INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235

Corrigendum 1
(01/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS
Infrastructure of audiovisual services – Systems aspects

Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals

Corrigendum 1

ITU-T Recommendation H.235 (2003) – Corrigendum 1

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

Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals

Corrigendum 1

Summary

Version 3 of ITU-T Rec. H.235 supersedes ITU-T Rec. H.235 version 2 featuring a procedure for encrypted DTMF signals, object identifiers for the AES encryption algorithm for media payload encryption, the enhanced OFB (EOFB) stream-cipher encryption mode for encryption of media streams, an authentication-only option in Annex D for smooth NAT/firewall traversal, a key distribution procedure on the RAS channel, procedures for more secure session key transport and more robust session key distribution and updating, procedures for securing multiple payload streams, better security support for direct-routed calls in a new Annex I, signalling means for more flexible error reporting, clarifications and efficiency improvements for fast start security and for Diffie-Hellman signalling along with longer Diffie-Hellman parameters and changes incorporated from the ITU-T Rec. H.323 implementors guide.

Amendment 1 extended version 3 of ITU-T Rec. H.235 by inclusion of new Annex H and by extending the functionality of Annex I. The ASN.1 changes are added in support of Annex H, they may be used by any other purpose as identified by the ClearToken **profileInfo**. This amendment also include<u>d</u> some corrections to and updates the ITU-T Rec. H.235 version 3 text.

Corrigendum 1 aligns the specification of the pseudo-random function defined in B.7 with the pseudo-random function defined in RFC 3830, corrects editorial defects in Figures F.2 and F.3 and corrects a couple of defects throughout Annex I.

Source

Corrigendum 1 to ITU-T Recommendation H.235 (2003) was approved on 8 January 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

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.

CONTENTS

Page

ITU-T Recommendation H.235

Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals

Corrigendum 1

…

2 References

…

– IETF RFC 3830 (2004), MIKEY: Multimedia Internet KEYing.

…

Annex B

H.323 specific topics

…

B.7 Pseudo-Random Function (PRF)

This clause defines a pseudo-random function for the purpose of deriving dynamic keys from a static key material and a random value.

NOTE – This PRF is identical to the MIKEY PRF (see [MIKEY]/RFC 3830 section 4.1.2~~xxxx~~).

The key derivation method has the following input parameters:

- *inkey*: the input key to the derivation function.
- *inkey_len*: the length in bits of the input key.
- *label*: a specific label, dependent on the type of the key to be derived and the random **challenge** value.
- *outkey_len*: desired length in bits of the output key.

The pseudo-random function has the following output:

- *outkey*: the output key of desired length.

This PRF shall use the PRF as is defined in RFC 3830 section 4.1.2. ~~Let HMAC (see RFC 2104) be the SHA1- [(see ISO/IEC 10118-3)] based message authentication function. Similar to RFC 2246, define:~~

~~$P(s, label, m) =$ HMAC $(s, A_1 \| label) \|$~~

~~HMAC $(s, A_2 \| label) \| \ldots$~~

~~HMAC $(s, A_m \| label)$~~

~~where:~~

~~A_0 = label,~~

~~A_i = HMAC (s, A_{i-1}).~~

~~While SHA1 ISO/IEC 10118-3 is the default, HMAC using other hash functions may be used; this is left as for further study.~~

~~The following procedure describes a pseudo-random function, denoted *PRF(inkey, label)*, applied to compute the output key, outkey:~~

- ~~let n = *inkey_len/512*, rounded up to the nearest integer;~~
- ~~split the *inkey* into *n* blocks, *inkey* = $s_1 \| \ldots \| s_n$, where all s_i, except possibly s_n, are 512 bits each;~~
- ~~let *m* = *outkey_len/160*, rounded up to the nearest integer.~~

~~Then, the output key, *outkey*, is obtained as the *outkey_len* most significant bits of:~~

~~*PRF(inkey, label)* = $P(s_1, label, m)$ *XOR* $P(s_2, label, m)$ *XOR* ... *XOR* $P(s_n, label, m)$.~~

…

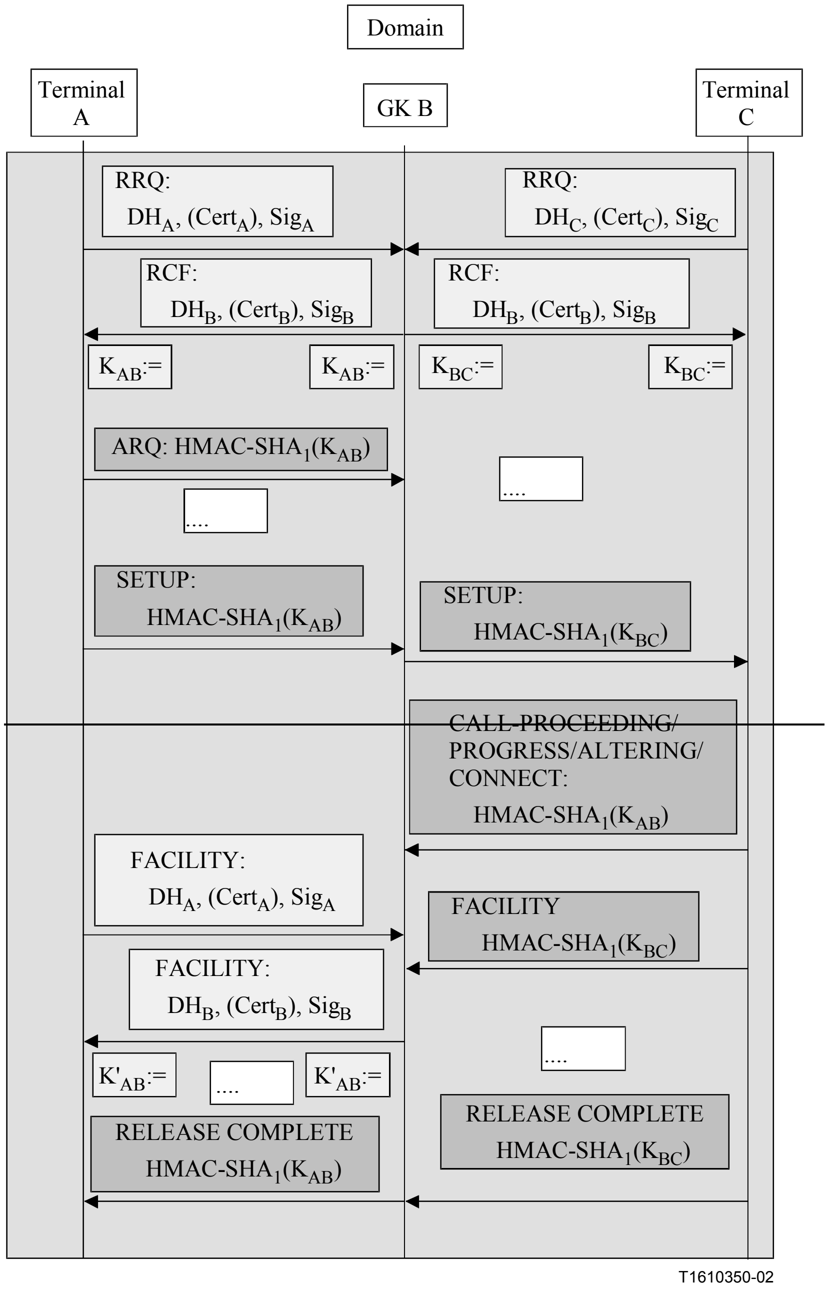
Annex F

Hybrid security profile

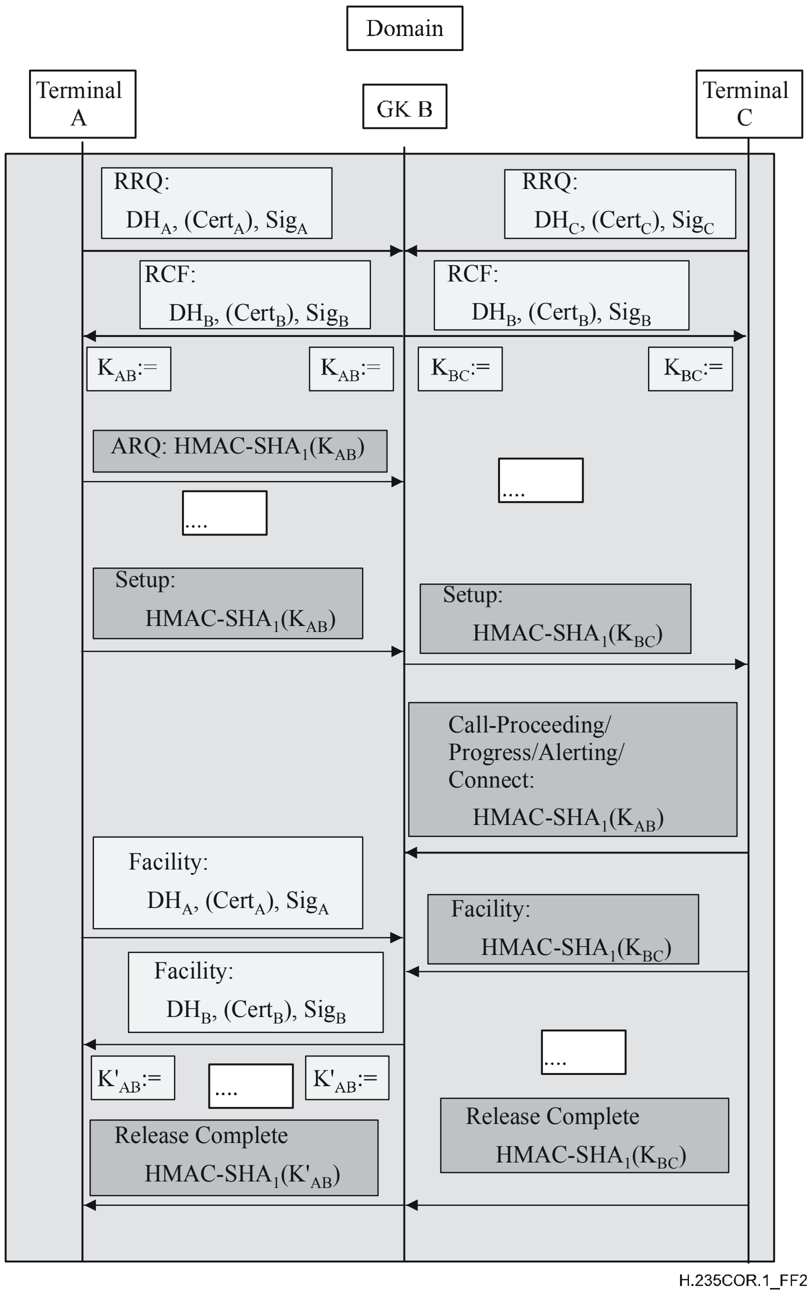
…

F.10 Illustration examples

…


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/ALTERING/ 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-$SHA_1(K_{BC})$
T1610350-02

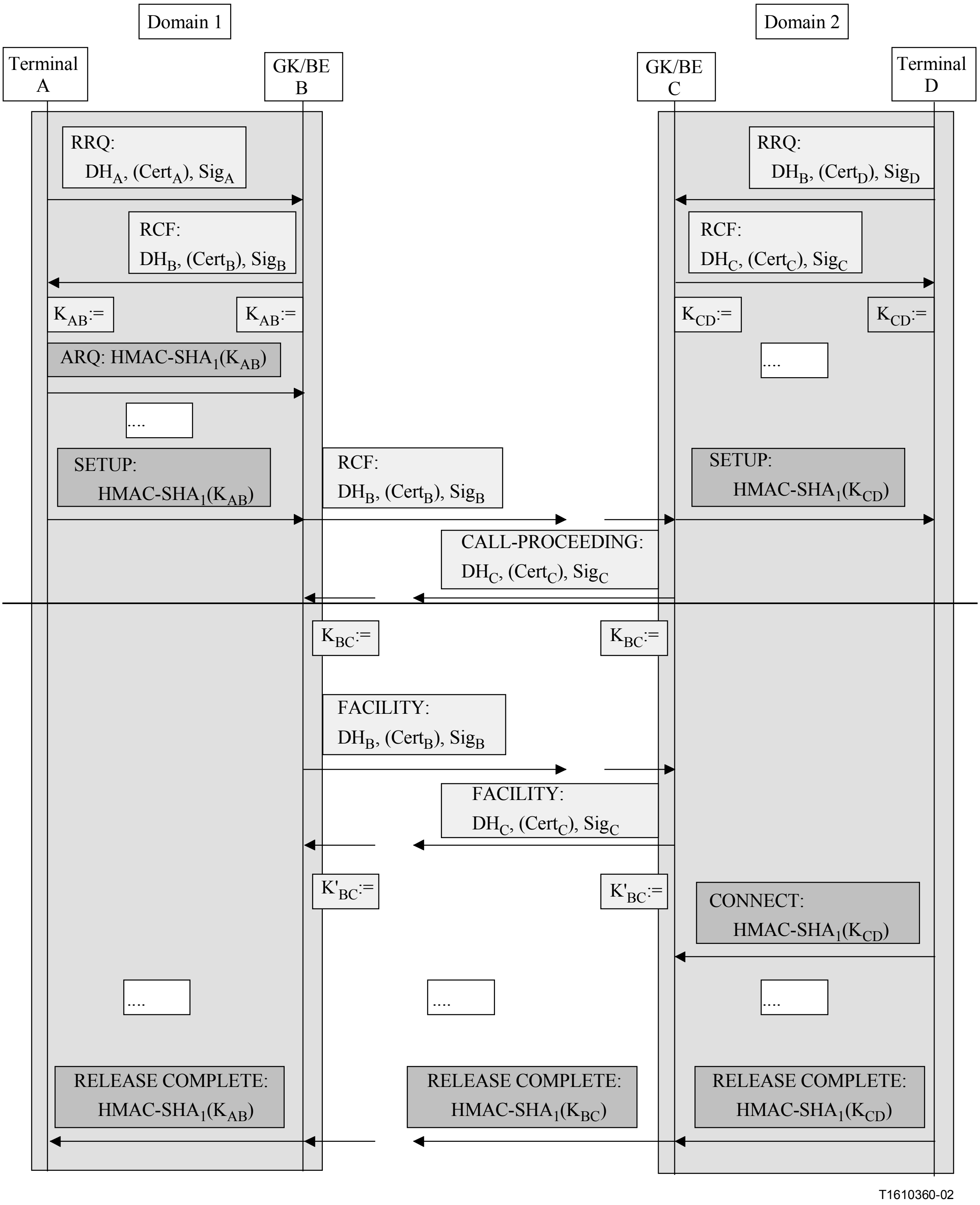
Cert	User certificate	K, K'	symmetric link key
DH_A	Diffie-Hellman Token g^a mod p	Sig	digital signature
DH_B	Diffie-Hellman Token g^b mod p		
EP	Endpoint (Terminal)		
GK	Gatekeeper		


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.235COR.1_FF2

Cert	User certificate	K, K'	Symmentric 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 F.2/H.235 – Flow diagram in a single administrative domain

…


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})
T1610360-02

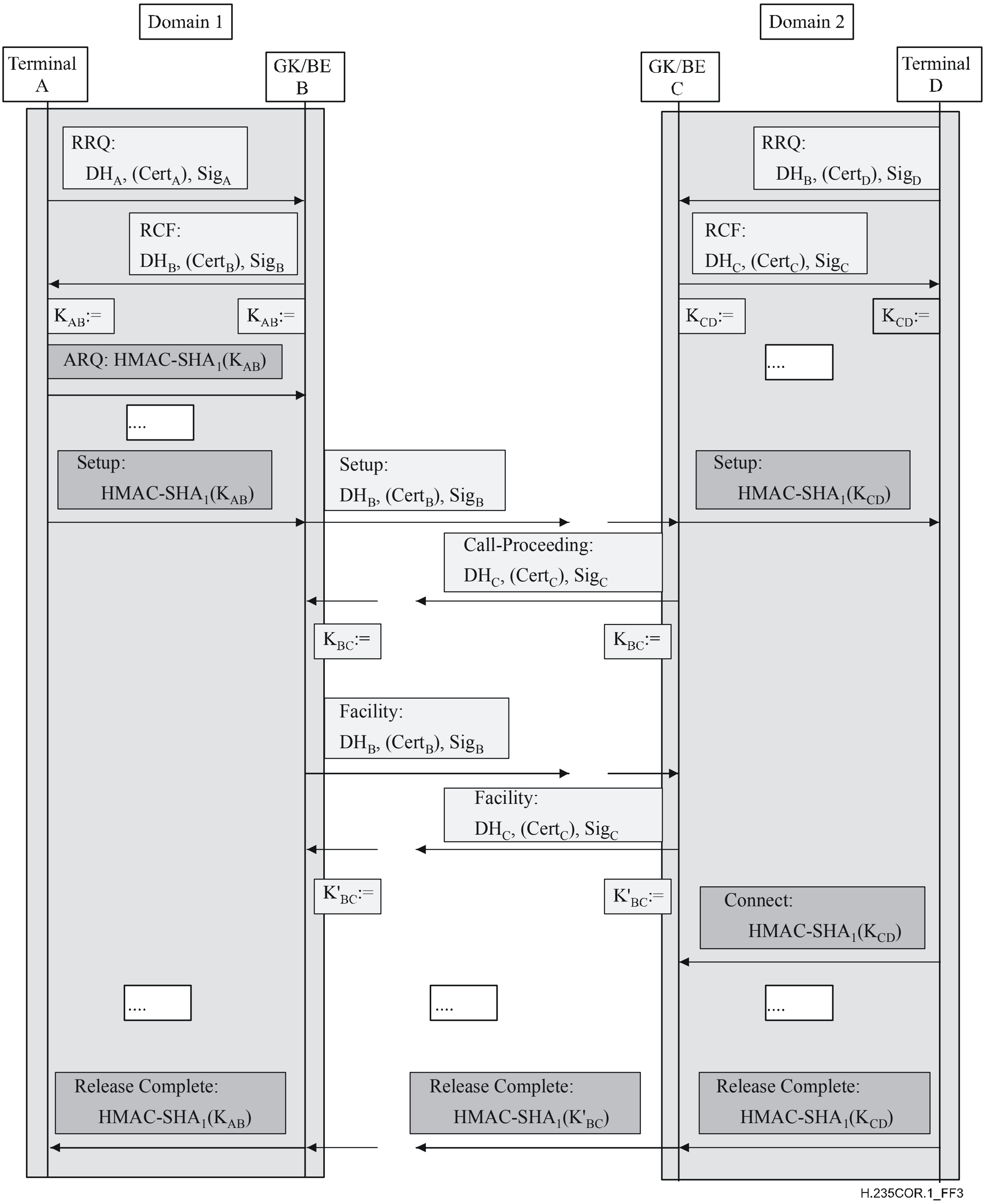

Domain 1
Domain 2
Terminal A
GK/BE B
GK/BE C
Terminal D
RRQ: DH$_A$, (Cert$_A$), Sig$_A$
RRQ: DH$_B$, (Cert$_D$), Sig$_D$
RCF: DH$_B$, (Cert$_B$), Sig$_B$
RCF: DH$_C$, (Cert$_C$), Sig$_C$
K$_{AB}$:=
K$_{AB}$:=
K$_{CD}$:=
K$_{CD}$:=
ARQ: HMAC-SHA$_1$(K$_{AB}$)
....
....
Setup: HMAC-SHA$_1$(K$_{AB}$)
Setup: 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.235COR.1_FF3

Figure F.3/H.235 – Flow diagram in a multi-administrative domain

…

Annex I

Support of direct-routed calls

…

I.5 Symbols and abbreviations

This annex uses the following abbreviations:

$ENC_{K;\,S,\,IV}(M)$	EOFB Encryption of *M* using secret key *K* and secret salting key *S* and initial vector *IV*
CT	ClearToken
DRC	Direct-Routed Call
EPID	Endpoint Identifier
GKID	Gatekeeper Identifier
K_{AG}	Shared secret (Annex D, Annex F) between EP A and GK G
K_{BH}	Shared secret (Annex D, Annex F) between EP B and GK H
K_{GH}	Shared, secret (Annex D, Annex F) between GK G and GK H
KS_{AG}	Secret, shared salting key between EP A and GK G
KS_{BH}	Secret, shared salting key between EP B and GK H
KS_{GH}	Secret, shared salting key between GK G and GK H
EK_{AG}	The encryption key shared between EP A and GK G
EK_{BH}	The encryption key shared between EP B and GK H
EK_{GH}	The encryption key shared between GK G and GK H
K_{AB}	The encryption key shared between EP A and EP B
PRF	Pseudo-Random Function

…

I.9 Procedure DRC

Endpoints capable of supporting this security profile shall indicate this fact during **GRQ** and/or **RRQ** by including a separate ClearToken with **tokenOID** set to "I0"; any other fields in that ClearToken should not be used. The Annex I-capable gatekeeper that is willing to provide this functionality shall reply with **GCF** or~~resp.~~ **RCF** with a separate ClearToken included with **tokenOID** set to "I0" and all other fields in the ClearToken unused.

Before an endpoint A starts sending call signalling messages to another endpoint B directly, the endpoint A or B shall apply for admission at the gatekeeper G or H using **ARQ**. Endpoint A shall include within **ARQ** a separate ClearToken with **tokenOID** set to "I0" and all other fields in the ClearToken unused.

This procedure covers the case of both a single, common gatekeeper to the endpoints and the case of multiple, chained gatekeepers. In case of multiple involved gatekeepers, gatekeeper G – in which zone the call originates – should locate gatekeeper H using the (multicast) **LRQ** mechanism as described in 8.1.6/H.323, "Optional called endpoint signalling". The communication between two gatekeepers shall be secured according to Annex D. For this, it is assumed that a common shared secret K_{GH} is available. Since **LRQ** among gatekeepers is typically a multicast message, the shared secret K_{GH} typically cannot be a pair-wise shared secret but is assumed to be actually a group-based shared secret within the potential cloud of gatekeepers.

NOTE – This assumption limits scalability in the general case, and does not allow source authentication. However, it is believed that in corporate networks with a limited, small number of well-known gatekeepers such constraint and security limitations are still acceptable. Securing intergatekeeper multicast communication using digital signatures could overcome those limitations; yet this is left as for further study.

If the **LRQ** mechanism is used to locate the far-end gatekeeper, then **LRQ** shall convey a separate ClearToken with **tokenOID** set to "I0"; any other fields in that ClearToken should not be used. For the multicast case, the **generalID** in the Clear~~rypto~~Token of **LRQ** shall not be used. Intergatekeeper communication using H.501 and/or H.510 are left as for further study.

EK_{BH} denotes the encryption key and KS_{BH} denotes the salting key that are~~is~~ shared between endpoint B and gatekeeper H. As is described below, both Gatekeeper H and endpoint B separately compute this keying material from the shared secret K_{BH} using a PRF.

Gatekeeper H shall generate a random Challenge-B, encryption key material EK_{BH} and salting key material KS_{BH} from the shared secret K_{BH} using the PRF-**based** key derivation procedure as defined in I.10 where Challenge-B is substituted as **challenge** and CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→keyDerivationOID** ~~**in V3KeySyncMaterial**~~ shall hold "AnnexI-HMAC-SHA1-PRF"; see I.12.

EK_{GH} denotes the encryption key and KS_{GH} denotes the salting key that are shared between gatekeeper G and gatekeeper H. Gatekeeper H shall generate one random Challenge-G. Gatekeeper H shall generate encryption key material EK_{GH} and salting key material KS_{GH} from the shared secret K_{GH} using the PRF-based key derivation procedure as defined in clause 11 where Challenge-G is substituted for **challenge**. CT_{HG}**→challenge** shall hold challenge-G, the endpoint ID of the endpoint B shall be set in CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→generalID**.

Gatekeeper H shall transmit the encrypted EK_{BH} to gatekeeper G. The enhanced OFB (EOFB) encryption mode (see B.2.5) shall be used with the secret, endpoint-specific salting key KS_{GH}. Applicable encryption algorithms are (see D.11):

- DES (56 bits) in EOFB mode using OID "Y1": optional;
- 3DES (168 bits) in outer-EOFB mode using OID "Z1": optional;
- AES (128 bits) in EOFB mode using OID "Z2": default and recommended;
- RC2-compatible (56 bits) in EOFB mode using OID "X1": optional.

For the EOFB encryption mode, gatekeeper~~GK~~ H shall generate a random initial value IV. For OID "X1", OID "Y1" and OID "Z1" the IV has 64 bits and shall be conveyed within CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv8**~~ of **params** within **V3KeySyncMaterial**~~; whereas the IV has 128 bits for OID "Z2" and shall be conveyed within CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv16**~~ of **params** within **V3KeySyncMaterial**~~.

Gatekeeper H shall include $ENC_{EK_{GH}, KS_{GH}, IV}(EK_{BH})$ in ClearToken CT_{HG} with **tokenOID** set to "I3". The obtained ciphertext $ENC_{EK_{GH}, KS_{GH}, IV}(EK_{BH})$ shall be conveyed in CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→encryptedSessionKey**~~the **h235key** data structure as part of **secureSharedSecret** where it shall be placed within the **encryptedSessionKey** of the **secureSharedSecret** data structure~~. The encryption algorithm shall be indicated in CT_{HG}**→h235Key→V3KeySyncMaterial→algorithmOID** ("X1", "Y1", "Z1" or "Z2") ~~within **V3KeySyncMaterial**~~. Challenge-B shall be placed within CT_{HG}**→h235Key→V3KeySyncMaterial→secureSharedSecret→clearSaltingKey**. CT_{HG}**→generalID** shall be set to the gatekeeper identifier G wheras CT_{HG}**→sendersID** shall be set to the gatekeeper identifier H. The **LCF** response shall hold the ClearToken CT_{HG}.

The gatekeeper G, recognizing that endpoints A and B support this annex, shall generate key material and ClearTokens as specified below.

The gatekeeper is able to calculate a call-based shared secret K_{AB}, besides the normal **ARQ** operation. This call-based shared secret is then propagated to both endpoints using ClearTokens. Those ClearTokens are conveyed within the **ACF** message and are sent back to the caller.

Two ClearTokens shall be included, one CT_A for the caller A and another one CT_B for the callee B. Each **ClearToken** shall contain an OID ("I1" or "I2") within **tokenOID** that indicates whether the token is destined for the caller (OID "I1" for CT_A) or for the callee (OID "I2" for CT_B).

The **ClearToken** as defined in this annex may be used in conjunction with other security profiles such as with Annex D or with Annex F that deploy **ClearTokens** as well. In such a case, Annex I ClearToken shall use those other **ClearToken** fields too. For example, in order to use Annex I in conjunction with Annex D, the fields **timeStamp**, **random**, **generalID**, **sendersID**, and **dhkey** shall be present and shall be used, as described by the Annex D security profiles.

The gatekeeper ID (GKID) of gatekeeper G shall be placed within CT_A→**sendersID** and within CT_B→**sendersID** whereas CT_A→**generalID** shall hold ~~either~~ the endpoint identifier of endpoint A and CT_B→**generalID** the endpoint identifier ~~(CT_A) or~~ of endpoint B ~~(CT_B)~~.

~~EK denotes the encryption key that is shared between an endpoint and its GK.~~ Gatekeeper G shall generate salting key material KS_{HG} and encryption key material EK_{HG} from K_{HG} using using the PRF-based key derivation procedure as defined in clause 11 with **challenge** substituted by CT_{HG}→**challenge**.

The encryption keys EK_{AG} and EK_{BH} for the encrypted end-to-end key K_{AB} shall be derived from the shared secret between the gatekeeper and the endpoints (EK_{AG} or EK_{BH}) using the PRF-**based** key derivation procedure as defined in I.10 where both CT_A→**h235Key→V3KeySyncMaterial→secureSharedSecret→keyDerivationOID** and CT_B→**h235Key→V3KeySyncMaterial→secureSharedSecret→keyDerivationOID**~~in **V3KeySyncMaterial**~~ shall hold "Annex I-HMAC-SHA1-PRF", see I.12 and CT_A→**challenge** shall hold Challenge-A.

~~The gatekeeper G shall generate a common shared session secret K_{AB}, which is shared between endpoint A and endpoint B.~~Gatekeeper G shall copy Challenge-B from CT_{HG}→**h235Key→V3KeySyncMaterial→secureSharedSecret→clearSaltingKey** into CT_B→**challenge**.

This session secret K_{AB} shall be encrypted by EK_{AG} (for CT destined to endpoint A) or by EK_{BH} (for the CT destined to endpoint B) using an encryption algorithm.

The enhanced OFB (EOFB) encryption mode (see B.2.5) shall be used with the secret, endpoint-specific salting key KS_{AG} or~~resp.~~ KS_{BH}~~G~~. Applicable encryption algorithms are (see clause D.11):

- DES (56 bits) in EOFB mode using OID "Y1": optional;
- 3DES (168 bits) in outer-EOFB mode using OID "Z1": optional;
- AES (128 bits) in EOFB mode using OID "Z2": default and recommended;
- RC2-compatible (56 bits) in EOFB mode using OID "X1": optional.

For the EOFB encryption mode, the gatekeeper G~~GK~~ shall generate a random initial value IV. For OID "X1", OID "Y1" and OID "Z1" the IV has 64 bits and shall be conveyed within CT_A→**h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv8** and ~~of **paramS** within **V3KeySyncMaterial**~~ CT_B→**h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv8**; whereas the IV has 128 bits for OID "Z2" and shall be conveyed within CT_A→**h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv16** ~~of **params**~~ and within CT_B**→h235Key→V3KeySyncMaterial→secureSharedSecret→params→iv16** ~~**V3KeySyncMaterial**~~.

The obtained ciphertext $\mathrm{ENC}_{EK_{AG}, KS_{AG}, IV}(K_{AB})$ shall be conveyed in CT_A**→h235Key→V3KeySyncMaterial→secureSharedSecret→encryptedSessionKey** and~~resp.~~ $\mathrm{ENC}_{EK_{BH\cancel{G}}, KS_{BH\cancel{G}}, IV}(K_{AB})$ shall then be conveyed in ~~the **h235key** data structure as part of **secureSharedSecret** where it shall be placed within the~~ CT_B**→h235Key→V3KeySyncMaterial→secureSharedSecret→encryptedSessionKey**~~ of the **secureSharedSecret** data structure~~. The encryption algorithm shall be indicated in CT_A**→h235Key→V3KeySyncMaterial→secureSharedSecret→algorithmOID** and in CT_B**→h235Key→V3KeySyncMaterial→secureSharedSecret→algorithmOID** ("X1", "Y1", "Z1" or "Z2")~~ within **V3KeySyncMaterial**~~.

For the ClearToken destined to endpoint A, the endpoint identifier of endpoint B ($EPID_B$) shall be placed within CT_A**→h235Key→V3KeySyncMaterial→secureSharedSecret→generalID**~~ of **V3KeySyncMaterial**~~. Likewise for the ClearToken destined to endpoint B, the endpoint identifier of endpoint A ($EPID_A$) shall be placed within CT_B**→h235Key→V3KeySyncMaterial→secureSharedSecret→generalID**~~ of **V3KeySyncMaterial**~~.

For the EOFB encryption algorithms, **encryptedSaltingKey** shall not be used.

The gatekeeper G shall include both ClearTokens CT_A and CT_B in the **ACF** towards endpoint A.

Endpoint A shall identify CT_A by inspection of the **tokenOID** "I1" within ClearToken.

Endpoint A shall verify that the obtained CT_A is fresh by checking the **timestamp**. Further security checks shall verify the **generalID** and **sendersID** of the ClearToken and **generalID** within **V3KeySyncMaterial**. If the received CT_A was verified as being fresh, endpoint A shall retrieve the IV and compute EK_{AG} and KS_{AG} as described above for the gatekeeper G. Endpoint A shall decrypt the **encryptedSessionKey** information found within **V3KeySyncMaterial** of CT_A to obtain ~~E~~K_{AB}.

If the received CT_A was verified as being fresh, endpoint A is able to send a SETUP message to endpoint B. This SETUP message includes CT_B. The SETUP message shall be secured (authenticated and/or integrity protected) according to Annex D or according to Annex F using K_{AB} as the applied shared secret. For this, **generalID** in the Annex D hashed ClearToken (not CT_B!) shall not be used unless endpoint A has already an $EPID_B$ available (e.g., through configuration or memorized from former communication). If endpoint A uses an $EPID_B$ value for **generalID** in SETUP then endpoint A shall accept the value of the **sendersID** in the returned call signalling message as the true $EPID_B$.

Endpoint B shall identify CT_B by inspection of the **tokenOID** "I2" within ClearToken.

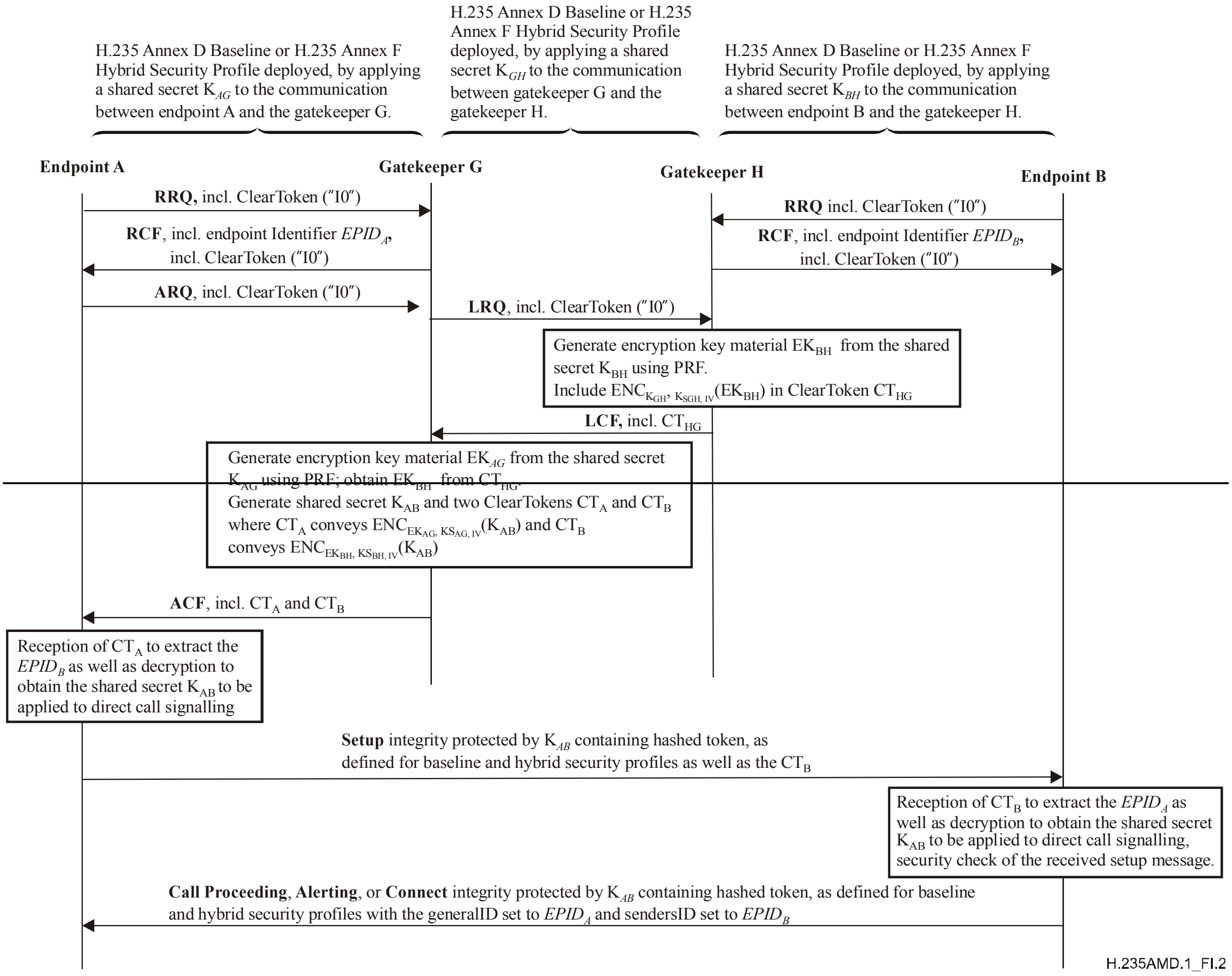
Endpoint B shall verify that the obtained CT_B is fresh by checking the **timestamp**. Further security checks shall verify the **sendersID** of the ClearToken and **generalID** within **V3KeySyncMaterial**. If the received CT_B was verified as being fresh, endpoint B shall retrieve the IV and compute EK_{BHG} and $KS_{BH\cancel{G}}$ as described above for the gatekeeper. Endpoint B shall decrypt the **encryptedSessionKey** information found within **V3KeySyncMaterial** of CT_B to obtain ~~E~~K_{AB}.

In case CT_B was verified as being fresh, endpoint B is able to proceed the call signalling by replying with CALL-PROCEEDING, ALERTING or CONNECT etc., as appropriate. In case CT_B was found not to be fresh or the security verification of the SETUP message failed, endpoint B shall reply with RELEASE-COMPLETE and the **ReleaseCompleteReason** set to a security error as defined by B.2.2.

When media security is to be deployed (see D.7), endpoint A and endpoint B shall exchange Diffie-Hellman half-keys according to D.7.1 and establish a dynamic session-based master key from which media-specific session keys can then be derived.

Endpoint B shall include generalID set to EPID_A and sendersID set to EPID_B for protection of any H.225.0 Call signalling message destined to EP A (e.g., Call Proceeding, Alerting or Connect).

Figure I.2 shows the basic communication flow:


H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{AG} to the communication between endpoint A and the gatekeeper G.
H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{GH} to the communication between gatekeeper G and the gatekeeper H.
H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{BH} to the communication between endpoint B and the gatekeeper H.
Endpoint A
Gatekeeper G
Gatekeeper H
Endpoint B
RRQ, incl. ClearToken ("I0")
RRQ incl. ClearToken ("I0")
RCF, incl. endpoint Identifier $EPID_A$, incl. ClearToken ("I0")
RCF, incl. endpoint Identifier $EPID_B$, incl. ClearToken ("I0")
ARQ, incl. ClearToken ("I0")
LRQ, incl. ClearToken ("I0")
Generate encryption key material EK_{BH} from the shared secret K_{BH} using PRF.
Include $ENC_{K_{GH}, K_{SGH, IV}}(EK_{BH})$ in ClearToken CT_{HG}
LCF, incl. CT_{HG}
Generate encryption key material EK_{AG} from the shared secret K_{AG} using PRF; obtain EK_{BH} from CT_{HG}.
Generate shared secret K_{AB} and two ClearTokens CT_A and CT_B where CT_A conveys $ENC_{EK_{AG}, KS_{AG, IV}}(K_{AB})$ and CT_B conveys $ENC_{EK_{BH}, KS_{BH, IV}}(K_{AB})$
ACF, incl. CT_A and CT_B
Reception of CT_A to extract the $EPID_B$ as well as decryption to obtain the shared secret K_{AB} to be applied to direct call signalling
Setup integrity protected by K_{AB} containing hashed token, as defined for baseline and hybrid security profiles as well as the CT_B
Reception of CT_B to extract the $EPID_A$ as well as decryption to obtain the shared secret K_{AB} to be applied to direct call signalling, security check of the received setup message.
Call Proceeding, **Alerting**, or **Connect** integrity protected by K_{AB} containing hashed token, as defined for baseline and hybrid security profiles with the generalID set to $EPID_A$ and sendersID set to $EPID_B$
H.235AMD.1_FI.2

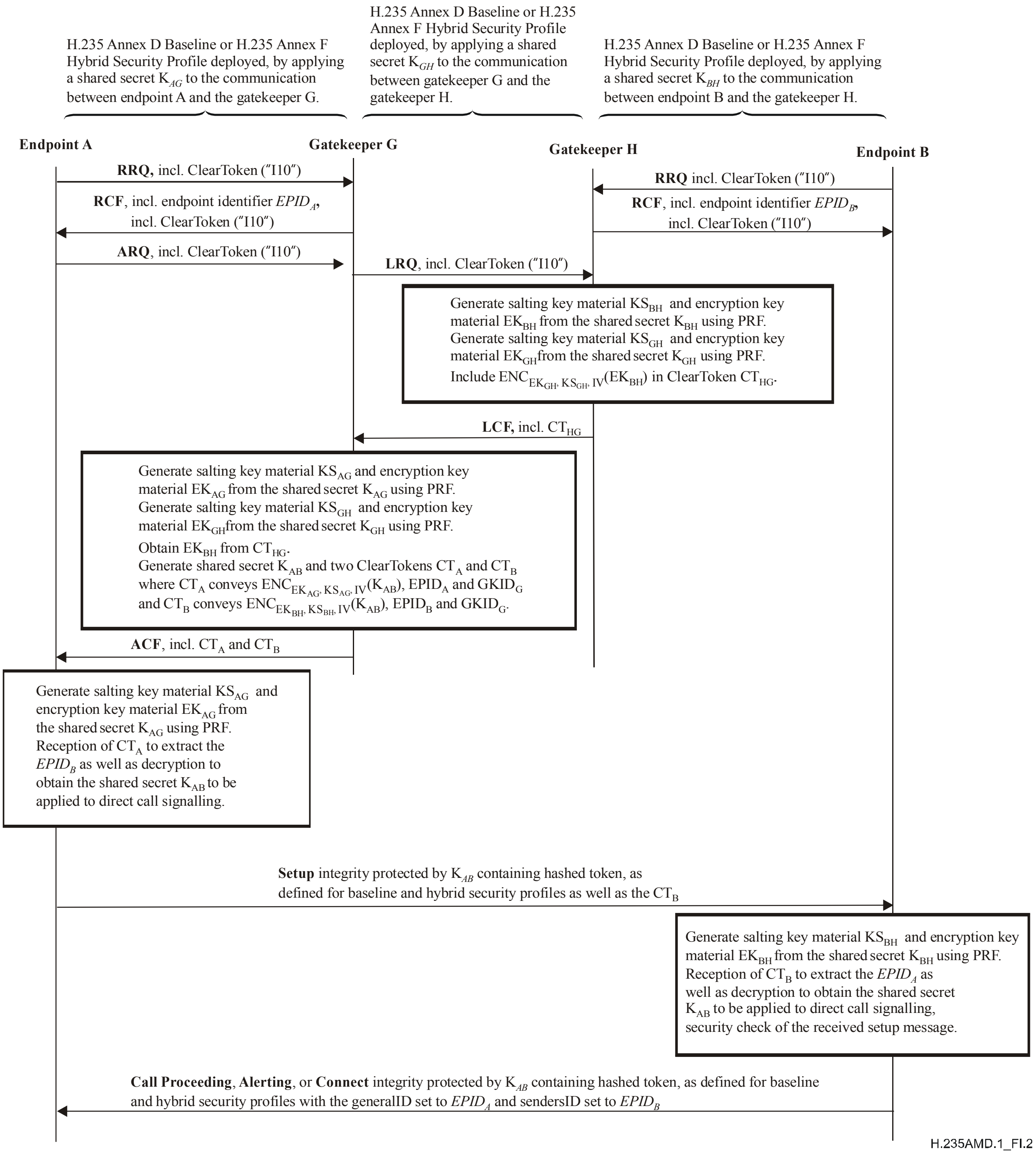

H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{AG} to the communication between endpoint A and the gatekeeper G.
H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{GH} to the communication between gatekeeper G and the gatekeeper H.
H.235 Annex D Baseline or H.235 Annex F Hybrid Security Profile deployed, by applying a shared secret K_{BH} to the communication between endpoint B and the gatekeeper H.
Endpoint A
Gatekeeper G
Gatekeeper H
Endpoint B
RRQ, incl. ClearToken ("I10")
RRQ incl. ClearToken ("I10")
RCF, incl. endpoint identifier $EPID_A$, incl. ClearToken ("I10")
RCF, incl. endpoint identifier $EPID_B$, incl. ClearToken ("I10")
ARQ, incl. ClearToken ("I10")
LRQ, incl. ClearToken ("I10")
Generate salting key material KS_{BH} and encryption key material EK_{BH} from the shared secret K_{BH} using PRF. Generate salting key material KS_{GH} and encryption key material EK_{GH} from the shared secret K_{GH} using PRF. Include $ENC_{EK_{GH}, KS_{GH}, IV}(EK_{BH})$ in ClearToken CT_{HG}.
LCF, incl. CT_{HG}
Generate salting key material KS_{AG} and encryption key material EK_{AG} from the shared secret K_{AG} using PRF. Generate salting key material KS_{GH} and encryption key material EK_{GH} from the shared secret K_{GH} using PRF. Obtain EK_{BH} from CT_{HG}. Generate shared secret K_{AB} and two ClearTokens CT_A and CT_B where CT_A conveys $ENC_{EK_{AG}, KS_{AG}, IV}(K_{AB})$, $EPID_A$ and $GKID_G$ and CT_B conveys $ENC_{EK_{BH}, KS_{BH}, IV}(K_{AB})$, $EPID_B$ and $GKID_G$.
ACF, incl. CT_A and CT_B
Generate salting key material KS_{AG} and encryption key material EK_{AG} from the shared secret K_{AG} using PRF. Reception of CT_A to extract the $EPID_B$ as well as decryption to obtain the shared secret K_{AB} to be applied to direct call signalling.
Setup integrity protected by K_{AB} containing hashed token, as defined for baseline and hybrid security profiles as well as the CT_B
Generate salting key material KS_{BH} and encryption key material EK_{BH} from the shared secret K_{BH} using PRF. Reception of CT_B to extract the $EPID_A$ as well as decryption to obtain the shared secret K_{AB} to be applied to direct call signalling, security check of the received setup message.
Call Proceeding, Alerting, or Connect integrity protected by K_{AB} containing hashed token, as defined for baseline and hybrid security profiles with the generalID set to $EPID_A$ and sendersID set to $EPID_B$
H.235AMD.1_FI.2

Figure I.2/H.235 – Basic communication flow

I.10 PRF-based key derivation procedure

This clause describes a procedure that defines how to derive key material from the shared secret and other parameters.

~~The encryption key EK_{AG} shall be computed using the PRF (see clause B.7) with the *inkey* parameter set to K_{AG} and *label* shall be set to the constant 0x2AD01C64 || **challenge**.~~

~~Likewise, the encryption key EK_{BG} shall be computed using that PRF with the *inkey* parameter set to K_{BH} and *label* shall be set to the constant 0x1B5C7973 || **challenge**. In both cases, *outkey_len*~~

~~shall be set to the length of the required length of the encryption key for the chosen encryption algorithm.~~

~~Using that same PRF, a secret, shared salting key shall be generated by the gatekeeper and by each endpoint. The salting key, when being used in conjunction with the EOFB encryption mode, guards against known-plaintext attacks of the CT_B by EP A where EP A might otherwise attempt to discover K_{BH}.~~

~~KS_{AG} denotes the secret, shared salting key that is shared between EP A and the GK G. KS_{AG} shall be computed using the PRF with the *inkey* parameter set to K_{AG} and *label* shall be set to the constant 0x150533E1 || **challenge**. KS_{BH} shall be computed using PRF with the *inkey* parameter set to K_{BH} and *label* shall be set to the constant 0x39A2C14B || **challenge**.~~ The procedure in this clause allows computing an encryption key and a salting key from a shared key. The procedure is uniform irrespective of the shared secret (K_{AG}, K_{BH} or K_{GH}).

In order to obtain the target keying material (e.g., EK_{AG}), the PRF (see B.7) shall be used with the parameters taken from Table I.0 where with the *inkey* parameter set to the corresponding shared key (e.g., K_{AG}), and *label* shall be set to the corresponding constant (e.g., 0x2AD01C64 || **challenge-A**) where || denotes concatenation. The *outkey_len* shall be set to the length of the required length of the target key material which depends on the chosen encryption algorithm.

~~NOTE – The 32-bit constant integers (i.e., 0x2AD01C64 etc.) are taken from the decimal digits of *e* (i.e., 2.7182...), and where each constant consists of nine decimal digits (e.g., the first nine decimal digits 718281828 = 0x2AD01C64). The strings of nine decimal digits are not chosen at random, but as consecutive "chuncks" from the decimal digits of *e*.~~

NOTE – For each EK_{AG}, KS_{AG}, EK_{BH}, and KS_{BH}, the 32-bit constant integers (i.e., 0x2AD01C64 etc.) are taken from the decimal digits of *e* (i.e., 2.7182...), and for EK_{GH} and KS_{GH}, the 32-bit constants integers are taken from the decimal digits of π (i.e., 3.1414...). For EK_{AG}, EK_{BH}, and KS_{BH}, the 32-bit integers are from blocks of 9 decimal digits, respectively the first, second, fourth and seventh blocks. The value for EK_{GH} comes from the first 10 decimal digits of π, while KS_{GH} comes from the subsequent 8 decimal digits of n.

Table I.0/H.235 – Calculating encryption and salting keys from a shared secret

Target key	PRF inkey	Constant || challenge
EK_{AG}	K_{AG}	`0x2AD01C64` || **Challenge-A**
KS_{AG}	K_{AG}	`0x150533E1` || **Challenge-A**
EK_{BH}	K_{BH}	`0x1B5C7973` || **Challenge-B**
KS_{BH}	K_{BH}	`0x39A2C14B` || **Challenge-B**
EK_{GH}	K_{GH}	`0x54655307` || **Challenge-G**
KS_{GH}	K_{GH}	`0x35855C60` || **Challenge-G**

I.11 FIPS-140-based key derivation procedure

This clause may describe a procedure that defines how to derive key material from a shared secret and other parameters using a FIPS-140 compliant crypto module. This is left as for further study.

I.12 List of object identifiers

Table I.1/H.235 – Object identifiers used by H.235 Annex I

Object identifier reference	Object identifier value	Description
"I0"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 48}	Used in procedure DRC during GRQ/RRQ and GCF/RCF and ARQ to let the EP/GK indicate support of Annex I.
"I1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 49}	Used in procedure DRC for the ClearToken tokenOID indicating that the ClearToken $\underline{CT_A}$ holds an end-to-end key for the caller.
"I2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 50}	Used in procedure DRC for the ClearToken tokenOID indicating that the ClearToken $\underline{CT_B}$ holds an end-to-end key for the callee.
"I3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 52}	Used in procedure DRC for the inter-gatekeeper ClearToken tokenOID indicating that the ClearToken $\underline{CT_{HG}}$ holds an encryption key for the originating gatekeeper.
"Annex I-HMAC-SHA1-PRF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 51}	Used in procedure DRC for keyDerivationOID within V3KeySyncMaterial to indicate the applied key derivation method in I.10 using the HMAC-SHA1 pseudo-random function.

...

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, 2005