the number of served endpoints and improve robustness of the entire system.

Figure I.1 shows such a decomposed GK architecture where the GKSP holds the H.235.3 security functions.

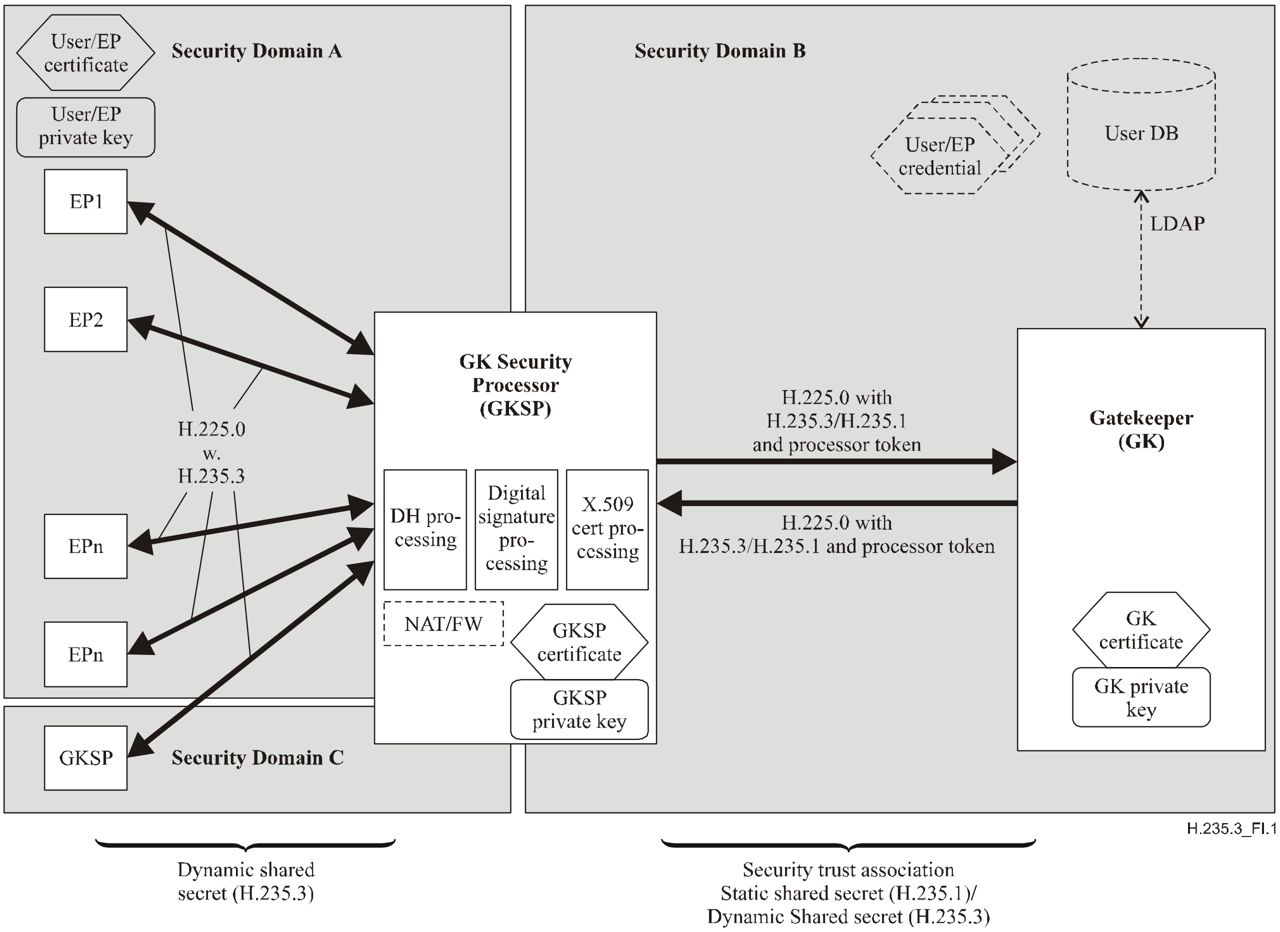


Figure I.1/H.235.3 – Gatekeeper Security Processor architecture

NOTE 1 – The GKSP may hold further useful functions such as for example a NAT (network address translation function), a firewall, an application-level gateway (ALG) etc.; such functions may be part of the security processing or may be kept as separate internal functions, yet such functions are not described in this clause and remain as for further study.

The GKSP serves a certain number of endpoints within an administrative security domain A. The GKSP may also communicate with another GKSP in some other administrative security domain C (not shown).

NOTE 2 – The three administrative security domains do not necessarily have to be distinct in practice. The GKSP may be placed entirely within administrative security domain B where the GK belongs to, or alternatively, the GKSP may be placed within security domain A, or within a separate, own security domain (not shown).

Using the GKSP, the Gatekeeper would not need to get involved in executing computational-expensive security operations. The GK still determines authorization and admission by matching an appropriate credential (e.g., alias name/DN name/certificate serial number, X.509 certificate) against the (internal/external) database holding the subscribed users with their permissions and credentials. Clause I.3 defines suitable credentials for use by an H.235.3-enabled GKSP.

NOTE 3 – A possible LDAP interface between the GK and subscriber/user database is not subject to this Recommendation. It is also left to the policy of the gatekeeper on which criteria and credentials (e.g., alias name/DN name/certificate serial number) to realize the access control decision. It is left to the discretion of such a user database, which credentials (alias name/DN name/certificate serial number) are stored therein.

NOTE 4 – The GKSP does not need to get involved in any matters regarding configuration or administration of users, subscribers and the GKSP does not need to access a user database.

NOTE 5 – Those endpoints which deploy H.235.3 and the GKSP typically also hold a root certificate (not shown in Figure I.1). The root certificate allows the entity to verify the certificate of the entity (EP, GKSP).

Communication between the GKSP and its GK, or among two GKSPs, is secured. For example, H.235.1 is applied when a statically configured shared secret is assumed. H.235.3 is applied that allows establishing a dynamic shared secret. In either case, GK and GKSP are assumed to have established a mutual trust relationship, be it either a static or a dynamic security association. The trust relationship may be chained when multiple GKSPs are involved.

Thus, the GK trusts the GKSP for executing the far-end authentication procedures and for correctly realizing the security procedures. The GKSP reports the outcome of its security processing in a simple security assertion using the processor token to the GK.

It is assumed that each H.235.3-enabled EP and the GKSP hold a X.509 certificate that trustfully binds the identity of the legitimate owner of the public key and a corresponding private key for signing.

NOTE 6 – The public key corresponding to the private key is not shown explicitly in Figure I.1; typically, the certified public key is conveyed within the X.509 user/EP certificate.

NOTE 7 – Not all certificates/private keys are shown for all endpoints/GKSP.

NOTE 8 – The GKSP certificate typically is a server certificate.

The GK is required to hold a distinct, unique GK certificate and a private key only if the GK deploys H.235.3 for communication with the GKSP.

The GKSP is a stateful proxy operating inbetween the endpoints and the GK, or inbetween two gatekeepers. There is at least one GKSP entity for each GK, yet one GK may also serve multiple GKSPs to increase scalability in the number of served endpoints, and to improve robustness of the entire system. It is possible to arrange linearly chained H.235.3-specific GKSP elements as shown in Figure I.2. It is possible to arrange H.235.3-specific GKSP elements in a hierarchical architecture as shown in Figure I.3.

There is at least one generic GKSP entity for each GK, yet one GK may also serve multiple generic GKSPs to increase scalability in the number of served endpoints and to improve robustness of the entire system. There may be one or more generic GKSPs between an endpoint and a GK: thus, linear or hierarchical, cascading configurations of several GKSP should in principle be possible. An endpoint always establishes a trust relation with its associated GK through one or more GKSPs. A single GK may have multiple trust relationships with multiple EPs.

Figure I.2 shows an architecture with linearly chained GKSP elements.

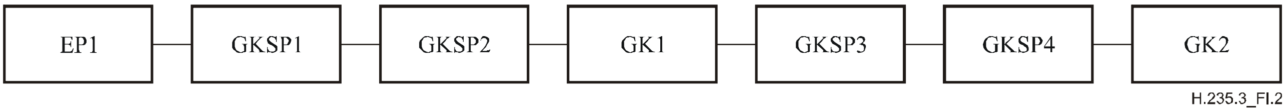


Figure I.2/H.235.3 – Chained GKSP architecture

In Figure I.2, GKSP1 authenticates the RRQ message received from EP1, while GK1 or GK2 decide upon the authorization of EP1. GKSP1, GKSP2 (resp. GKSP3 and GKSP4) rely H.323 signalling messages between EP1 and GK1 (resp. GK1 and GK2).

Figure I.3 shows an architecture with hierarchical, cascaded GKSPs elements.

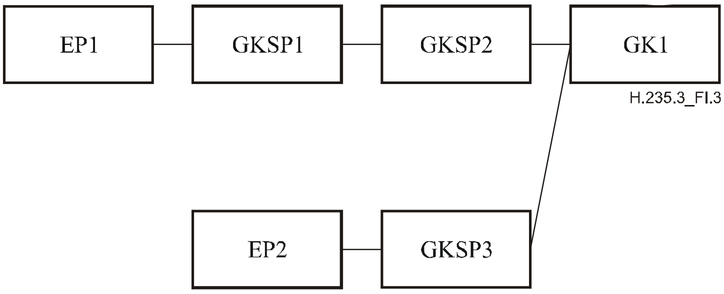


Figure I.3/H.235.3 – Hierarchical GKSP architecture

The GKSP has at least one IP address; typically a GKSP is an edge security device that is located at the border of two distinct administrative security domains. Thus, the GKSP may own two IP addresses, one IP address towards the H.323 endpoints/peer GKSP (administrative security domains A and C) and a different one internally towards the GK (administrative security domain B).

I.1 Discovery of a gatekeeper security processor

It is assumed that an H.323 endpoint is not required to know of presence of a GKSP. The endpoint may have configured the IP address of the GKSP as the GK contact point. In a scenario with a GKSP present, the EP behaves exactly as in a scenario without a GKSP. The EP may use the GK discovery phase using **GRQ** to locate its serving GKSP.

In the case where there is a GKSP that serves the requesting EP, the GKSP needs to find out if its GK supports a security processor.

In the case where the GKSP intends to use H.235.1 towards the GK but a shared secret has not been configured between the GKSP and GK, the GKSP returns **GRJ** to the endpoint with **reason** set to **securityDenial/securityDenied**. Otherwise, the GKSP forwards the **GRQ** and includes a processor ClearToken and sets the profile element with element ID 0 as defined in Table I.1. Since, in this case, the GK supports the GKSP, the GK returns a **GCF/GRJ** and includes a processor token.

In the case where the GKSP intends to use H.235.3 towards the GK, the GKSP forwards **GRQ** to the GK with the inclusion of a processor ClearToken and sets the profile element with element ID 0 as defined in Table I.1. A GKSP-enabled GK supporting this appendix responds by **GCF** and the inclusion of a processor ClearToken.

A GK that does not support a security processor or a GK that has not implemented this appendix would ignore the conveyed processor token and would respond by **GCF/GRJ**. The GKSP is able to recognize this situation, as the received **GRQ/GRJ** does not convey a processor token. The GKSP then sends a **GRJ** to the endpoint with **reason** set to **securityDenial/securityDenied**.

A GK that has received a **GRQ** from an endpoint directly without relay through a GKSP, and where the GK is aware of a GKSP, responds with an **GRJ** with **reason** set to **securityDenial**/**securityDenied** (without inclusion of a processor token).

I.2 Gatekeeper security processor operation

The GK security processor performs at least the following functions:

- Terminates the H.235.3 protocol to the H.323 endpoints, or to the peer GKSP, as defined by procedure IV.
- Runs the Diffie-Hellman H.235.3 protocol towards the H.323 endpoints/peer GKSP; i.e. performs the Diffie-Hellman modular-exponentiation operations.
- Performs verification of digital signatures received from H.323 endpoints or peer GKSP in H.235.3 secured messages.
- Security checks of received X.509 digital certificates: path verification, validity check, CRL check, etc.
- For message forwarding from the GKSP to the GK or to another GKSP, the GKSP generates new H.235 tokens (H.235.1 or H.235.3). The GKSP uses its GKSP identifier as **sendersID** and uses the gatekeeper identifier (GKID) as **generalID** in the baseline H.235 ClearToken.
- For messages received from the H.323 endpoint, the GKSP includes a processor token. For the initial **RRQ**/**GRQ** message, the processor token holds a security profile element with ElementID 0 that indicates the encountered authentication method. The GKSP may include a security profile element with ElementID 0 in any other H.225.0 RAS and/or Call Signalling message too.

Further, the processor token holds one or more security profile elements that convey the credentials.

Suitable credentials in the context of this appendix are:

- ElementID 1 for providing the subject found in an X.509 certificate;
- ElementID 2 for providing the subjectAltName found in an X.509 certificate;
- ElementID 3 for providing the serial number found in an X.509 certificate;
- ElementID 4 for providing the issuer name found in an X.509 certificate;
- ElementID 5 for providing the endpoint identifier of the H.323 terminal.

NOTE – The GK may additionally interpret the H.323 alias element in H.225.0 messages as a credential. Since the alias element is present anyway in messages, there is no need to define a separate alias element within a security profile element.

GKSP also includes a security profile element with ElementID 6 to indicate an encountered error. If the authentication between H.323 endpoint and GKSP has been successful, then the GKSP may include a security profile element with ElementID 6 to indicate that no security error has been encountered.

- In the case where the GKSP encounters security errors (digital signature wrong, certificate validation failed, etc.) in a message received from the H.323 endpoint or from the peer GKSP, the GKSP logs the error and forwards the message to the GK, includes a processor token with a security profile element of type ElementID 6 by indicating the type of error and lets the GK decide and react accordingly.
- In the case where the GKSP encounters security errors in a message received from the GK or from another GKSP, the GKSP logs the error and discards the message.
- Computes digital signatures for outgoing H.235.3 messages to the H.323 endpoints or peer GKSP.

- Relays any H.225.0 messages between H.323 endpoint and gatekeeper or GKSP back and forth and performs the following operations on tokens:
  - Communicates with its gatekeeper using the H.225.0 protocol where the H.235.3 tokens, received from the H.323 endpoints or from peer GKSP in the 1st handshake, are stripped off.
  - Verifies embedded H.235.1 tokens received from H.323 endpoints, or from a peer GKSP, and strips them off for further relay of messages to the gatekeeper.
  - Terminates H.235.1/H.235.3 protocol to its gatekeeper.
  - Includes H.235.1/H.235.3 tokens towards the H.323 endpoints or towards peer GKSP for outgoing messages.
  - Leaves the received H.225.0 messages received from H.323 EPs or GK basically intact; only rewrites tokens as defined above.
  - The H.225.0 protocol between GKSP and its GK is secured using either H.235.1 baseline security profile or H.235.3 hybrid security profile.
- In the case where GKSP and GK or the GKSP and another GKSP deploy the H.235.3 hybrid security profile, the GKSP either:
  a) runs the H.235.3 protocol towards the GK or GKSP for establishing a new dynamic key upon reception of the first message from the first endpoint or peer GKSP; or
  b) initiates the H.235.3 protocol towards the GK or GKSP for establishing a new dynamic key before any other H.323 endpoint or peer GKSP have started communication. This would permit a dynamic shared secret to be in place, ready for application to protect the first handshake messages received from an H.323 terminal or peer GKSP; this would further shorten the overall security association setup time.
- The GKSP does not forward any H.235.3-specific FACILITY messages for key update.
- In the case where GKSP and GK or the GKSP and another GKSP deploy the H.235.1 baseline security profile, the GKSP applies the static shared key for protection of the H.225.0 RAS and/or Call Signalling messages.
- Keeps track of security associations; i.e., establishes the DH shared secret; maintains the dynamic shared secrets. Depending on its security policy, the GKSP may invoke re-keying for the maintained dynamic shared secret(s) using FACILITY messages. Once the H.323 terminal or peer GKSP has de-registered, the GKSP should discard the dynamic shared key and consider no security association in place.
- Maps transport ports (EP-GKSP and GKSP-GK) for H.225.0 RAS and/or Call Signalling protocols one-to-one.

I.3 Processor token

Upon reception of an H.235.3 secured H.225.0 RAS and/or Call Signalling message with a conveyed X.509 certificate and digital signature, the GKSP removes the H.235.3 tokens and includes a separate processor token to the forwarded message to its GK or next GKSP (if any).

With the processor token, the GKSP reports the encountered authentication method, the encountered endpoint identifier, the encountered name in the certificate (name or subjectAltName), the encountered serial number in the X.509 certificate, the encountered issuer name in the X.509 certificate or an error indication. The processor token plays the role of a simple security assertion testifying to the asserted security relationship (successful or failed) between the GKSP and the H.323 endpoints towards the GK.

The GK is able to detect the presence of a GKSP by inspecting the received message and recognizing an included processor token. The GK interprets the absence of any processor token to indicate absence of any GKSP.

The processor token is a ClearToken with the following fields used:

– **tokenOID** holds an OID for "PT"; see Table I.2.

– **generalID** holds either:
  - the endpoint identifier of the H.323 endpoint in the case of a received H.235 secured message from an H.323 endpoint or holds;
  - the GK identifier in the case of a received H.235 secured message from the GK.

– **certificate** may optionally hold the received H.235.2/H.235.3 certificate received from the H.323 endpoint or peer GKSP. If this feature is implemented, the GKSP forwards the certificate to the GK.

  The usage of the subject/subjectAltName, or of the endpoint ID or of the certificate serial number or other lightweight credential should be preferred over including the entire certificate within the **certificate** field. This is because X.509 certificates tend to be a bigger piece of data and because of the potential problem of message fragmentation when certificates are included in UDP transported H.225.0 messages.

– **profileInfo** holds at least one profile element.

  The processor token may hold several profile elements; among them as listed in Table I.1:

Any other fields within GK security processor ClearToken remain unused.

Table I.1/H.235.3 – Specification of profile elements

ElementID value	Description	Specification
0	Indicates a profile element that conveys the authentication method. The usage of this profile element is mandatory for the initial handshake (GRQ or RRQ) and optional otherwise.	• **ParamS** remains unused. • **Element** holds an Element where **integer** is set to one of the following values to indicate the encountered authentication method at the H.323 endpoint or peer GKSP: 1) other, unspecified and non-standard authentication method; 2) none (i.e. no authentication); 3) H.235.1 shared secret, (undefined by this appendix); 4) H.235.2; 5) H.235.3; 6) H.235.5, (undefined by this appendix); 7) H.235.4, (undefined by this appendix); 8) H.530, (undefined by this appendix).

Table I.1/H.235.3 – Specification of profile elements

ElementID value	Description	Specification
1	Indicates a profile element that holds the **subject** of the received certificate. The usage of this profile element is optional.	• **ParamS** remains unused. • **Element** holds an Element where **name** or **octets** holds the **subject** of the received certificate. NOTE – The GKSP may need to re-encode the **subject** from the X.509 Name representation into an **octets** string or BMP **name** representation.
2	Indicates a profile element that holds the **subjectAltName** of the received certificate. The usage of this profile element is optional.	• **ParamS** remains unused. • **Element** holds an Element where **name** or **octets** holds the **subjectAltName** of the received certificate. NOTE – The GKSP may need to re-encode the **subjectAltName** from the X.509 Name representation into an **octets** string or BMP **name** representation.
3	Indicates a profile element that holds the serial number of the certificate. The usage of this profile element is mandatory.	• **paramS** remains unused. • **element** holds an Element where **integer** holds the **CertificateSerialNumber** of the received X.509 certificate.
4	Indicates a profile element that holds the issuer of the certificate. The usage of this profile element is mandatory.	• **paramS** remains unused. • **element** holds an Element where **name** or **octets** holds the **issuer** name of the received X.509 certificate. NOTE – The GKSP may need to re-encode the **issuer** name from the X.509 Name representation into an **octets** string or BMP **name** representation.
5	Indicates a profile element that holds the endpoint ID of the originating endpoint/terminal. The usage of this profile element is optional.	• **paramS** remains unused. • **element** holds an Element where **name** holds the endpoint identifier of the originating endpoint/ terminal.
6	Indicates a profile element that holds an error indication. The usage of this profile element is mandatory in any error case (> 0) but optional to indicate no error (0).	• **paramS** remains unused. • **element** holds an Element where **integer** holds one of the following encoded error values: 0: no error 1: securityDenied 2: securityWrongSyncTime 3: securityReplay 4: securityWrongGeneralID 5: securityWrongSendersID 6: securityMessageIntegrityFailed 7: securityWrongOID 8: securityDHmismatch 9: securityCertificateExpired 10: securityCertificateDateInvalid

Table I.1/H.235.3 – Specification of profile elements

ElementID value	Description	Specification
		11: securityCertificateRevoked 12: securityCertificateNotReadable 13: securityCertificateSignatureInvalid 14: securityCertificateMissing 15: securityCertificateIncomplete 16: securityUnsupportedCertificateAlgOID 17: securityUnknownCA 18: unspecified security error 19: GKSP not supported.

I.4 GKSP illustration example

This clause shows example message flow diagrams (see Figures I.4 and I.5) for a GK security processor operating within an administrative security domain. Note, that Figures I.4 and I.5 only show those messages that are crucial for H.235.3; in practice there may be many more H.225.0 RAS and/or Call Signalling messages.

In both figures, H.235.3-enabled H.323 terminal A and GKSP deploy the H.235.3 hybrid security profile; thus Terminal A and GKSP B do not share any static shared secret. In Figure I.4, GKSP and GK deploy the H.235.1 baseline security profile for protection of the H.225.0 RAS and Call Signalling messages. K_{BC} represents the static shared secret that GKSP B and GK C share.

Figure I.4 illustrates an entire call from terminal A through GKSP B and GK C. The call is GK-routed. At the beginning, terminal A and GKSP B negotiate a dynamic link key K_{AB} according to H.235.3 during RAS registration. For this, Terminal A generates the **RRQ** message that conveys the Diffie-Hellman half-key DH_A of A, holds A's certificate (optional), and A's digital signature upon all or parts of the **RRQ** message.

GKSP B receives the **RRQ** and verifies the digital signature. This includes validation and verification of the conveyed digital X.509 certificate (if included) against A's root certificate, path verification, CRL checks etc.

The GKSP forwards the **RRQ** message to GK C, adds a processor token (PT) including security profile elements:

– 0 indicating H.235.3 (5);

– 2 holding the subjectAltName of A's certificate;

– 3 the serial number of A's certificate;

– 5 the endpoint ID of A.

and applies the H.235.1 baseline security profile with the shared key K_{BC}; the HMAC-SHA1 integrity check is computed either on the entire **RRQ** message or only on certain parts of it.

In case the certificate validation or digital signature validation fails, GKSP B cannot authenticate and authorize Terminal A; GKSP then logs an error, and forwards the incorrect RRQ to GK C.

Gatekeeper C receives the **RRQ** message, verifies the integrity check by applying K_{BC} and processes the processor token PT with the included profile elements. If GK C is able to successfully validate the **RRQ**, GK C authorizes Terminal A. GK C then responds by an **RCF** that is sent to GKSP B.

GKSP B receives the **RCF**, recognizes that GK C has successfully authorized Terminal A and forwards an **RCF** to Terminal A, by computing and including its Diffie-Hellman half-key DH_B, its certificate (optional) and signs the **RRQ** (fully or partially) with its private key. Terminal A validates the authenticity of the received **RCF** message.

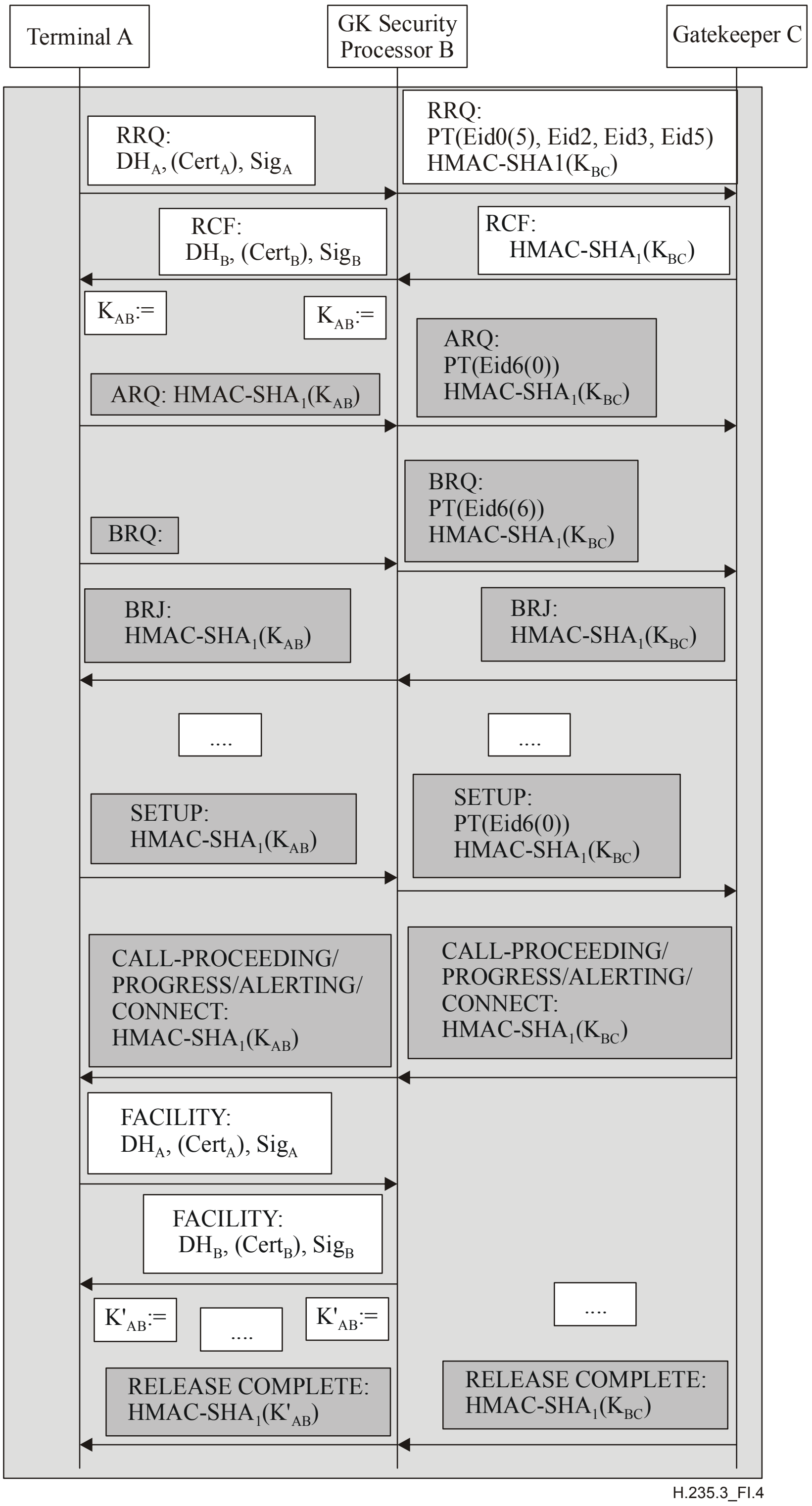
In the case where GKSP B successfully authenticates and authorizes Terminal A, GKSP B and Terminal A compute the dynamic shared secret K_{AB}. This dynamic shared secret represents the established trust relationship between Terminal A and GKSP B. Otherwise, and in the case where GK C does not authorize Terminal A, GKSP B forwards an **RCF** to Terminal A by computing and including its Diffie-Hellman half-key DH_B, its certificate (optional) and signs the **RRQ** (fully or partially) with its private key. Since Terminal A is not authorized, GKSP B does not keep K_{AB} any further. GKSP B may record the failed **RCF** in a log file.

Terminal A and GKSP B use this dynamic shared secret K_{AB} to further protect H.225.0 RAS and Call Signalling messages using H.235.1 baseline security profile. GKSP B and GK C use H.235.1 baseline security profile for protection of the all H.225.0 RAS and Call Signalling messages.

In the case where Terminal A receives an **RCF**, Terminal A does not proceed with the call setup.

Figure I.4 also shows an error case where terminal A (or someone else) sends an unprotected **BRQ** message to the GKSP; this message could also have been due to an attack where the attacker somehow removed or compromised the H.235.1 security protection. GKSP detects the failed integrity verification and forwards the **BRQ** message including a processor token to the GK, whereas the security profile element indicates securityMessageIntegrityFailed (6). The GK recognizes the security violation and does not authorize the bandwidth request by rejecting it with a **BRJ** reply.

At some point in time after the call has been established, terminal A decides to refresh the key K_{AB} by performing a key updating procedure for K_{AB} with GKSP B; K'_{AB} represents the new updated key. At the end of the call, it is terminated by GK C.


Terminal A
GK Security Processor B
Gatekeeper C
RRQ: DH_A, $(Cert_A)$, Sig_A
RRQ: PT(Eid0(5), Eid2, Eid3, Eid5) HMAC-SHA1(K_{BC})
RCF: DH_B, $(Cert_B)$, Sig_B
RCF: HMAC-SHA$_1$(K_{BC})
K_{AB}:=
K_{AB}:=
ARQ: HMAC-SHA$_1$(K_{AB})
ARQ: PT(Eid6(0)) HMAC-SHA$_1$(K_{BC})
BRQ:
BRQ: PT(Eid6(6)) HMAC-SHA$_1$(K_{BC})
BRJ: HMAC-SHA$_1$(K_{AB})
BRJ: HMAC-SHA$_1$(K_{BC})
....
....
SETUP: HMAC-SHA$_1$(K_{AB})
SETUP: PT(Eid6(0)) HMAC-SHA$_1$(K_{BC})
CALL-PROCEEDING/ PROGRESS/ALERTING/ CONNECT: HMAC-SHA$_1$(K_{AB})
CALL-PROCEEDING/ PROGRESS/ALERTING/ CONNECT: HMAC-SHA$_1$(K_{BC})
FACILITY: DH_A, $(Cert_A)$, Sig_A
FACILITY: DH_B, $(Cert_B)$, Sig_B
K'_{AB}:=
....
K'_{AB}:=
....
RELEASE COMPLETE: HMAC-SHA$_1$(K'_{AB})
RELEASE COMPLETE: HMAC-SHA$_1$(K_{BC})
H.235.3_FI.4

Cert	User certificate	GKSP	Gateekeeper Security Processor
DH_A	Diffie-Hellman Token g^a mod p	HMAC-SHA1	Computed integrity check value
DH_B	Diffie-Hellman Token g^b mod p	K, K'	Symmetric link key
Eid*n*	Security Profile ElementID with value *n*	PT	Processor Token
EP	Endpoint (Terminal)	Sig	Digital signature
GK	Gatekeeper		

Figure I.4/H.235.3 – Call flow with GK-security processor and H.235.1 message protection (GKSP-to-GK)

In Figure I.5, GKSP and GK deploy the H.235.3 hybrid security profile for protection of the H.225.0 RAS and Call Signalling messages. K_{BC} represents the dynamic shared secret that GKSP and GK first negotiate and then share for further use within H.235.1 baseline security profile for protection of the H.225.0 RAS and Call Signalling messages. Figure I.5 also shows a H.235.1-enabled H.323 terminal D that shares a static shared secret K_{DB} with its GKSP B.

Figure I.5 illustrates an entire call flow from terminal A through GKSP B and GK C. The call is GK-routed. In Figure I.5, it is assumed that terminal A is actually the first endpoint that registers at the GK through the GKSP.

Terminal A and GKSP B use this dynamic shared secret K_{AB} to protect further H.225.0 RAS and Call Signalling messages using H.235.1 baseline security profile. GKSP B and GK C use H.235.1 baseline security profile for protection of the further H.225.0 RAS and Call Signalling messages using the dynamic shared secret K_{BC}.

At the beginning, terminal A and GKSP B negotiate a dynamic link key K_{AB} according to H.235.3. During the first **RRQ**/**RCF** handshake between terminal A and GKSP where both entities establish a dynamic shared secret K_{AB}, the GKSP and the GK also deploy H.235.3 to establish a dynamic shared secret K_{BC}.

The GKSP forwards the **RRQ** message received from terminal A, adds a processor token including three security profile elements:

– 0 indicating H.235.3 (5);

– 3 the serial number of A's certificate;

– 6 indicating no error (0),

and applies the H.235.3 hybrid security profile. Since GKSP B and GK C do not yet share any shared secret, GKSP and GK run the H.235.3 protocol and establish a dynamic shared secret K_{BC}.

Sometime later, terminal D registers at the GKSP B using H.235.1-secured **RRQ**. GKSP B forwards this **RRQ** to the GK C and includes a processor token. The processor token conveys three security profile elements

– 0 indicating H.235.1 (3);

– 5 providing D's endpoint identifier;

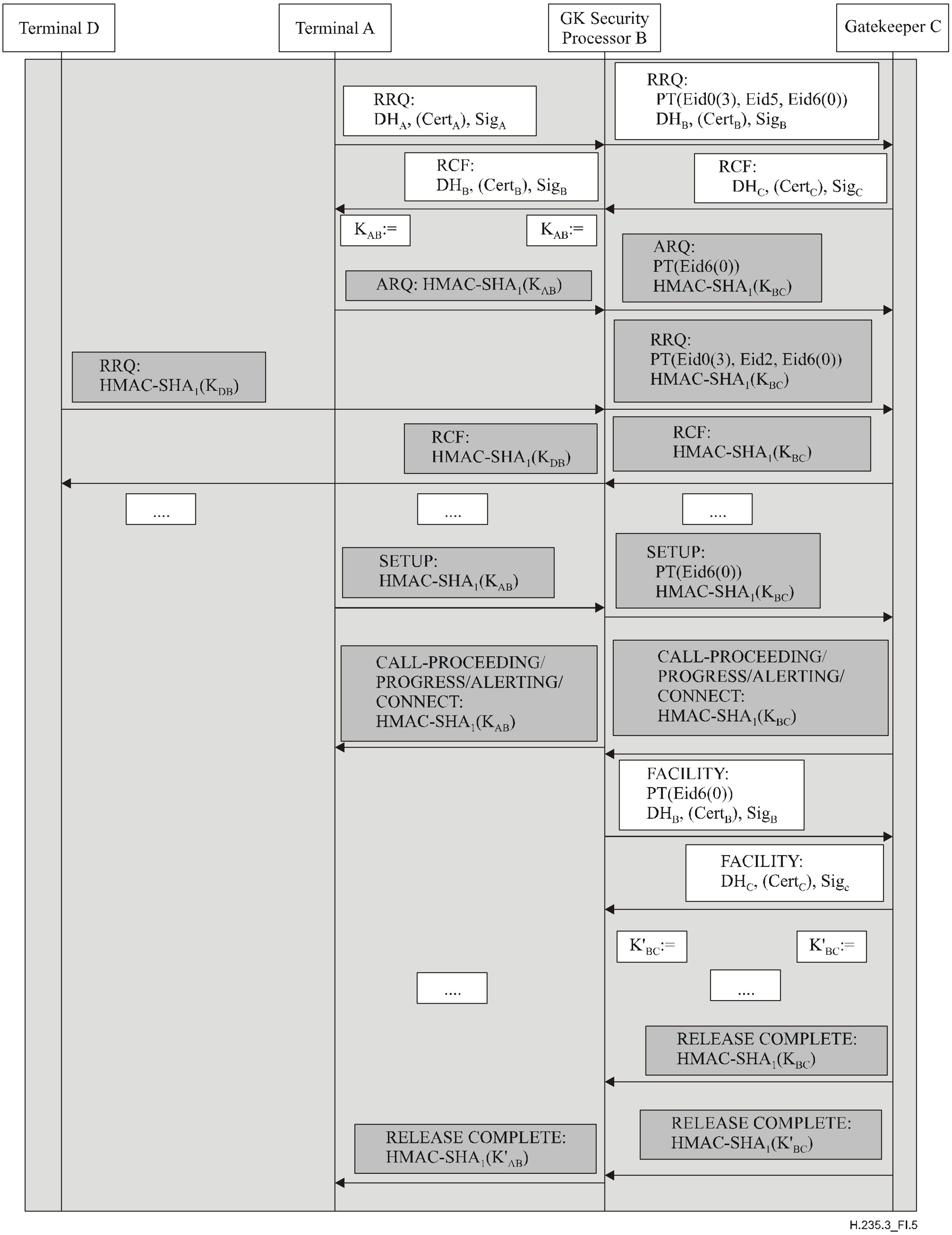
– 6 indicating no error (0),

and applies the H.235.3 hybrid security profile. Since the H.235.3 dynamic shared secret K_{BC} has already been established before, GKSP secures the forwarded **RRQ** message using H.235.1 by applying K_{BC}. GK C authorizes terminal D and replies with an **RCF** that the GKSP forwards to terminal D.

At some point in time after the call from terminal A has been established through GK C, GKSP B decides to refresh the key K_{BC} by performing a key updating procedure for K_{BC} with GK C; K'_{BC} represents the new updated key.

Figure I.5 also shows an error case where GKSP receives a RELEASE-COMPLETE message from the GK. GKSP B detects that integrity verification fails; this message does not use the current key. The message could have been replayed or manipulated by an attacker or the GK uses an old and expired key. GKSP B logs the security event and discards the message without forwarding it to terminal A.

At the end of the call, it is terminated by GK C.


Terminal D
Terminal A
GK Security Processor B
Gatekeeper C
RRQ: DH_A, ($Cert_A$), Sig_A
RRQ: PT(Eid0(3), Eid5, Eid6(0)) DH_B, ($Cert_B$), Sig_B
RCF: DH_B, ($Cert_B$), Sig_B
RCF: DH_C, ($Cert_C$), Sig_C
K_{AB}:=
K_{AB}:=
ARQ: PT(Eid6(0)) HMAC-SHA$_1$(K_{BC})
ARQ: HMAC-SHA$_1$(K_{AB})
RRQ: PT(Eid0(3), Eid2, Eid6(0)) HMAC-SHA$_1$(K_{BC})
RRQ: HMAC-SHA$_1$(K_{DB})
RCF: HMAC-SHA$_1$(K_{DB})
RCF: HMAC-SHA$_1$(K_{BC})
....
....
....
SETUP: HMAC-SHA$_1$(K_{AB})
SETUP: PT(Eid6(0)) HMAC-SHA$_1$(K_{BC})
CALL-PROCEEDING/ PROGRESS/ALERTING/ CONNECT: HMAC-SHA$_1$(K_{AB})
CALL-PROCEEDING/ PROGRESS/ALERTING/ CONNECT: HMAC-SHA$_1$(K_{BC})
FACILITY: PT(Eid6(0)) DH_B, ($Cert_B$), Sig_B
FACILITY: DH_C, ($Cert_C$), Sig_c
K'_{BC}:=
K'_{BC}:=
....
....
RELEASE COMPLETE: HMAC-SHA$_1$(K_{BC})
RELEASE COMPLETE: HMAC-SHA$_1$(K'_{BC})
RELEASE COMPLETE: HMAC-SHA$_1$(K'_{AB})
H.235.3_FI.5

Cert	User certificate	GK	Gatekeeper
DH_A	Diffie-Hellman Token g^a mod p	GKSP	Gatekeeper Security Processor
DH_B	Diffie-Hellman Token g^b mod p	HMAC-SHA1	Computed integrity check value
DH_C	Diffie-Hellman Token g^c mod p	K, K'	Symmetric link key
Eid*n*	Security Profile ElementID with value *n*	PT	Processor Token
EP	Endpoint (Terminal)	Sig	Digital signature

Figure I.5/H.235.3 – Call flow with GK-security processor and H.235.3 message protection (GKSP-to-GK)

I.5 List of object identifiers

Table I.2 lists the referenced OID that is to be used in conjunction with Table I.1.

Table I.2/H.235.3 – Object identifiers used by Appendix I

Object identifier reference	Object identifier value(s)	Description
"PT"	{itu-t (0) recommendation (0) h (8) 235 version (0) 4 15}	Used to indicate the GK processor Clear token for communication from a GKSP to a GK.

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