INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235

(02/98)

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

ITU-T Recommendation H.235

(Previously CCITT Recommendation)

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Characteristics of transmission channels used for other than telephone purposes	H.10–H.19
Use of telephone-type circuits for voice-frequency telegraphy	H.20–H.29
Telephone circuits or cables used for various types of telegraph transmission or simultaneous transmission	H.30–H.39
Telephone-type circuits used for facsimile telegraphy	H.40–H.49
Characteristics of data signals	H.50–H.99
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.399

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

ITU-T RECOMMENDATION H.235

SECURITY AND ENCRYPTION FOR H-SERIES
(H.323 AND OTHER H.245-BASED)
MULTIMEDIA TERMINALS

Summary

This Recommendation describes enhancements within the framework of the H.3xx-Series Recommendations to incorporate security services such as *Authentication* and *Privacy* (data encryption). The proposed scheme is applicable to both simple point-to-point and multipoint conferences for any terminals which utilize Recommendation H.245 as a control protocol.

For example, H.323 systems operate over packet-based networks which do not provide a guaranteed quality of service. For the same technical reasons that the base network does not provide QOS, the network does not provide a secure service. Secure real-time communication over insecure networks generally involves two major areas of concern – *authentication* and *privacy*.

This Recommendation describes the security infrastructure and specific privacy techniques to be employed by the H.3xx-Series of multimedia terminals. This Recommendation will cover areas of concern for interactive conferencing. These areas include, but are not strictly limited to, authentication and privacy of all real-time media streams that are exchanged in the conference. This Recommendation provides the protocol and algorithms needed between the H.323 entities.

This Recommendation utilizes the general facilities supported in Recommendation H.245 and as such, any standard which operates in conjunction with this control protocol may use this security framework. It is expected that, wherever possible, other H-Series terminals may interoperate and directly utilize the methods described in this Recommendation. This Recommendation will not initially provide for complete implementation in all areas, and will specifically highlight endpoint authentication and media privacy.

This Recommendation includes the ability to negotiate services and functionality in a generic manner, and to be selective concerning cryptographic techniques and capabilities utilized. The specific manner in which they are used relates to systems capabilities, application requirements and specific security policy constraints. This Recommendation supports varied cryptographic algorithms, with varied options appropriate for different purposes; e.g. key lengths. Certain cryptographic algorithms may be allocated to specific security services (e.g. one for fast media stream encryption and another for signalling encryption).

It should also be noted that some of the available cryptographic algorithms or mechanisms may be reserved for export or other national issues (e.g. with restricted key lengths). This Recommendation supports signalling of well-known algorithms in addition to signalling non-standardized or proprietary cryptographic algorithms. There are no specifically mandated algorithms; however, it is strongly suggested that endpoints support as many of the applicable algorithms as possible in order to achieve interoperability. This parallels the concept that the support of Recommendation H.245 does not guarantee the interoperability between two entities' codecs.

Source

ITU-T Recommendation H.235 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 6th February 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The 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 Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 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.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The 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, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Recommendation H.235

SECURITY AND ENCRYPTION FOR H-SERIES (H.323 AND OTHER H.245-BASED) MULTIMEDIA TERMINALS

(Geneva, 1998)

1 Scope

The primary purpose of this Recommendation is to provide for authentication, privacy, and integrity within the current H-Series protocol framework. The current text of this Recommendation (1998) provides details on implementation with Recommendation H.323. This framework is expected to operate in conjunction with other H-Series protocols that utilize Recommendation H.245 as their control protocol.

Additional goals in this Recommendation include:

1) Security architecture should be developed as an extensible and flexible framework for implementing a security system for H-Series terminals. This should be provided through flexible and independent services and the functionality that they supply. This includes the ability to negotiate and to be selective concerning cryptographic techniques utilized, and the manner in which they are used.

2) Provide security for all communications occurring as a result of H.3xx protocol usage. This includes aspects of connection establishment, call control, and media exchange between all entities. This requirement includes the use of confidential communication (privacy), and may exploit functions for peer authentication as well as protection of the user's environment from attacks.

3) This Recommendation should not preclude integration of other security functions in H.3xx entities which may protect them against attacks from the network.

4) This Recommendation should not limit the ability for any H.3xx-Series Recommendation to scale as appropriate. This may include both the number of secured users and the levels of security provided.

5) Where appropriate, all mechanisms and facilities should be provided independent of any underlying transport or topologies. Other means that are outside the scope of this Recommendation may be required to counter such threats.

6) Provisions are made for operation in a mixed environment (secured and unsecured entities).

7) This Recommendation should provide facilities for distributing session keys associated with the cryptography utilized. (This does not imply that public-key-based certificate management must be part of this Recommendation.)

The security architecture, described in this Recommendation, does not assume that the participants are familiar with each other. It does however assume that appropriate precautions have been taken to physically secure the H-Series endpoints. The principal security threat to communications, therefore, is assumed to be eavesdropping on the network or some other method of diverting media streams.

Recommendation H.323 (1996) provides the means to conduct an audio, video and data conference between two or more parties, but does not provide the mechanism to allow each participant to authenticate the identity of the other participants, nor provide the means to make the communications private (i.e. encrypt the streams).

Recommendations H.323, H.324 and H.310 make use of the logical channel signalling procedures of Recommendation H.245, in which the content of each logical channel is described when the channel is opened. Procedures are provided for expression of receiver and transmitter capabilities, transmissions are limited to what receivers can decode, and receivers may request a particular desired mode from transmitters. The security capabilities of each endpoint are communicated in the same manner as any other communication capability.

Some H-Series (H.323) terminals may be used in multipoint configurations. The security mechanism described in this Recommendation will allow for secure operation in these environments, including both centralized and decentralized MCU operation.

2 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; all 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.

– ITU-T Recommendation H.225.0 (1998), *Call signalling protocols and media stream packetization for packet-based multimedia communications systems.*

– ITU-T Recommendation H.245 (1998), *Control protocol for multimedia communication.*

– ITU-T Recommendation H.323 (1998), *Packet-based multimedia communications systems.*

– ITU-T Recommendation Q.931 (1993), *ISDN user-network interface layer 3 specification for basic call control.*

– ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, *Information technology – Open Systems Interconnection – The Directory: Authentication framework.*

– CCITT 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.*

– ISO/IEC 9798-2:1994, *Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms.*

– ISO/IEC 9798-3:1993, *Information technology – Security techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public key algorithm.*

– ISO/IEC 9798-4:1995, *Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function.*

– ATKINSON (R.): Security Architecture for the Internet Protocol, RFC 1825, *Internet Engineering Task Force*, 1995.

– KRAWCZYK (H.), BELLARE (M.), CANETTI (R.): HMAC: Keyed-Hashing for Message Authentication, RFC 2104, *Internet Engineering Task Force*, 1997.

3 Definitions

For the purposes of this Recommendation the definitions given in clause 3 of Recommendations H.323, H.225.0 and H.245 apply along with those in this clause. Some of the following terms are used as defined in CCITT Rec. X.800 | ISO 7498-2 and in Recommendations X.803, X.810 and X.811.

3.1 access control: The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner (X.800).

3.2 authentication: The provision of assurance of the claimed identity of an entity (X.811).

3.3 authorization: The granting of permission on the basis of authenticated identification.

3.4 attack: The activities undertaken to bypass or exploit deficiencies in a system's security mechanisms. By a direct attack on a system they exploit deficiencies in the underlying algorithms, principles, or properties of a security mechanism. Indirect attacks are performed when they bypass the mechanism, or when they make the system use the mechanism incorrectly.

3.5 certificate: A set of security-relevant data issued by a security authority or trusted third party, together with security information which is used to provide the integrity and data origin authentication services for the data (X.810). In this Recommendation the term refers to "public key" certificates which are values that represent an owners public key (and other optional information) as verified and signed by a trusted authority in an unforgeable format.

3.6 cipher: A cryptographic algorithm, a mathematical transform.

3.7 confidentiality: The property that prevents disclosure of information to unauthorized individuals, entities, or processes.

3.8 cryptographic algorithm: Mathematical function that computes a result from one or several input values.

3.9 encipherment: Encipherment (encryption) is the process of making data unreadable to unauthorized entities by applying a cryptographic algorithm (an encryption algorithm). Decipherment (decryption) is the reverse operation by which the ciphertext is transformed to the plaintext.

3.10 integrity: The property that data has not been altered in an unauthorized manner.

3.11 key management: The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy (X.800).

3.12 media stream: A media stream can be of type audio, video or data or a combination of any of them. Media stream data conveys user or application data (payload) but no control data.

3.13 nonrepudiation: Protection from denial by one of the entities involved in a communication of having participated in all or part of the communication.

3.14 privacy: A mode of communication in which only the explicitly enabled parties can interpret the communication. This is typically achieved by encryption and shared key(s) for the cipher.

3.15 private channel: For this Recommendation, a private channel is one that is a result of prior negotiation on a secure channel. In this context it may be used to handle media streams.

3.16 public key cryptography: An encryption system utilizing asymmetric keys (for encryption/decryption) in which the keys have a mathematical relationship to each other – which cannot be reasonably calculated.

3.17 symmetric (secret-key based) cryptographic algorithm: An algorithm for performing encipherment or the corresponding algorithm for performing decipherment in which the same key is required for both encipherment and decipherment (X.810).

3.18 threat: A potential violation of security (X.800).

4 Symbols and abbreviations

This Recommendation uses the following abbreviations:

DSS	Digital Signature Standard
IPSEC	Internet Protocol Security
QOS	Quality of Service
RSA	Rivest, Shamir and Adleman (public key algorithm)
SDU	Service Data Unit
TLS	Transport Level Security

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.

References to clauses, subclauses, annexes and appendices refer to those items within this Recommendation unless another Recommendation is explicitly listed. For example, "1.4" refers to subclause 1.4 of this Recommendation; "6.4/H.245" refers to subclause 6.4 in Recommendation H.245.

This Recommendation describes the use of "n" different message types: H.245, RAS, Q.931, etc. To distinguish between the different message types, the following convention is followed. H.245 message and parameter names consist of multiple concatenated words highlighted in bold typeface (**maximumDelayJitter**). RAS message names are represented by three-letter abbreviations (**ARQ**). Q.931 message names consist of one or two words with the first letters capitalized (**Call Proceeding**).

6 System introduction

6.1 Summary

1) The call signalling channel may be secured using TLS **[TLS]** or IPSEC **[13/IPSEC]** on a secure well-known port (H.225.0).

2) Users may be authenticated either during the initial call connection, in the process of securing the H.245 channel and/or by exchanging certificates on the H.245 channel.

3) The encryption capabilities of a media channel are determined by extensions to the existing capability negotiation mechanism.

4) Initial distribution of key material from the master is via H.245 **OpenLogicalChannel** or **OpenLogicalChannelAck** messages.

5) Re-keying may be accomplished by H.245 commands: **EncryptionUpdateRequest** and **EncryptionUpdate**.

6) Key material distribution is protected either by operating the H.245 channel as a private channel or by specifically protecting the key material using the selected exchanged certificates.

7) The security protocols presented either conform to ISO published standards or IETF proposed standards.

6.2 Authentication

The process of authentication verifies that the respondents are, in fact, who they say they are. Authentication may be accomplished in conjunction with the exchange of public key based certificates. Authentication may also be accomplished by an exchange which utilizes a shared secret between the entities involved. This may be a static password or some other *a priori* piece of information.

This Recommendation describes the protocol for exchanging the certificates, but does not specify the criteria by which they are mutually verified and accepted. In general, certificates give some assurance to the verifier that the presenter of the certificate is who he says he is. The intent behind the certificate exchange is to authenticate the *user* of the endpoint, not simply the physical endpoint. Using digital certificates, an authentication protocol proves that the respondents possess the private keys corresponding to the public keys contained in the certificates. This authentication protects against man-in-the-middle attacks, but does not automatically prove who the respondents are. To do this normally requires that there be some policy regarding the other contents of the certificates. For authorization certificates, for example, the certificate would normally contain the service-provider's identification along with some form of user account identification prescribed by the service provider.

The authentication framework in this Recommendation does not prescribe the contents of certificates (i.e. does not specify a certificate policy) beyond that required by the authentication protocol. However, an application using this framework may impose high-level policy requirements such as presenting the certificate to the user for approval. This higher level policy may either be automated within the application or require human interaction.

For authentication which does not utilize digital certificates, this Recommendation provides the signalling to complete various challenge/response scenarios. This method of authentication requires prior coordination by the communicating entities so that a shared secret may be obtained. An example of this method would be a customer of a subscription-based service.

As a third option, the authentication may be completed within the context of a separate security protocol such as TLS **[TLS]** or IPSEC **[13/IPSEC]**.

Both bidirectional and unidirectional authentication may be supported by peer entities. This authentication may occur on some or all of the communication channels.

All of the specific authentication mechanisms described in this Recommendation are identical to, or derived from, ISO developed algorithms as specified in Parts 2 to 3 of ISO/IEC 9798, or based on IETF protocols.

6.2.1 Certificates

The standardization of certificates, including their generation, administration and distribution is outside the scope of this Recommendation. The certificates used to establish secure channels (call signalling and/or call control) shall conform to those prescribed by whichever protocol has been negotiated to secure the channel.

It should be noted that for authentication utilizing public key certificates, the endpoints are required to provide digital signatures using the associated private key value. The exchange of public key certificates alone does not protect against man-in-the-middle attacks. The H.235 protocols conform to this requirement.

6.3 Call establishment security

There are at least two reasons to motivate securing the call establishment channel (e.g. H.323 using Q.931). The first is for simple authentication, before accepting the call. The second reason is to allow for call authorization. If this functionality is desired in the H-Series terminal, a secure mode of communication should be used (such as TLS/IPSEC for H.323) before the exchange of call connection messages. Alternatively, the authorization may be provided based upon a service-specific authentication. The constraints of a service-specific authorization policy are outside the scope of this Recommendation.

6.4 Call control (H.245) security

The call control channel (H.245) should also be secured in some manner to provide for subsequent media privacy. The H.245 channel shall be secured using any negotiated privacy mechanism (this includes the option of "none"). H.245 messages are utilized to signal encryption algorithms and encryption keys used in the shared, private, media channels. The ability to do this, on a logical channel by logical channel basis, allows different media channels to be encrypted by different mechanisms. For example, in centralized multipoint conferences, different keys may be used for streams to each endpoint. This may allow media streams to be made private for each endpoint in the conference. In order to utilize the H.245 messages in a secure manner, the entire H.245 channel (logical channel 0) should be opened in a negotiated secure manner.

The mechanism by which H.245 is made secure is dependent on the H-Series terminals involved. The only requirement on all systems that utilize this security structure is that each shall have some manner in which to negotiate and/or signal that the H.245 channel is to be operated in a particular secured manner before it is actually initiated. For example, H.323 will utilize the H.225.0 connection signalling messages to accomplish this.

6.5 Media stream privacy

This Recommendation describes media privacy for media streams carried on packet-based transports. These channels may be unidirectional with respect to H.245 logical channel characterizations. The channels are not required to be unidirectional on a physical or transport level.

A first step in attaining media privacy should be the provision of a private control channel on which to establish cryptographic keying material and/or set up the logical channels which will carry the encrypted media streams. For this purpose, when operating in a secure conference, any participating endpoints may utilize an encrypted H.245 channel. In this manner, cryptographic algorithm selection and encryption keys as passed in the H.245 **OpenLogicalChannel** command are protected.

The H.245 secure channel may be operated with characteristics different from those in the private media channel(s) as long as it provides a mutually acceptable level of privacy. This allows for the security mechanisms protecting media streams and any control channels to operate in a completely independent manner, providing completely different levels of strength and complexity.

If it is required that the H.245 channel be operated in a non-encrypted manner, the specific media encryption keys may be encrypted separately in the manner signalled and agreed to by the participating parties. A logical channel of type **h235Control** may be utilized to provide the material to protect the media encryption keys. This logical channel may be operated in any appropriately negotiated mode.

The privacy (encryption) of data carried in logical channels shall be in the form specified by the **OpenLogicalChannel**. Transport-specific header information shall not be encrypted. The privacy of data is to be based upon end-to-end encryption.

6.6 Trusted elements

The basis for authentication (trust) and privacy is defined by the terminals of the communications channel. For a connection establishment channel, this may be between the caller and a hosting network component. For example, a telephone "trusts" that the network switch will connect it with the telephone whose number has been dialled. For this reason, any entity which terminates an encrypted H.245 control channel or any **encryptedData** type logical channels shall be considered a trusted element of the connection; this may include MC(U)s and gateways. The result of trusting an element is the confidence to reveal the privacy mechanism (algorithm and key) to that element.

Given the above, it is incumbent upon participants in the communications path to authenticate any and all "trusted" elements. This will normally be done by certificate exchange as would occur for the "standard" end-to-end authentication. This Recommendation will not require any specific level of authentication, other than to suggest that it be acceptable to all entities using the trusted element. Details of a trust model and certificate policy are for further study.

Privacy can be assured between the two endpoints only if connections between trusted elements are proven to be protected against man-in-the-middle attacks.

6.6.1 Key escrow

Although not specifically required for operation, this Recommendation contains provision for entities utilizing the H.235 protocol to support a key recovery technique within the signalling elements.

The ability to recover lost media encryption keys should be supported in installations where this functionality is desired or required.

Key escrow is a facility which is often referred to as a Trusted Third Party (TTP). This facility is for further study.

6.7 Non-repudiation

FFS.

7 Connection establishment procedures

7.1 Introduction

As stated in the system introduction clause, both the call connection channel (H.225.0 for H.323 series) and call control (H.245) channel shall operate in the negotiated secured or unsecured mode starting with the first exchange. For the call connection channel, this is done *a priori* [for H.323, a TLS secured TSAP (port 1300) shall be utilized for the Q.931 messages]. For the call control channel, security mode is determined by information passed in the initial connection setup protocol in use by the H-Series terminal.

In the cases in which there are no overlapping security capabilities, the called terminal may refuse the connection. The error returned should convey no information about any security mismatch; the calling terminal will have to determine the problem by some other means. In cases where the calling terminal receives a CONNECT ACKNOWLEDGE message without sufficient security capabilities, it should terminate the call.

If the calling and called terminals have compatible security capabilities, it shall be assumed by both sides that the H.245 channel shall operate in the secure mode negotiated. Failure to set up the H.245 channel in the secure mode determined here should be considered a protocol error and the connection terminated.

8 H.245 signalling and procedures

In general, the privacy aspects of media channels are controlled in the same manner as any other encoding parameter; each terminal indicates its capabilities, the source of the data selects a format to use, and the receiver acknowledges or denies the mode. All transport-independent aspects of the mechanism such as algorithm selection are indicated in generic logical channel elements. Transport specifics such as key/encryption algorithm synchronization are passed in transport-specific structures.

8.1 Secure H.245 channel operation

Assuming that the connection procedures in the previous clause (Connection establishment procedures) indicate a secure mode of operation, the negotiated handshake and authentication shall occur for the H.245 logical channel before any other H.245 messages are exchanged. If negotiated, any exchange of certificates shall occur using any mechanism appropriate for the H-Series terminal(s). After completing the securing of the H.245 channel, the terminals use the H.245 protocol in the same manner that they would in an insecure mode.

8.2 Unsecured H.245 channel operation

Alternatively, the H.245 channel may operate in an unsecured manner and the two entities open a secure logical channel with which to perform authentication and/or shared-secret derivation. For example TLS or IPSEC may be utilized by opening a logical channel with the **dataType** containing a value for **encryptionData**. This channel could then be used to derive a shared secret which protects any media session keys or to transport the **EncryptionSync**.

8.3 Capability exchange

Following the procedures in 8.3/H.245 (Capability exchange procedures) and the appropriate H-Series system Recommendation, endpoints exchange capabilities using H.245 messages. These capability sets may now contain definitions which indicate security and encryption parameters. For example, an endpoint might provide capabilities to send and receive H.261 video. It may also signal the ability to send and receive encrypted H.261 video.

Each encryption algorithm that is utilized in conjunction with a particular media codec implies a new capability definition. As with any other capability, endpoints may supply both independent and dependent encrypted codecs in their exchange. This will allow endpoints to scale their security capabilities based upon overhead and resources available.

After capability exchange has been completed, endpoints may open secure logical channels for media in the same manner that they would in an insecure manner.

8.4 Master role

The H.245 master-slave is used to establish the master entity for the purpose of bidirectional channel operation and other conflict resolution. This role of master is also utilized in the security methods. Although the security mode(s) of a media stream is set by the source (in deference to the capabilities of the receiver), the master is the endpoint which generates the encryption key. This generation of the encryption key is done, regardless of whether the master is the receiver or the source of the encrypted media. In order to allow for multicast channel operation with shared keys, the MC (also the master) should generate the keys.

8.5 Logical channel signalling

Endpoints open secure media logical channels in the same manner that they open unsecured media logical channels. Each channel may operate in a completely independent manner from other channels – in particular where this pertains to security. The particular mode shall be defined in the **OpenLogicalChannel dataType** field. The initial encryption key shall be passed in either the **OpenLogicalChannel** or **OpenLogicalChannelAck** depending on the master/slave relationship of the originator of the **OpenLogicalChannel**.

The **OpenLogicalChannelAck** shall act as confirmation of the encryption mode. If the **openLogicalChannel** is unacceptable to the recipient, either **dataTypeNotSupported** or **dataTypeNotAvailable** (transient condition) shall be returned in the cause field of the **OpenLogicalChannelReject**.

During the protocol exchange that establishes the logical channel, the encryption key shall be passed from the master to the slave (regardless of who initiated the **OpenLogicalChannel**). For media channels opened by an endpoint (other than the master), the master shall return the initial encryption key and the initial synchronization point in the **OpenLogicalChannelAck** (in the **encryptionSync** field). For media channels opened by the master, the **OpenLogicalChannel** shall include the initial encryption key and the synchronization point in the **encryptionSync** field.

9 Multipoint procedures

9.1 Authentication

Authentication shall occur between an endpoint and the MC(U) in the same manner that it would in a point-to-point conference. The MC(U) shall set the policy concerning level and stringency of authentication. As stated in 6.6, the MC(U) is trusted; existing endpoints in a conference may be limited by the authentication level employed by the MC(U). New **ConferenceRequest**/ **ConferenceResponse** commands allow endpoints to obtain the certificates of other participants in the conference from the MC(U). As outlined in H.245 procedures, endpoints in a multipoint conference may request other endpoint certificates via the MC, but may not be able to perform direct cryptographic authentication within the H.245 channel.

9.2 Privacy

MC(U) shall win all master/slave exchanges and as such shall supply encryption key(s) to participants in a multipoint conference. Privacy for individual sources within a common session (assuming multicast) may be achieved with individual or common keys. These two modes may be arbitrarily chosen by the MC(U) and shall not be controllable from any particular endpoint except in modes allowed by MC(U) policy. In other words, a common key may be used across multiple logical channels as opened from different sources.

10 Authentication signalling and procedures

10.1 Introduction

There are two types of authentication that may be utilized. The first type is symmetric encryption-based that requires no prior contact between the communicating entities. The second type is based on the ability to have some prior shared secret (further referenced as "subscription" based). Two forms of subscription-based authentication are provided: password and certificate.

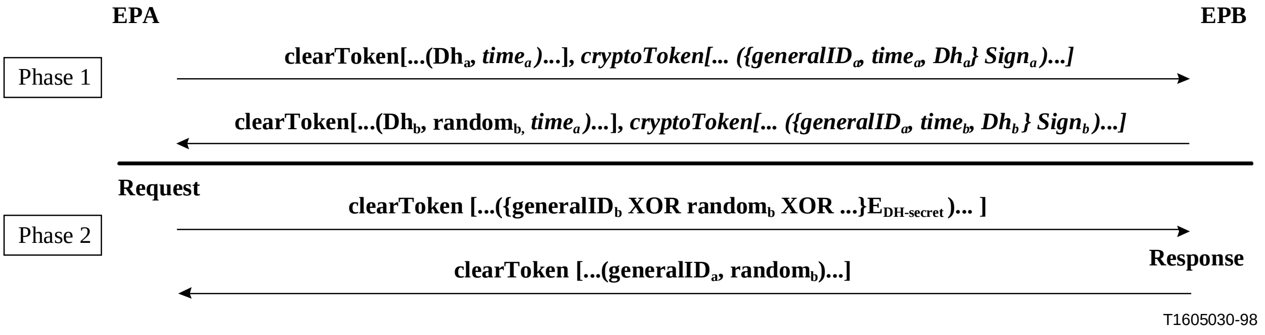
10.2 Diffie-Hellman with optional authentication

The intent is not to provide absolute, user-level authentication. This method provides signalling to generate a shared secret between two entities which may lead to keying material for private communications.

At the end of this exchange both the entities will possess a shared secret key along with a chosen algorithm with which to utilize this key. This shared secret key may now be used on any subsequent request/response exchanges. It should be noted that in rare cases the Diffie-Hellman exchange may generate known *weak* keys for particular algorithms. When this is the case, either entity should disconnect and reconnect to establish a new key set.

The first phase of Figure 1 demonstrates the data exchanged during the Diffie-Hellman. The second phase allows for application- or protocol-specific request messages to be authenticated by the responder. Note that a new random value may be returned with each response.

NOTE – An optional signature element may also be provided; these are illustrated in ***italics*** below.


EPA
EPB
Phase 1
clearToken[...(Dh_a, time_a)...], cryptoToken[... ({generalID_a, time_a, Dh_a} Sign_a)...]
clearToken[...(Dh_b, random_b, time_a)...], cryptoToken[... ({generalID_a, time_b, Dh_b} Sign_b)...]
Request
Phase 2
clearToken [...({generalID_b XOR random_b XOR ...}E_DH-secret)...]
Response
clearToken [...(generalID_a, random_b)...]
T1605030-98

[… …] indicates a sequence of tokens

() indicates a particular token, which may contain multiple elements

{}$E_{DH\text{-}Secret}$ indicates the contained values are encrypted utilizing the Diffie-Hellman secret

(EPB) knows which shared secret key to use to decipher the **generalID**$_b$ identifier by associating it with the **generalID**$_a$, which should also be passed in the message. Note that the encrypted value in phase 2 is passed in the **generalID** field of a **clearToken** to simplify encoding.

Figure 1/H.235

10.3 Subscription-based authentication

10.3.1 Introduction

Although the procedures outlined here (and the ISO algorithms from which they are derived) are bidirectional in nature, they may be utilized in only one direction if authentication is only needed in that direction. These exchanges assume that each end possesses some well-known identifier (such as a text identifier) which uniquely identifies it. A further assumption is made that there is a mutually acceptable reference to time (from which to derive timestamps). The amount of time skew that is acceptable is a local implementation matter.

There are three different variations that may be implemented depending on requirements:

1) password-based with symmetric encryption;

2) password-based with hashing;

3) certificate-based with signatures.

In all cases the token will contain the information as described in the following subclauses depending on the variation chosen. Note that, in all cases, the **generalID** may be known through configuration or directory lookup rather than in band protocol exchange.

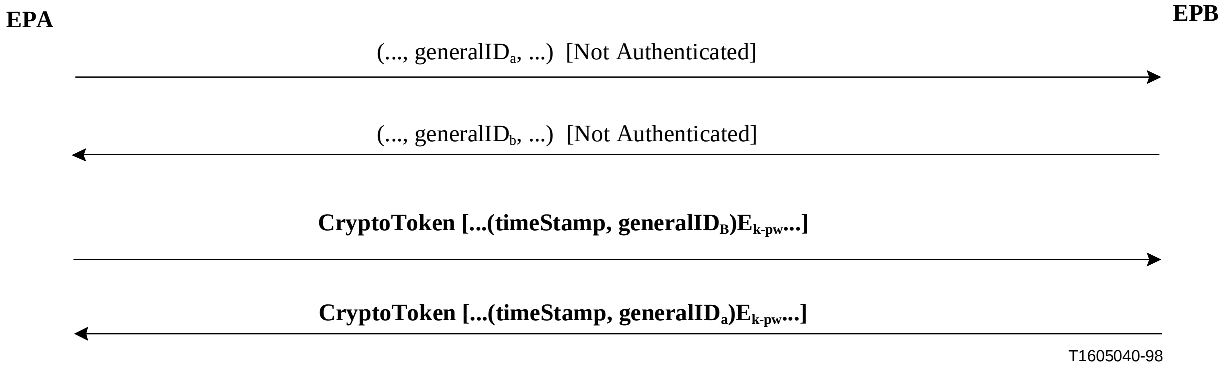
10.3.2 Password with symmetric encryption

Figure 2 shows the token format and the message exchange required to perform this type of authentication. This protocol is based on 5.2.1 of ISO/IEC 9798-2; it is assumed that an identifier and associated password are exchanged during subscription. The encryption key is length N octets (as indicated by the AlgorithmID), and is formed as follows:

– If password length = N, Key = password;

– if password length < N, the key is padded with zeros;

– if password length > N, the first N octets are assigned to the key, then the N + M*th* octet of the password is XOR'd to the Mmod(N)*th* octet (for all octets beyond N) (i.e. all "extra" password octets are repeatedly folded back on the key by XORing).


EPA
EPB
(..., generalID$_a$, ...) [Not Authenticated]
(..., generalID$_b$, ...) [Not Authenticated]
CryptoToken [...(timeStamp, generalID$_B$)E$_{k\text{-}pw}$...]
CryptoToken [...(timeStamp, generalID$_a$)E$_{k\text{-}pw}$...]
T1605040-98

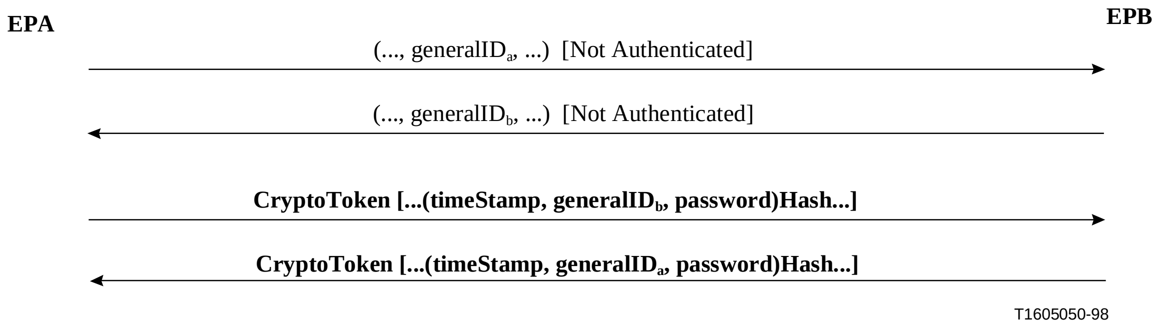
NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – **E$_{k\text{-}pw}$** indicates values that are encrypted using the key "k" derived from the password "pw".

Figure 2/H.235

10.3.3 Password with hashing

Figure 3 shows the token format and the message exchange required to perform this type of authentication. This protocol is based on 5.2.1 of ISO/IEC 9798-4; it is assumed that an identifier and associated password are exchanged during subscription.


EPA
EPB
(..., generalID$_a$, ...) [Not Authenticated]
(..., generalID$_b$, ...) [Not Authenticated]
CryptoToken [...(timeStamp, generalID$_b$, password)Hash...]
CryptoToken [...(timeStamp, generalID$_a$, password)Hash...]
T1605050-98

NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

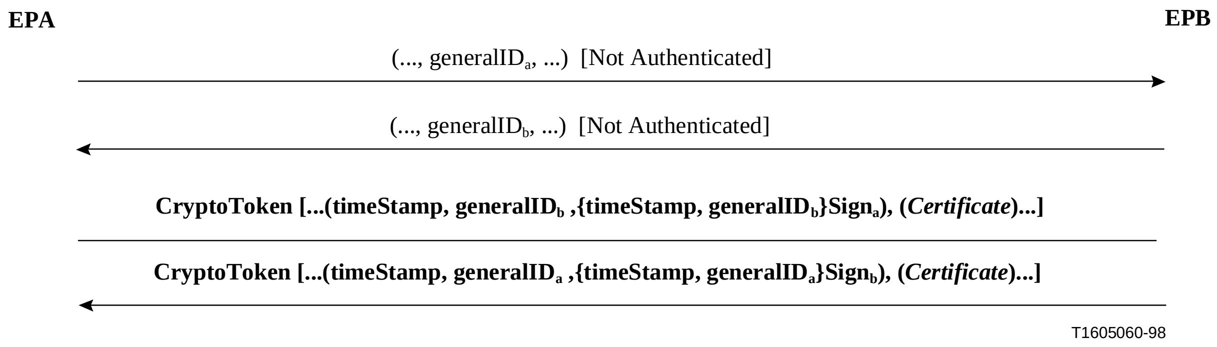
NOTE 2 – **Hash** indicates a hashing function that operates on the contained values.

Figure 3/H.235

10.3.4 Certificate-based with signatures

Figure 4 shows the token format and the message exchange required to perform this type of authentication. This protocol is based on 5.2.1 of ISO/IEC 9798-3; it is assumed that an identifier and associated certificate are assigned/exchanged during subscription.

NOTE – An optional certificate element may also be provided; these are illustrated in ***italics*** below.


EPA
EPB
(..., generalID$_a$, ...) [Not Authenticated]
(..., generalID$_b$, ...) [Not Authenticated]
CryptoToken [...(timeStamp, generalID$_b$,{timeStamp, generalID$_b$}Sign$_a$), (Certificate)...]
CryptoToken [...(timeStamp, generalID$_a$,{timeStamp, generalID$_a$}Sign$_b$), (Certificate)...]
T1605060-98

NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – A "payment" type certificate may be optionally included by the EPA originator.

NOTE 3 – **Sign** indicates a signing function (from associated certificate) performed on the contained values.

Figure 4/H.235

11 Media stream encryption procedures

Media streams shall be encoded using the algorithm and key as presented in the H.245 channel. Figures 5 and 6 show the general flow. Note that the transport header is attached to the transport SDU after the SDU has been encrypted. The opaque segments indicate privacy. As new keys are received by the transmitter and used in the encryption, the SDU header shall indicate in some manner to the receiver that the new key is now in use. For example, in H.323 the RTP header (SDU) will change its payload type to indicate the switch to the new key.

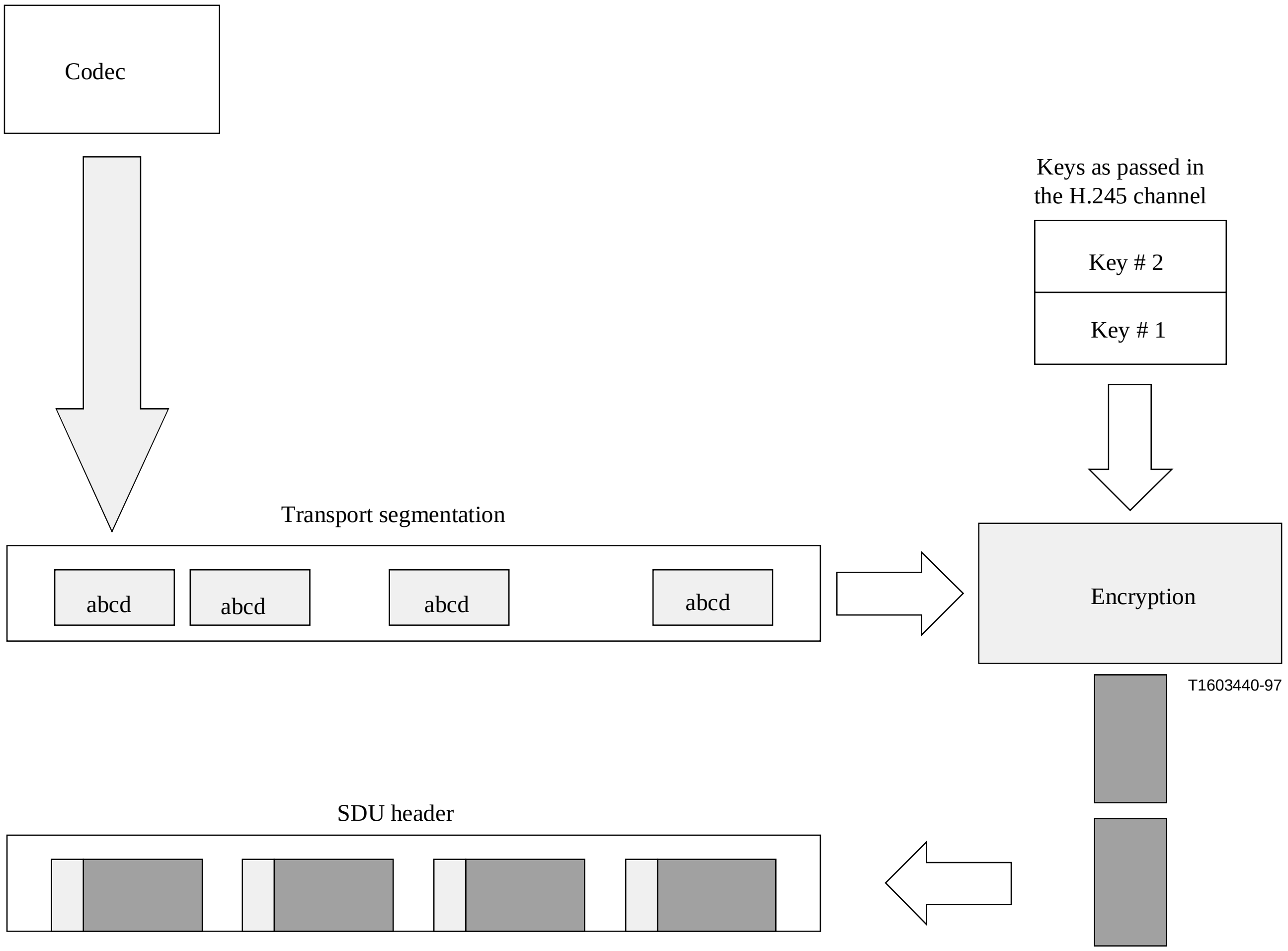


Figure 5/H.235 – Encryption of media

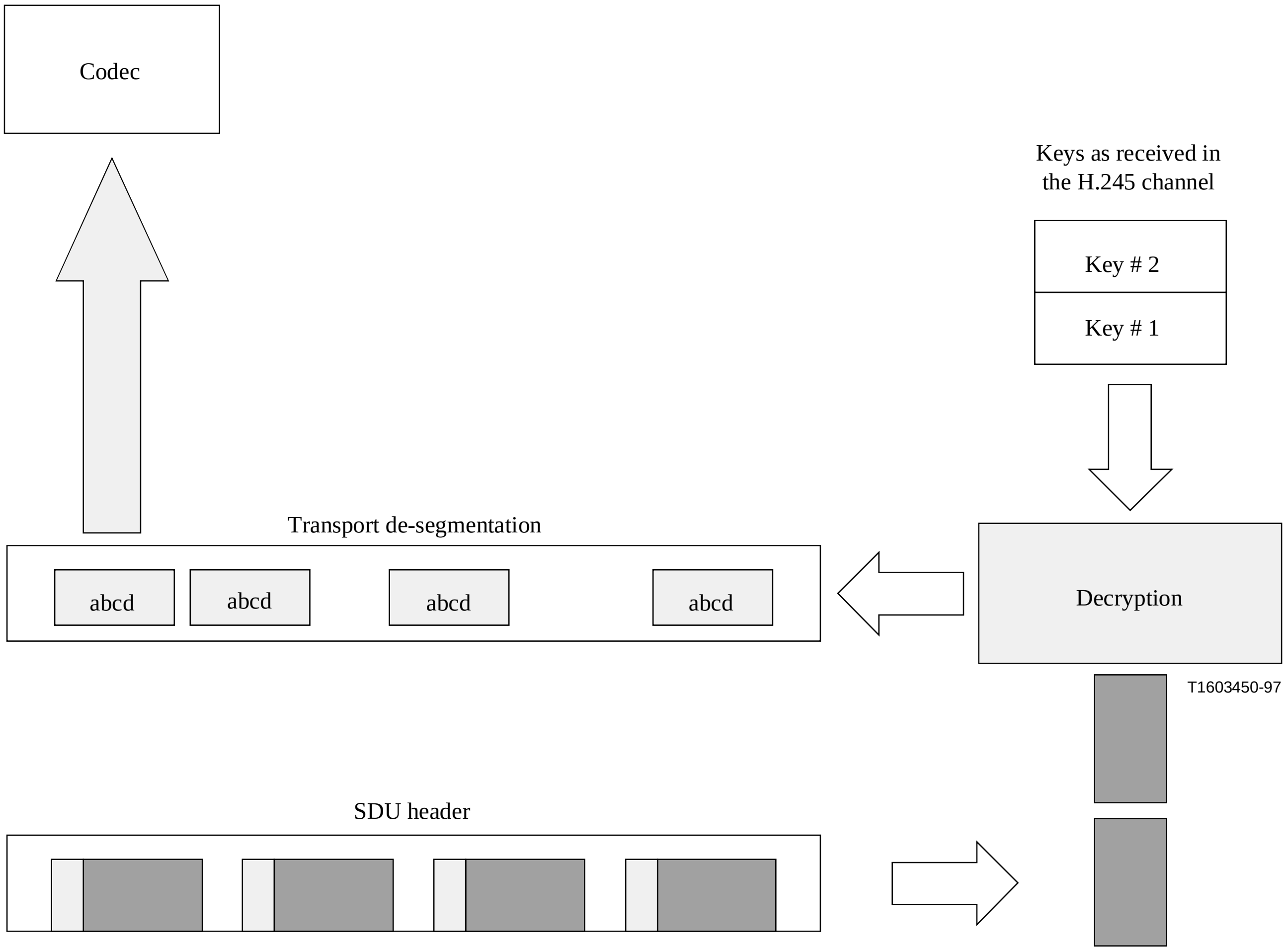


Figure 6/H.235 – Decryption of media

11.1 Media session keys

Included in the **encryptionUpdate** is the **h235Key**. The **h235Key** is ASN.1 encoded within the context of the H.235 ASN.1 tree and passed as an opaque octet string with respect to H.245. The key may be protected by utilizing one of the three possible mechanisms as they are passed between two endpoints.

- If the H.245 channel is secure, no additional protection is applied to the key material. The key is passed "in the clear" with respect to this field; the ASN.1 choice of **secureChannel** is utilized.
- If a secret key and algorithm has been established outside the H.245 channel as a whole (i.e. outside H.323 or on a **h235Control** logical channel), the shared secret is used to encrypt the key material; the resultant enciphered key is included here. In this case, the ASN.1 choice of **sharedSecret** is used.
- Certificates may be used when the H.245 channel is not secure, but may also be used in addition to the secure H.245 channel. When certificates are utilized, the key material is enciphered using the certificate's public key and the ASN.1 construct **certProtectedKey**.

At any point in a conference, a receiver (or transmitter) may request a new key (**encryptionUpdateRequest)**. One reason it might do this is if it suspects that it has lost synchronization of one of the logical channels. The master receiving this request shall generate new key(s) in response to this command. The master may also decide asynchronously to distribute new key(s), if so it shall use the **encryptionUpdate** message.

After receiving an **encryptionUpdateRequest**, a master shall send out **encryptionUpdate**. If the conference is a multipoint one, the MC (also the master) should distribute the new key to all receivers before it gives this key to the transmitter. The transmitter of the data on the logical channel shall utilize the new key at the earliest possible time after receiving the message.

A transmitter (assuming it is not the master) may also request a new key. If the transmitter is part of a multipoint conference the procedure shall be as follows:

- The transmitter shall send the **encryptionUpateRequest** to the MC (master).
- The MC should generate a new key(s) and send an **encryptionUpdate** message to all conference participants except the transmitter.
- After distributing the new keys to all other participants, the MC shall send the **encryptionUpdate** to the transmitter. The transmitter shall then utilize the new key.

12 Security error recovery

This Recommendation does not specify or recommend any methods by which endpoints may monitor their absolute privacy. It does however recommend actions to be taken when privacy loss is detected.

If either endpoint detects a breach in the security of the call connection channel (e.g. H.225.0 for H.323), it should immediately close the connection following the protocol procedures appropriate to the particular endpoint [for 8.5/H.323 with the exception of step 5)].

If either endpoint detects a breach in the security of the H.245 channel or the secured data (**h235Control**) logical channel, it should immediately close the connection following the protocol procedures appropriate to the particular endpoint [for 8.5/H.323 with the exception of step 5)].

If any endpoint detects a loss of privacy on one of the logical channels, it should immediately request a new key **(encryptionUpdateRequest)** and/or close the logical channel. At the discretion of the MC(U), a loss of privacy on one logical channel may cause all other logical channels to be closed and/or re-keyed at the discretion of the MC(U). MC(U) shall forward **encryptionUpdateRequest, encryptionUpdate** to any and all endpoints affected.

At the discretion of the MC(U), a security error on an individual channel may cause the connections to be closed on all of the conference endpoints – thus ending the conference.

ANNEX A

H.235 ASN.1

```
H235-SECURITY-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- EXPORTS All

ChallengeString     ::= OCTET STRING (SIZE(8..128))
TimeStamp                ::= INTEGER(1..4294967295) -- seconds since 00:00 1/1/1970 UTC
RandomVal                ::= INTEGER
Password                 ::= BMPString (SIZE (1..128))
Identifier               ::= BMPString (SIZE (1..128))
KeyMaterial              ::= BIT STRING(SIZE(1..2048))
```

```
NonStandardParameter ::= SEQUENCE
{
        nonStandardIdentifier      OBJECT IDENTIFIER,
        data                       OCTET STRING
}

-- if local octet representations of these bit strings are used they shall
-- utilize standard Network Octet ordering (e.g. Big Endian)
DHset ::= SEQUENCE
{
        halfkey                    BIT STRING (SIZE(0..2048)), -- = g^x mod n
        modSize                    BIT STRING (SIZE(0..2048)), --  n
        generator                  BIT STRING (SIZE(0..2048)), -- g
        ...
}

TypedCertificate ::= SEQUENCE
{
        type                       OBJECT IDENTIFIER,
        certificate                OCTET STRING,
        ...
}

AuthenticationMechanism ::=CHOICE
{
        dhExch                     NULL, -- Diffe-Hellman
        pwdSymEnc                  NULL, -- password with symmetric encryption
        pwdHash                    NULL, -- password with hashing
        certSign                   NULL, -- Certificate with signature
        ipsec                      NULL, -- IPSEC based connection
        tls                        NULL,
        nonStandard                NonStandardParameter, -- something else.
        ...
}

ClearToken            ::= SEQUENCE  -- a "token" may contain multiple value types.
{
        timeStamp                  TimeStamp OPTIONAL,
        password                   Password OPTIONAL,
        dhkey                      DHset OPTIONAL,
        challenge                  ChallengeString OPTIONAL,
        random                     RandomVal OPTIONAL,
        certificate                TypedCertificate OPTIONAL,
        generalID                  Identifier OPTIONAL,
        nonStandard                NonStandardParameter OPTIONAL,
        ...
}

--
-- Start all the cryptographic parameterized types here...
--


SIGNED { ToBeSigned } ::= SEQUENCE {
   toBeSigned         ToBeSigned,
   algorithmOID       OBJECT IDENTIFIER,
   paramS             Params,        -- any "runtime" parameters
   signature          BIT STRING
} ( CONSTRAINED BY { -- Verify or Sign Certificate -- } )
```

```
ENCRYPTED { ToBeEncrypted } ::= SEQUENCE {
  algorithmOID              OBJECT IDENTIFIER,
  paramS                    Params,       -- any "runtime" parameters
  encryptedData             OCTET STRING
} ( CONSTRAINED BY { -- Encrypt or Decrypt -- ToBeEncrypted } )

HASHED { ToBeHashed } ::= SEQUENCE {
  algorithmOID              OBJECT IDENTIFIER,
  paramS                    Params,       -- any "runtime" parameters
  hash                      BIT STRING
} ( CONSTRAINED BY { -- Hash -- ToBeHashed } )

IV8 ::= OCTET STRING (SIZE(8))

-- signing algorithm used must select one of these types of parameters
-- needed by receiving end of signature.

Params ::= SEQUENCE {
        ranInt          INTEGER OPTIONAL, -- some integer value
        iv8             IV8 OPTIONAL,     -- 8 octet initialization vector
        ...
}

EncodedGeneralToken ::= TYPE-IDENTIFIER.&Type (ClearToken -- general usage token
PwdCertToken ::= ClearToken (WITH COMPONENTS {..., timeStampPRESENT, generalIDPRESENT})
EncodedPwdCertToken ::= TYPE-IDENTIFIER.&Type (PwdCertToken)

CryptoToken::= CHOICE
{

        cryptoEncryptedToken SEQUENCE -- General purpose/application specific token
        {
                tokenOID        OBJECT IDENTIFIER,
                token           ENCRYPTED { EncodedGeneralToken }
        },
        cryptoSignedToken  SEQUENCE -- General purpose/application specific token
        {
                tokenOID        OBJECT IDENTIFIER,
                token           SIGNED { EncodedGeneralToken }
        },
        cryptoHashedToken SEQUENCE -- General purpose/application specific token
        {
                tokenOID                OBJECT IDENTIFIER,
                hashedVals              ClearToken,
                token HASHED { EncodedGeneralToken }
        },
        cryptoPwdEncr       ENCRYPTED { EncodedPwdCertToken },
        ...
}

-- These allow the passing of session keys within the H.245 OLC structure.
-- They are encoded as standalone ASN.1 and based as an OCTET STRING within H.245
H235Key      ::=CHOICE  -- this is used with the H.245 "h235Key" field
{
        secureChannel       KeyMaterial,
        sharedSecret        ENCRYPTED {EncodedKeySyncMaterial},
        certProtectedKey            SIGNED { EncodedKeySignedMaterial },
        ...
}
```

```
KeySignedMaterial ::= SEQUENCE {
        generalId               Identifier, -- slave’s alias
        mrandom       RandomVal, -- master’s random value
        srandom       RandomVal OPTIONAL, -- slave’s random value
        timeStamp               TimeStamp OPTIONAL, -- master’s timestamp for unsolicited EU
        encrptval               ENCRYPTED {EncodedKeySyncMaterial }
}
EncodedKeySignedMaterial ::= TYPE-IDENTIFIER.&Type (KeySignedMaterial)

H235CertificateSignature ::=SEQUENCE
{
        certificate                     TypedCertificate,
        responseRandom                  RandomVal,
        requesterRandom                 RandomVal OPTIONAL,
        signature                       SIGNED { EncodedReturnSig },
        …
}

ReturnSig ::= SEQUENCE {
        generalId                       Identifier, -- slave’s alias
        responseRandom                  RandomVal,
        requestRandom                   RandomVal OPTIONAL,
        certificate                     TypedCertificate OPTIONAL -- requested certificate
}

EncodedReturnSig ::= TYPE-IDENTIFIER.&Type (ReturnSig)
KeySyncMaterial    ::=SEQUENCE
{
        generalID               Identifier,
        keyMaterial             KeyMaterial,
        …
}
EncodedKeySyncMaterial::=TYPE-INDENTIFIER.&Type (KeySyncMaterial)



END   -- End of H235-SECURITY-MESSAGES DEFINITIONS
```

ANNEX B

H.323 specific topics

B.1 Background

Figure B.1 gives an overview of the scope of this Recommendation within Recommendation H.323.

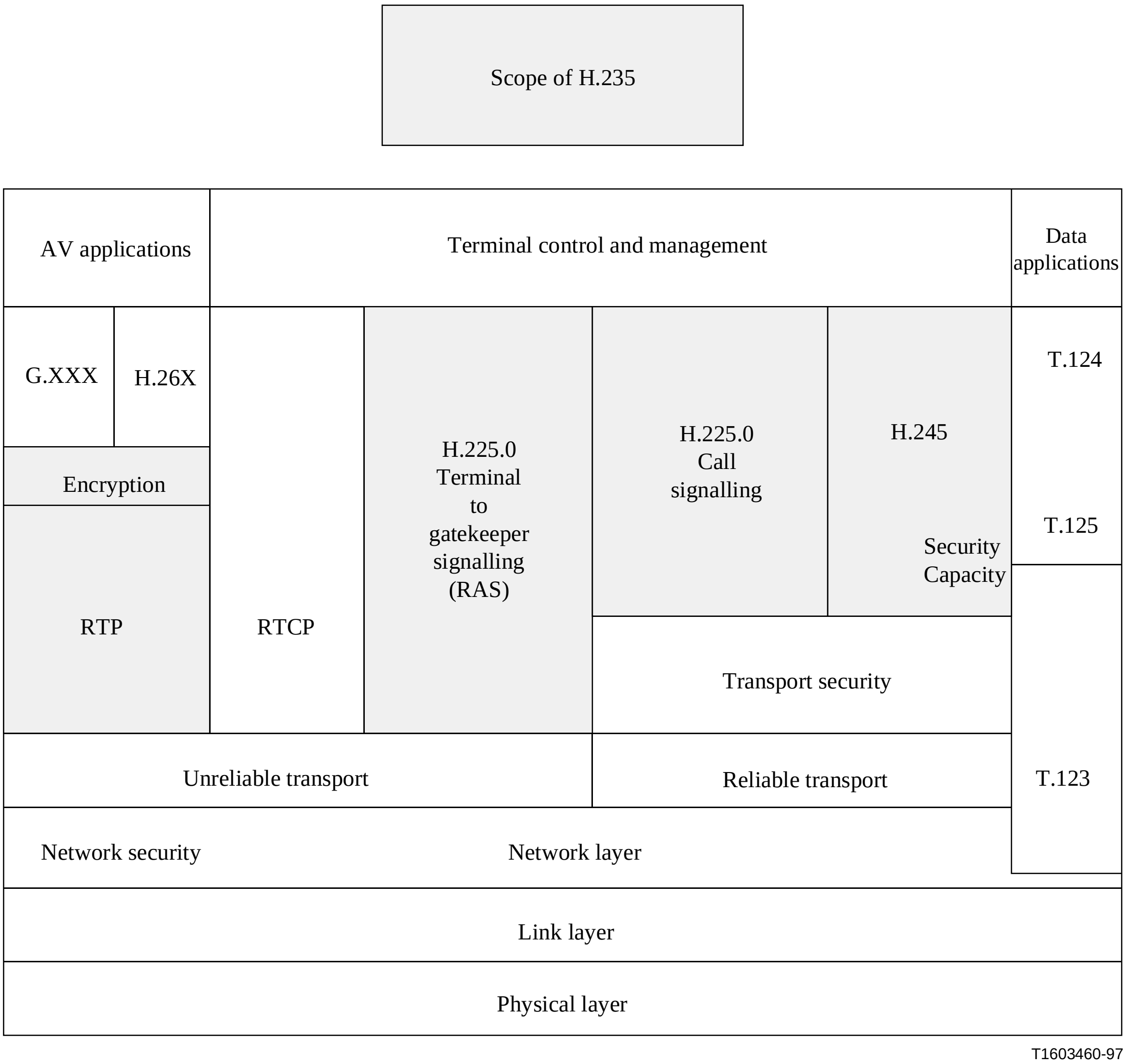


Figure B.1/H.235

For H.323, the signalling of usage of TLS, IPSEC or a proprietary mechanism on the H.245 control channel shall occur on the secured or unsecured H.225.0 channel during the initial Q.931 message exchange.

B.2 Signalling and procedures

The procedures outlined in clause 8/H.323 (Call signalling procedures) shall be followed. The H.323 endpoints shall have the ability to encode and recognize the presence (or absence) of security requirements (for the H.245 channel) signalled in the H.225.0 messages.

In the case where the H.225.0 channel itself is to be secured, the same procedures in clause 8/H.323, shall be followed. The difference in operation is that the communications shall only occur after connecting to the secure TSAP identifier and using the predetermined security modes (e.g. TLS). Due to the fact that the H.225.0 messages are the first exchanged when establishing H.323 communications, there can be no security negotiations "in band" for H.225.0. In other words, both parties must know *a priori* that they are using a particular security mode. For H.323 on IP, an alternative Well Known Port (1300) is utilized for TLS secured communications.

One purpose of H.225.0 exchanges as they relate to H.323 security, is to provide a mechanism to set up the secure H.245 channel. Optionally, authentication may occur during the exchange of H.225.0 messages. This authentication may be certificate or password based, utilizing encryption and/or hashing (i.e. signing). The specifics of these modes of operation are described in 10.2 to 10.3.4.

An H.323 endpoint that receives a SETUP message with the **h245SecurityCapability** set shall respond with the corresponding acceptable **h245SecurityMode** in the CONNECT message. In the cases in which there are no overlapping capabilities, the called terminal may refuse the connection by sending a **Release Complete** with the reason code set to ***SecurityDenied***. This error is intended to convey no information about any security mismatch and the calling terminal will have to determine the problem by some other means. In cases where the calling terminal receives a CONNECT message without sufficient or an acceptable security mode, it may terminate the call with a **Release Complete** with ***SecurityDenied***. In cases where the calling terminal receives a CONNECT message without any security capabilities, it may terminate the call with a **Release Complete** with ***undefinedReason***.

If the calling terminal receives an acceptable **h245Security** mode, it shall open and operate the H.245 channel in the indicated secure mode. Failure to set up the H.245 in the secure mode determined here should be considered a protocol error and the connection terminated.

B.2.1 Revision 1 compatibility

A security capable endpoint shall not return any security related fields, indications or status to the non-security capable endpoint. If a caller receives a SETUP message that does not contain the **H245Security** capabilities and/or authentication token, it may return a **ReleaseComplete** to refuse the connection; but it shall use the reason code of ***UndefinedReason*** in this case. In a corresponding manner, if a caller receives a CONNECT message without an **H245SecurityMode** and/or authentication token having sent a SETUP message with **H245Security** and/or authentication token, it may also terminate the connection by issuing a **ReleaseComplete** with a reason code of ***UndefinedReason***.

B.3 RTP/RTCP issues

The use of encryption on the RTP stream will follow the general methodology recommended in the document referenced in **[RTP]**. The encryption of the media shall occur in an independent, packet by packet basis[1]. The RTP header (including the payload header) shall not be encrypted. Synchronization of new keys and encrypted text is based upon dynamic payload type.

Initial encryption key is presented by the master in conjunction with the dynamic payload number (via **EncryptionSync** in H.245). The receiver(s) of the media stream shall start initial use of the key upon receipt of this payload number in the RTP header. New key(s) may be distributed at any time by the master endpoint. The synchronization of the newer key with the media stream shall be indicated by the changing of the payload type to a new dynamic value. Note that the specific values do not matter, as long as they change for every new key that is distributed.

It is assumed that encryption is applied just to the payload in each RTP packet, the RTP headers remaining in the clear. It is assumed that all RTP packets must be a multiple of whole octets. How the RTP packets are encapsulated at the transport or network layer is not relevant to this Recommendation. All modes must allow for lost (or out-of-sequence) packets, in addition to padding packets to an appropriate multiple of octets.

[1] It should be noted that if RTP packet size is larger than MTU size, partial loss (of fragment) will cause the whole RTP packet to be indecipherable.

Deciphering the stream must be stateless due to the fact that packets may be lost; each packet decipherable on its own merits. Two requirements of block algorithm mode shall operate as follows:

a) Initialization vectors

Most block modes involve some "chaining"; each encryption cycle depends in some way on the output of the previous cycle. Therefore, at the beginning of a packet, some initial block value [usually called an Initialization Vector (IV)] must be provided in order to start the encryption process. Independent of how many stream octets are processed on each encryption cycle, the length of the IV is always equal to the length of a block. All modes except Electronic Code Book (ECB) mode require an IV. In all cases, an IV shall be constructed from the first B (where B is the block size) octets of: (Seq# + Timestamp). This pattern should be repeated until enough octets have been generated. It should be noted that the IV generated in this manner may produce a key pattern that is considered "weak" for a particular algorithm.

b) Padding

ECB and CBC modes always process the input stream a block at a time, and, while CFB and OFB can process the input in any number of octets, N (≤ B), it is recommended that N = B.

Two methods are available to handle packets whose payload is not a multiple of blocks:

1) Ciphertext Stealing for ECB and CBC; Zero pad for CFB and OFB.

2) Padding in the manner prescribed by **[RTP]** (Section 5.1).

[RTP] Section 5.1 describes a method of padding in which the payload is padded to a multiple of blocks, the last octet set with the number of padding octets (including the last), and the P bit set in the RTP header. The value of the pad should be determined by the normal convention of the cipher algorithm.

All H.235 implementations shall support both schemes. The scheme in use can be deduced as follows: if the P bit is set in the RTP header, then the packet is padded; if the packet is not a multiple of B and the P bit is not set, then Ciphertext Stealing applies, else the packet is a multiple of B, and padding does not apply.

Integrity and replay protection of the RTP stream is for further study.

Application of cryptographic techniques to RTCP elements is for further study.

B.4 RAS signalling/procedures for authentication

B.4.1 Introduction

This annex will not explicitly provide any form of message privacy between gatekeepers and endpoints. There are two types of authentication that may be utilized. The first type is symmetric encryption-based that requires no prior contact between the endpoint and gatekeeper. The second type is subscription-based and will have two forms, password or certificate. All of these forms are derived from the procedures shown in 10.2, 10.3.2, 10.3.3 and 10.3.4. In this annex, the generic labels (EPA and EPB) shown in the aforementioned subclauses will represent the endpoint and gatekeeper respectively.

B.4.2 Endpoint-gatekeeper authentication (non-subscription based)

This mechanism may provide the gatekeeper with a cryptographic link that a particular endpoint which previously registered, is the same one that issues subsequent RAS messages. It should be noted that this may not provide any authentication of the gatekeeper to the endpoint, unless the optional signature element is included. The establishment of the identity relationship occurs when the terminal issues the **GRQ** as outlined in 7.2.1/H.323. The Diffie-Hellman exchange shall occur in conjunction with the **GRQ** and **GCF** messages as shown in the first phase of 10.2. This shared secret key shall now be used on any subsequent **RRQ**/**URQ** from the terminal to the gatekeeper. If a gatekeeper operates in this mode and receives a **GRQ** without a token containing the *DHset* or an acceptable algorithm value, it shall return a **securityDenial** reason code in the **DRJ**.

The Diffie-Hellman shared secret key as created during the **GRQ**/**GCF** exchange may be used for authentication on subsequent **xRQ** messages. The following procedures shall be used to complete this mode of authentication.

Terminal (**xRQ**):

1) The terminal shall provide all of the information in the message as described in the appropriate subclauses of Recommendation H.225.0.

2) The terminal shall encrypt the **GatekeeperIdentifier** (as returned in the **GCF**) using the shared secret key that was negotiated. This shall be passed in a **cryptoToken** as the **generalID**.

The 16 bits of the **random** and then the **requestSeqNum** shall be XOR'd with each 16 bits of the **GatekeeperIdentifier**. If the **GatekeeperIdentifier** does not end on an even 16 boundary, the last 8 bits of the **GatekeeperIdentifier** shall be XOR'd with the least significant octet of the random value and then **requestSeqNum**. The **GatekeeperIdentifier** shall be encrypted using the selected algorithm in the **GCF** (integrity) and utilizing the entire shared secret.

In order to cryptographically link this and subsequent messages with the original registrant (the endpoint that issued the **RRQ**) the most recent **random** value returned shall be utilized (this value may be one newer than the value returned in the **RCF** – from a later **xCF** message).

Gatekeeper (**xCF**/**xRJ**):

1) Gatekeeper shall encrypt its **GatekeeperIdentifier** (following the above procedure) with the shared secret key associated with the endpoint alias and compare this to the value in the **xRQ**.

2) Gatekeeper shall return **xRJ** if the two encrypted values do not match.

3) If **GatekeeperIdentifier** matches gatekeeper shall apply any local logic and respond with **xCF** or **xRJ**.

4) If an **xCF** is sent by the gatekeeper, it should contain an assigned **EndpointIdentifier** and a new random value in the **random** field of a **clearToken**.

Refer to the second phase of Figure 1 in 10.2 for a graphical representation of this exchange. The gatekeeper knows which shared secret key to use to decipher the gatekeeper identifier by the alias name in the message.

B.4.3 Endpoint-gatekeeper authentication (subscription-based)

All RAS messages other than GRQ/GCF should contain the authentication tokens required by the specific mode of operation. There are three different variations that may be implemented depending on requirements and environment:

1) password-based with symmetric encryption;

2) password-based with hashing;

3) certificate-based with signatures.

In all cases, the token will contain the information as described in the following subclauses depending on the variation chosen. If a gatekeeper operates in a secure mode and receives a RAS message without an acceptable token value, it shall return a **securityDenial** reason code in the reject message. In all cases, the return token from GK is optional; if omitted, only one-way authentication is achieved.

B.4.3.1 Password with symmetric encryption

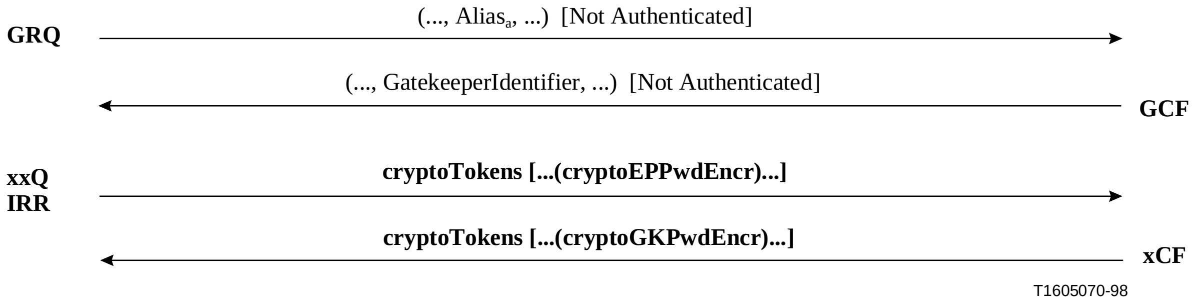

GRQ
(..., Alias$_a$, ...) [Not Authenticated]
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
xxQ
IRR
cryptoTokens [...(cryptoEPPwdEncr)...]
cryptoTokens [...(cryptoGKPwdEncr)...]
xCF
T1605070-98

Figure B.2/H.235

B.4.3.2 Password with hashing

It is assumed that an alias and associated password are exchanged out of band to this particular message exchange.

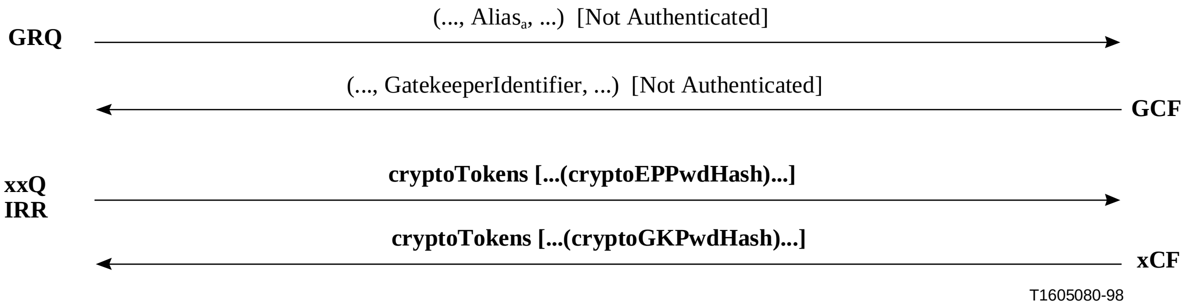

GRQ
(..., Alias$_a$, ...) [Not Authenticated]
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
xxQ
IRR
cryptoTokens [...(cryptoEPPwdHash)...]
cryptoTokens [...(cryptoGKPwdHash)...]
xCF
T1605080-98

Figure B.3/H.235

B.4.3.3 Certificate-based with signatures

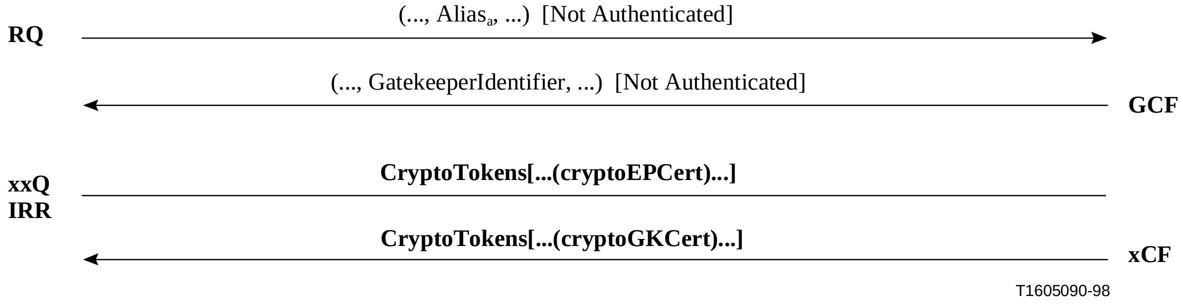

RQ
(..., Alias$_a$, ...) [Not Authenticated]
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
xxQ
IRR
CryptoTokens[...(cryptoEPCert)...]
CryptoTokens[...(cryptoGKCert)...]
xCF
T1605090-98

Figure B.4/H.235

B.5 Non-terminal interactions

B.5.1 Gateway

As stated in 6.6, an H.323 gateway should be considered a trusted element. This includes protocol gateways (H.323-H.320 etc., …) and security gateways (proxy/firewalls). The media privacy can be assured between the communicating endpoint and the gateway device; but what occurs on the far side of the gateway should be considered insecure by default.

ANNEX C

H.324 specific topics

For further study.

APPENDIX I

H.323 implementation details

I.1 Ciphertext padding methods

There is a description of Ciphertext Stealing in **[Schneier]**, Pages 191 and 196. Figures I.1 to I.5 illustrate the technique.

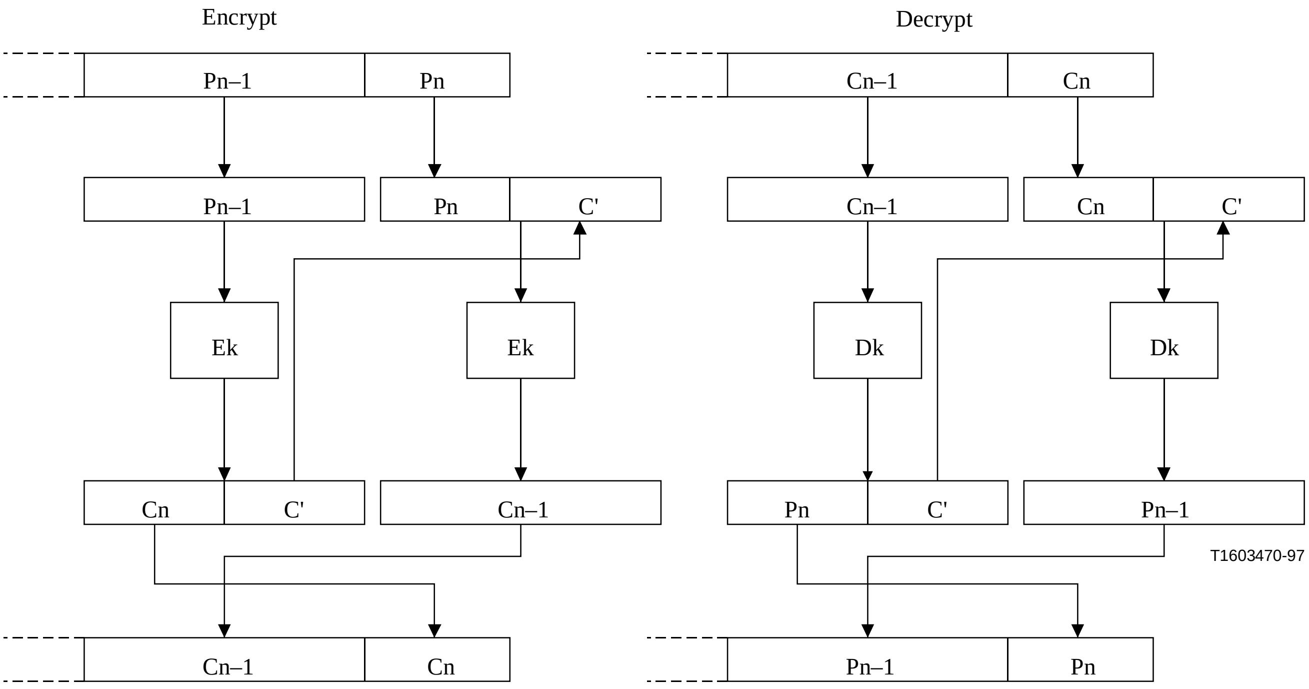


Figure I.1/H.235 – Ciphertext stealing in ECB mode

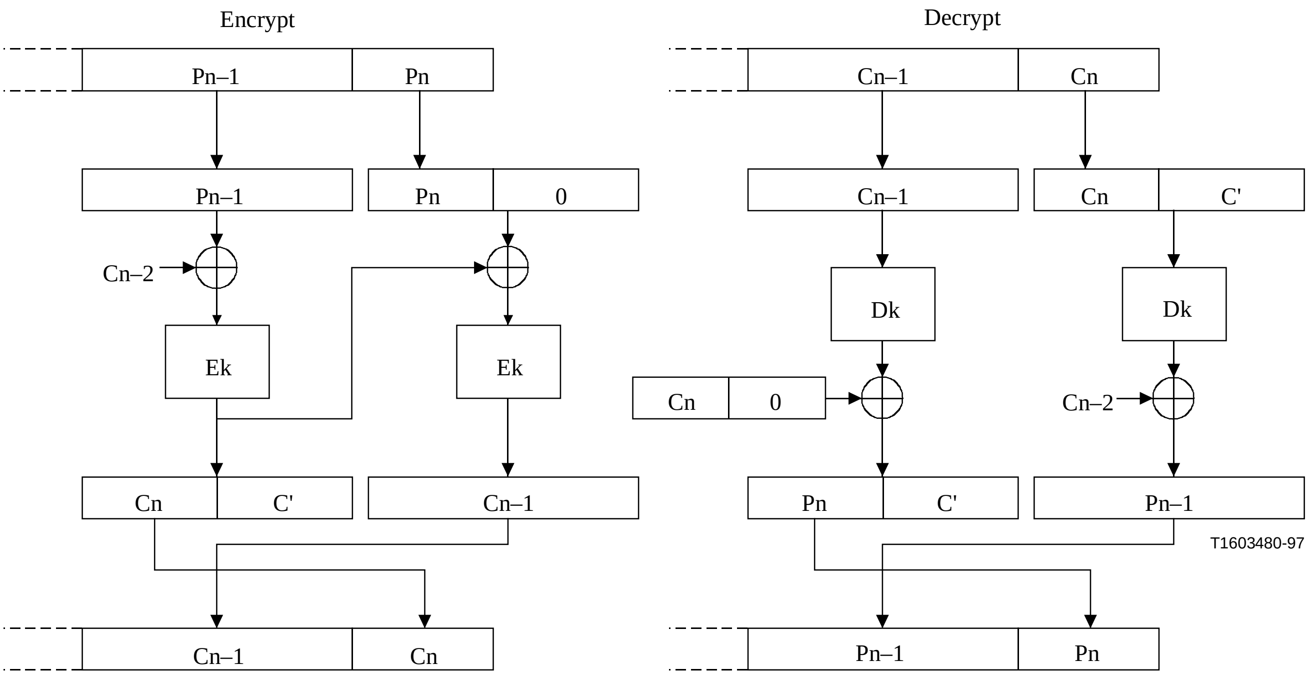


Figure I.2/H.235 – Ciphertext stealing in CBC mode

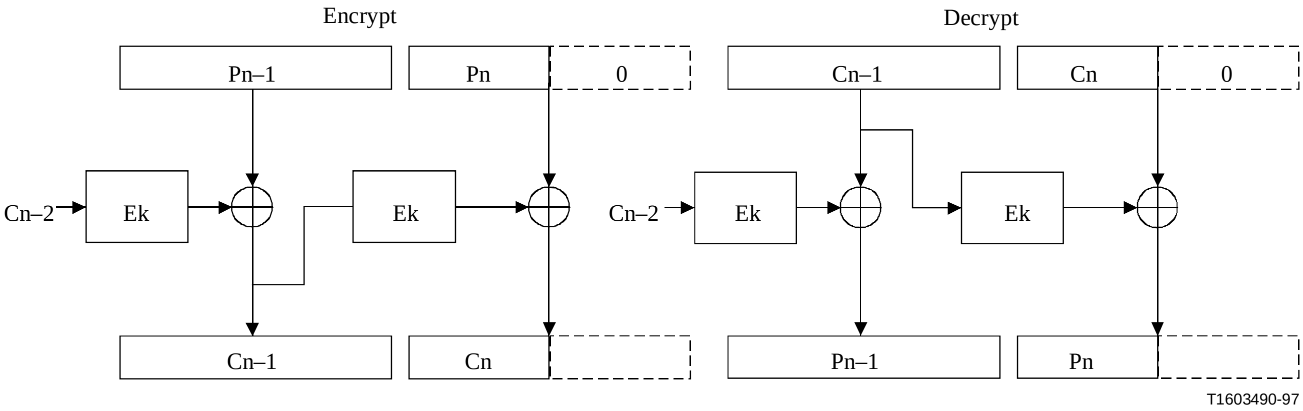
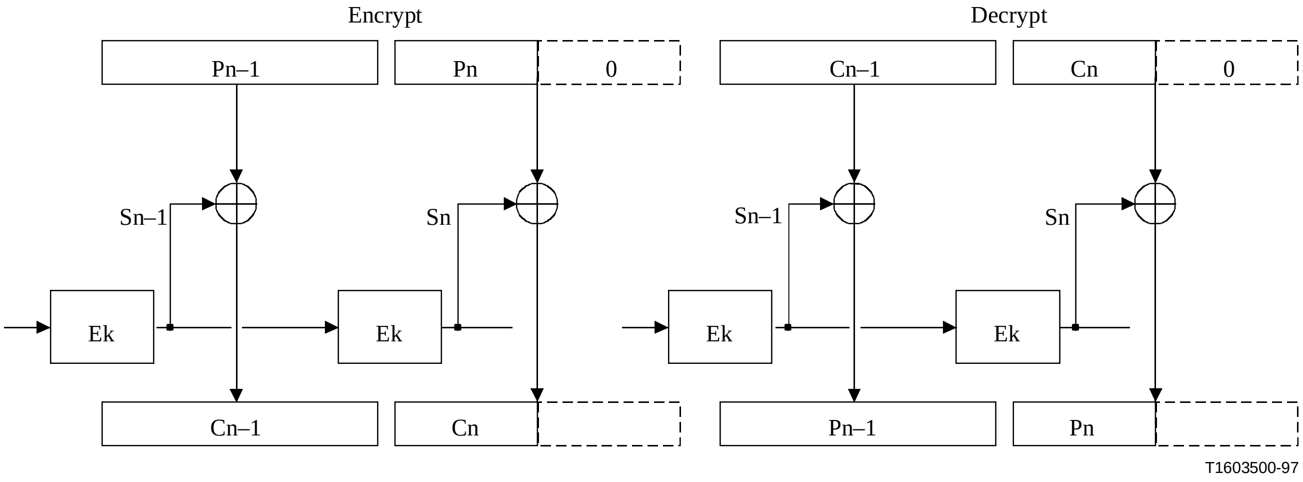


Figure I.3/H.235 – Zero padding in CFB mode



NOTE – Si is the result of repetitive encryption (i.e. permutations) of the IV.

Figure I.4/H.235 – Zero padding in OFB mode

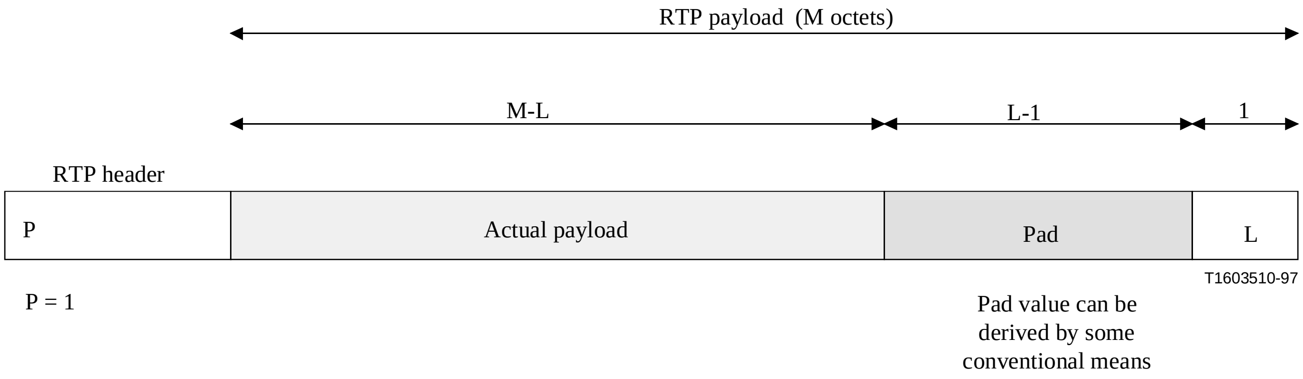


Figure I.5/H.235 – Padding as prescribed by RTP

I.2 New keys

The procedures outlined in 8.5/H.323 are completed by an MC to eject a participant from a conference. The master may generate new encryption keys for the logical channels (and not distribute them to the ejected party); this may be used to keep the ejected party from monitoring the media streams.

I.3 H.323 trusted elements

In general, MC(U)s, gateways, and gatekeepers (if implementing the gatekeeper-routed model) are trusted with respect to the privacy of the control channel. If the connections establishment channel (H.225.0) is secured *and* routed through the gatekeeper, it must also be trusted. If any of these H.323 components must operate on the media streams (i.e. mixing, transcoding) then, by definition, they shall also be trusted for the media privacy.

Firewall Proxies (though not H.323-specific elements) may also be trusted, since they terminate connections, and may well have to manipulate the messages and media streams.

I.4 Implementation examples

These next subclauses describe example implementations that might be developed within the H.235 framework. These are not intended to constrain the many other possibilities available within this Recommendation, but rather to give more concrete examples of usage within Recommendation H.323.

I.4.1 Tokens

This subclause will describe an example usage of security tokens to obscure or hide destination addressing information. The example scenario is an endpoint which wishes to make a call to another endpoint utilizing its well-known alias. More specifically, this involves an H.323 endpoint, gatekeeper, POTS-gateway, and telephone as illustrated below.

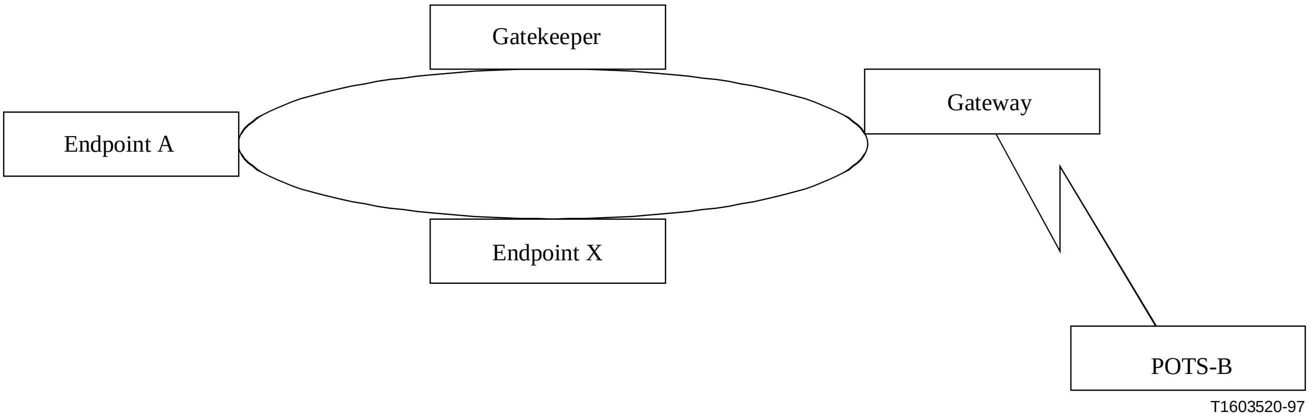


Figure I.6/H.235

Currently, H.323 may operate in a manner similar to a telephone network with caller-ID. This scenario will illustrate a situation in which the *caller* does not want to expose its physical address, while still allowing the call to complete. This may be important in POTS-H.323 gateways, where the target phone number may need to stay private.

Assume that EPA is trying to call POTS-B, and POTS-B does not want to expose its E.164 phone number to EPA. (How this policy is established is beyond the scope of this example.)

- EPA will send an ARQ to its gatekeeper to resolve the address of the POTS telephone as represented by its alias/GW. The gatekeeper would recognize this as a "private" alias, knowing that in order to complete the connection it must return the POTS-gateway address (similar to returning the address of an H.320 gateway if an H.320 endpoint is called by an H.323 endpoint).
- In the returned ACF, the gatekeeper returns the POTS-gateway's address as expected. The addressing information that is required to dial to the end telephone (i.e. the telephone number) is returned in an encrypted token included in the ACF. This encrypted token contains the actual E.164 (phone number) of the telephone which cannot be deciphered nor understood by the caller (i.e. EPA).
- The endpoint issues the SETUP message to the gateway device (whose call signalling address was returned in the ACF) including the opaque token(s) that it received with the ACF.

- The gateway, upon receiving the SETUP, issues its ARQ to its gatekeeper, including any token(s) that were received in the SETUP.
- The gatekeeper is able to decipher the token(s) and return the phone number in the ACF.

Partial ASN.1 of an example token structure is shown below, with the field contents described. Assume we utilize the **cryptoEncodedGeneralToken** to contain the encrypted telephone number.

An implementation might choose a **tokenOID** denoting this token as containing the E.164 phone number. The particular method that is used to encrypt this phone number (for example, 56 bit DES) would be included in the "ENCRYPT" definition **algorithmOID**.

```
CryptoToken::= CHOICE
{
	cryptoEncodedGeneralToken SEQUENCE   -- General purpose/application specific token
	{
		tokenOID  OBJECT IDENTIFIER,
		ENCRYPTED { EncodedGeneralToken }
	},
.
.
. [abbreviated text]
.

}
```

The **CryptoToken** would be passed in the SETUP (from EPA to GW) and the **ARQ** (from the GW to the gatekeeper) messages as outlined above. After the gatekeeper decrypted the token (the telephone number) it would pass the clear version of this in the **clearToken**.

I.4.2 Password

In this example, it is assumed that the user is a subscriber to the gatekeeper (i.e. the user will be in its zone) and has an associated subscription ID and password. The user would register with the gatekeeper using the subscription ID (as passed in an alias – H323ID) and encrypting a challenge string presented by the gatekeeper. This assumes that the gatekeeper also knows the password associated with the subscription ID. The gatekeeper will authenticate the user by verifying that the challenge string was correctly encrypted.

The example registration procedure with gatekeeper authentication is as follows:

1) If the endpoint uses **GRQ** to discover a gatekeeper, one of the aliases in the message would be the subscription ID (as an **H323ID**). The **authenticationcapability** would contain an **AuthenticationMechanism** of **pwdSymEnc** and the **algorithmOIDs** would be set to indicate the entire set of encryption algorithms supported by the endpoint. (For example, one of these would be 56 bit DES in EBC mode.)

2) The gatekeeper would respond with **GCF** (assuming it recognizes the alias) carrying a **tokens** element containing one **ClearToken**. This **ClearToken** would contain both a **challenge** and a **timeStamp** element. The **challenge** would contain 16 octets. (To prevent replay attacks, the **ClearToken** should contain a **timeStamp**.) The **authenticationmode** should be set to **pwdSymEnc** and the **algorithmOID** should be set to indicate the encryption algorithm required by the gatekeeper (for example, 56 bit DES in EBC mode).

   If the gatekeeper does not support any of the **algorithmOIDs** indicated in the **GRQ**, then it would respond with a **GRJ** containing a **GatekeeperRejectReason** of **resourceUnavailable**.

3) The endpoint application should then attempt to register with (one of) the GK(s) that responded with a **GCF** by sending an **RRQ** containing a **cryptoEPPwdEncr** in the **cryptoTokens**. The **cryptoEPPwdEncr** would have the **algorithmOID** of the encryption algorithm agreed to in the **GRQ**/**GCF** exchange, and the encrypted challenge.

   The encryption key is constructed from the user's password using the procedure described in 10.3. The resulting octet "string" is then used as the DES key to encrypt the **challenge**.

4) When the gatekeeper receives the encrypted challenge in the **RRQ**, it would compare it to an identically generated encrypted challenge to authenticate the registering user. If the two encrypted strings do not match, the gatekeeper should respond with an **RRJ** with the **RegistrationRejectReason** set to **securityDenial**. If they match, the gatekeeper sends an **RCF** to the endpoint.

5) If the gatekeeper receives an **RRQ** which does not contain an acceptable **cryptoTokens** element, then it should respond with an **RRJ** with a **GatekeeperRejectReason** of **discoveryRequired**. The endpoint, upon receiving such an **RRJ** may perform discovery which will allow the gatekeeper/endpoint to exchange a new challenge. Note that the **GRQ** message may be unicast to the gatekeeper.

I.4.3 IPSEC

In general, IPSEC **[13/IPSEC]** can be used to provide authentication and, optionally confidentiality (i.e. encryption) at the IP layer transparent to whatever (application) protocol runs above. The application protocol does not have to be updated to allow this; only security policy at each end.

For example, to make maximum use of IPSEC for a simple point-to-point call, the following scenario could be followed:

1) The calling endpoint and its gatekeeper would set policy to require the use of IPSEC (authentication and, optionally, confidentiality) on the RAS protocol. Thus, before the first RAS message is sent from the endpoint to the gatekeeper, the ISAKMP/Oakley daemon on the endpoint will negotiate security services to be used on packets to and from the RAS channel's well-known port. Once negotiation is complete, the RAS channel will operate exactly as if it were not secured. Using this secure channel the gatekeeper will inform the endpoint of the address and port number of the call signalling channel in the called endpoint.

2) After obtaining the address and port number of the call signalling channel, the calling endpoint would dynamically update its security policy to require the desired IPSEC security on that address and protocol/port pair. Now, when the calling endpoint attempts to contact this address/port, the packets would be queued while an ISAKMP/Oakley negotiation is performed between the endpoints. Upon completion of this negotiation, an IPSEC Security Association (SA) for the address/port will exist and the Q.931 signalling can proceed.

3) On the Q.931 SETUP and CONNECT exchange, the endpoints can negotiate the use of IPSEC for the H.245 channel. This will allow the endpoints to again dynamically update their IPSEC policy databases to force the use of IPSEC on that connection.

4) As with the call signalling channel, a transparent ISAKMP/Oakley negotiation will take place before any H.245 packets are transmitted. The authentication performed by this ISAKMP/Oakley exchange will be the initial attempt at user-to-user authentication, and will set up a (probably) secure channel between the two users on which to negotiate the characteristics of the audio channel. If, after some person-to-person Q&A, either user is not satisfied with the authentication, different certificates can be chosen and the ISAKMP/Oakley exchange repeated.

5) After each H.245 ISAKMP/Oakley authentication, new keying material is exchanged for the RTP audio channel. This keying material is distributed by the master on the secure H.245 channel. Because the H.245 protocol is defined for the master to distribute the media keying material on the H.245 channel (to allow for multipoint communication), it is not recommended that IPSEC be used for the RTP channel.

An encrypted H.245 channel is a potential problem for proxy or NAT firewall, since the dynamically-assigned port numbers are carried in the H.245 protocol. Such firewalls would have to decipher, modify and re-encipher the protocol to operate correctly. For this reason, the "Security" Logical Channel was introduced into Recommendation H.245. If this channel is used, the H.245 channel can remain unsecured; authentication and key-generation would be done with the "Security" Logical Channel. Logical channel signalling would allow this channel to be protected with IPSEC, and the secret key used on the "Security" Logical Channel would be used to protect the **EncryptionSync** distributed by the master on the H.245 channel.

APPENDIX II

H.324 implementation details

For further study.

APPENDIX III

Other H-series implementation details

For further study.

APPENDIX IV

Bibliography

[Daemon]

– DAEMON (J.): Cipher and Hash function design, Ph.D. Thesis, Katholieke Universiteit Leuven, March 1995.

[IPSEC]

– ORMAN (H.K.): The Oakley Key Determination Protocol, draft-ietf-ipsec-oakley-02.txt, *Internet Engineering Task Force*, 1997.

– MAUGHAN (D.), SCHERTLER (M.), SCHNEIDER (M.), TURNER (J.): Internet Security Association and Key Management Protocol (ISAKMP), draft-ietf-ipsec-isakmp-08.text, *Internet Engineering Task Force*, 1997.

- KENT (S.), ATKINSON (R.): IP Authentication Header, draft-ietf-ipsec-auth-header-01.txt, *Internet Engineering Task Force*, 1997.
- HARKINS (D.), CARREL (D.): The resolution of ISAKMP with Oakley, draft-ietf-ipsec-isakmp-oakley-04.txt, *Internet Engineering Task Force*, 1997.

[RTP]

- SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.), JACOBSON (V.): RTP: A transport Protocol for Real-Time Applications, RFC 1889, *Internet Engineering Task Force*, 1996.

[Schneier]

- SCHNEIER (B.): Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition, John Wiley & Sons, Inc., 1995.

[TLS]

- DIEKS (T.), ALLEN (C.): The TLS Protocol Version 1.0, draft-ietf-tls-protocol-03.txt, *Internet Engineering Task Force*, 1997.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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	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	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
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 and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages