International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235.8

(09/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Systems aspects

H.323 security: Key exchange for SRTP using secure signalling channels

ITU-T Recommendation H.235.8

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.235.8

H.323 security: Key exchange for SRTP using secure signalling channels

Summary

The purpose of this Recommendation is to describe security procedures for key exchange for SRTP using secure signalling channels over H.323/H.235 networks.

This Recommendation should be used in conjunction with ITU-T Recs H.323 and H.225.0 versions 4 or later.

Source

This Recommendation H.235.8 was approved on 13 September 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 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

ITU-T Recommendation H.235.8

H.323 security: Key exchange for SRTP using secure signalling channels

1 Scope

The purpose of this Recommendation is to provide recommendations for security procedures to support the IETF Secure Real Time Protocol (SRTP) between H.323 Endpoints in cases where the cryptographic material for the media channel is transported over a secure signalling channel, e.g., IPsec (RFC 2401), TLS (RFC 2246) or other H.235 mechanisms. These security procedures are offered as an alternative to other H.235 security procedures that support SRTP.

This Recommendation describes procedures used to support the IETF Secure Real Time Protocol (SRTP) in ITU-T Rec. H.323. SRTP provides security services for RTP media and relies on separate protocols to provide key management services and the negotiation of cryptographic parameters. These procedures should not be used when the secure signalling channel terminates at an intermediate system, in such cases the SRTP cryptographic material should be transported by a secure end-to-end mechanism.

These procedures support the signalling, negotiation and transport of the SRTP cryptographic keys, authentication and encryption algorithm identifiers and other session parameters between H.323 Endpoints.

A key aspect of these procedures is that the H.245 slave as well as the H.245 master shall be able to generate and distribute cryptographic keys.

SRTP security capabilities may be exchanged using existing terminal capabilities exchange using h235SecurityCapability entries in the capabilityTable of the H.245 TerminalCapabilitySet message. The genericH235SecurityCapability field in the encryptionAuthenticationAndIntegrity field in the h235SecurityCapability entry contains the SrtpCryptoCapability field which will specify the SRTP crypto-suites.

A SRTP crypto parameter is specified to signal and negotiate SRTP cryptographic parameters. The definition of the crypto parameter in this Recommendation is limited to two-party unicast media streams where each source has a unique cryptographic key; support for multicast media streams or multipoint unicast streams is for further study.

The SRTP crypto parameter is intended to be able to establish the SRTP cryptographic parameters in a single message or a single round trip message exchange. In the case of a round trip message exchange, the cryptographic parameters may be negotiated. For example, in Fast Connect, the offering H.323 Endpoint sends a set of offered SRTP crypto parameters to the answering H.323 Endpoint, each offer encapsulated in a separate H.245 OpenLogicalChannel message. The answering H.323 Endpoint may then accept one of the offered parameters and respond with an answer that includes the selected parameter subset encapsulated in a H.245 OpenLogicalChannel message.

In the case of a single message exchange there is no negotiation. The offering H.323 Endpoint sends the SRTP crypto parameters to the answering H.323 Endpoint which either accepts the offered parameters or rejects the call.

Public key cryptography procedures may be added to provide end-to-end confidentiality and authentication of the SRTP session key material exchanged between H.323 Endpoints by encrypting and then signing the SRTP key material in the case where the encapsulating security protocol, e.g., IPsec, TLS, terminates on an intermediate device and, therefore, does not provide end-to-end security.

2 References

2.1 Normative references

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

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

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

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

– ITU-T Recommendation H.460.11 (2004), *Delayed call establishment within H.323 Systems*.

– IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.

– IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.

– IETF RFC 2733 (1999), *An RTP Payload Format for Generic Forward Error Correction*.

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

– IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

– IETF RFC 3711 (2004), *The Secure Real-time Transport Protocol (SRTP)*.

– IETF RFC 3852 (2004), *Cryptographic Message Syntax (CMS)*.

2.2 Informative references

– IETF Draft, F. Andreasen, M. Baugher, D. Wing: *Session Description Protocol Security Descriptions for Media Streams*, <draft-ietf-mmusic-sdescriptions-11.txt>.

3 Symbols and abbreviations

This Recommendation uses the following abbreviations:

AES Advanced Encryption Algorithm

ASN.1 Abstract Syntax Notation One

CA Certificate Authority

CEK Content Encryption Key

CMS Cryptographic Message Syntax

EP Endpoint

FEC Forward Error Correction

FFS For Further Study

F8 UMTS Encryption Algorithm

GK Gatekeeper

GW Gateway

HMAC Keyed-Hash Message Authentication Code

IETF Internet Engineering Task Force

KDR Key Derivation Rate

MAC Message Authentication Code

MKI Master Key Identifier

OID Object Identifier

OLC Open Logical Channel

PKI Public Key Infrastructure

RAS Registration, Admission, Status

ROC Roll-over Counter

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

SHA1 Secure Hash Algorithm 1

SRTCP Secure Real-time Transport Control Protocol

SRTP Secure Real-time Transport Protocol

SSRC Synchronization Source

TLS Transport Level Security

WSH Window Size Hint

4 Parameter description

SRTP cryptographic capability and key material is exchanged using two parameters:

– **SrtpCryptoInfo** within **StrpCryptoCapability** shall contain the crypto-suite and session parameters. The **SrtpCryptoInfo** parameter shall be transported in the H.245 **genericH235SecurityCapability** parameter to signal and negotiate SRTP cryptographic parameters.

– **SrtpKeyParameters** within **SrtpKeys** shall contain the SRTP key material. The **SrtpKeys** container in the H.245 **h235Key** parameter shall transport one or more **SrtpKeyParameters** with the SRTP keys.

The use of the SRTP crypto parameters in this Recommendation is limited to two-party unicast media streams where each source has a unique cryptographic key; support for multicast media streams or multipoint unicast streams is for further study.

4.1 SRTP parameter transport

A full-duplex SRTP media connection consists of two unidirectional channels, one in each direction. Each crypto-offer is transported in a separate H.245 **OpenLogicalChannel** message.

4.1.1 SrtpKeys transport

The SRTP cryptographic key material **SrtpKeys** shall be transported in the **genericKeyMaterial** field of the **secureSharedSecret** (**V3KeySyncMaterial**) parameter contained within the **h235Key** container in the **encryptionSync** parameter of H.245 **OpenLogicalChannel** messages.

The SRTP cryptographic key content in the **genericKeyMaterial** container shall be identified using the H.235.8 object identifier value (see Table 1) in the **standard** field of **capabilityIdentifier**

within the **genericH235SecurityCapability** field of **encryptionAuthenticationAndIntegrity** in **h235Media** of the OLC **dataType**.

Alternative **OpenLogicalChannel** proposals for the same channel that contain the same **sessionID** value in **H2250LogicalChannelParameters** may use the same crypto-offer. Since only one of these alternate sessions will be accepted key uniqueness will be guaranteed.

4.1.2 SrtpCryptoCapability transport

The **SrtpCryptoCapability** parameter shall be transported in the **genericH235SecurityCapability** field of **encryptionAuthenticationAndIntegrity** in **h235Media** of the **dataType** parameter of **OpenLogicalChannel** messages.

The H.245 **TerminalCapabilitySet** message may include one or more **h235SecurityCapability** entries in the **capabilityTable**. In order to indicate support for these procedures the H.323 Endpoint shall set the **genericH235SecurityCapability** within **encryptionAuthenticationAndIntegrity** in an **h235SecurityCapability** entry as follows:

- **capabilityIdentifier** shall contain the H.235.8 OID (see Table 1) in **standard** field;
- **maxbitRate**, **collapsing**, **nonCollapsing**, and **transport** shall be unused;
- **nonCollapsingRaw** shall contain the **SrtpCryptoCapability** parameter.

Table 1/H.235.8 – H.235.8 object identifier

OID Value
{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 90 }

4.2 SrtpCryptoCapability parameter description

The **SrtpCryptoCapability** may contain one or more **SrtpCryptoInfo** parameters that may be used to specify capabilities for the SRTP session. The **BOOLEAN OPTIONAL** elements shall be interpreted as:

1) if FALSE, the capability is not supported;
2) if TRUE, the capability is supported and required;
3) if absent, the capability is supported but not required.

When using **SrtpCryptoCapability** in a capability exchange, it is possible to indicate all acceptable options within a single generic capability. In this use, the omission of a **BOOLEAN OPTIONAL** element will be interpreted to mean that the capability is supported but not required.

When used in an OLC **dataType** expression, only one option may be used. For this purpose, the following rules shall be observed:

- **FecOrder** may contain only one of the optional values.
- In **SrtpSessionParameters**, the **BOOLEAN OPTIONAL** values must be either TRUE or FALSE.
- **SrtpCryptoCapability** shall contain only a single **SrtpCryptoInfo** element.

The **SrtpCryptoInfo** parameter consists of mandatory **cryptoSuite** field and optional **sessionParams** and **allowMKI** fields which are described below.

Table 2/H.235.8 – H.235.8 crypto-suite object identifiers

Crypto-suite	OID value
AES_CM_128_HMAC_SHA1_80	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 91 }
AES_CM_128_HMAC_SHA1_32	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 92 }
F8_128_HMAC_SHA1_80	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 93 }

4.2.1 cryptoSuite

The object identifier (see Table 2) in the field **cryptoSuite** identifies the desired encryption and authentication algorithms to be used in the SRTP session. The SRTP specification has many parameters that are bundled into three options, called "Crypto Suites". These are extensible in that new Crypto Suites can be added. The three Crypto Suites that are defined are AES_CM_128_HMAC_SHA1_80, AES_CM_128_HMAC_SHA1_32, and F8_128_HMAC_SHA1_80. The SRTP parameters that are bundled into each of these are shown as rows in Table 3.

Table 3/H.235.8 – Crypto-suites default values

SRTP parameter	AES_CM_128_ HMAC_SHA1_80	AES_CM_128_ HMAC_SHA1_32	F8_128_ HMAC_SHA1_80
Master key length	128 bits	128 bits	128 bits
Salt value	112 bits	112 bits	112 bits
Lifetime	2^31 packets	2^31 packets	2^31 packets
Cipher	AES Counter	AES Counter	F8
Encryption key	128 bits	128 bits	128 bits
MAC	HMAC-SHA1	HMAC-SHA1	HMAC-SHA1
Authentication tag length	80 bits	32 bits	80 bits
SRTP auth. Key length	160 bits	160 bits	160 bits
SRTCP auth. Key length	160 bits	160 bits	160 bits

The field **cryptoSuite** is a negotiated parameter.

4.2.2 sessionParams

Session parameters may be either negotiated or declarative; the definition of a specific session parameter shall indicate whether it is negotiated or declarative. Negotiated parameters apply to data sent in both directions, whereas declarative parameters apply only to media sent by the entity that generated the session description. Thus, a declarative parameter in an offer applies to media sent by the offerer, whereas a declarative parameter in an answer applies to media sent by the answerer.

The optional field **sessionParams** contains SRTP session parameters.

4.2.2.1 kdr

KDR specifies the Key Derivation Rate, as described in section 4.3.1 of RFC 3711. The value shall be an integer in the set {1, 2, ..., 24}, which denotes a power of 2 from 2^1 to 2^24, inclusive. The SRTP key derivation rate controls how frequently a new session key is derived from an SRTP master key (RFC 3711). When the key derivation rate is not specified (i.e., the KDR parameter is omitted), a single initial key derivation is performed (RFC 3711). KDR is a declarative parameter.

4.2.2.2 unencryptedSrtp

This is an optional Boolean field: if present it signals that the SRTP packet payloads are not encrypted. This is a negotiated parameter.

4.2.2.3 unencryptedSrtcp

This is an optional Boolean field: if present it signals that the SRTCP packet payloads are not encrypted. This is a negotiated parameter.

4.2.2.4 unauthenticatedSrtp

SRTP and SRTCP packet payloads are authenticated by default. This is an optional Boolean field, if present it signals that the SRTP packet payloads are not authenticated. The SRTP specification requires use of message authentication for SRTCP, but not for SRTP (RFC 3711). This is a negotiated parameter.

4.2.2.5 fecOrder

fecOrder signals the order of forward error correction processing for the RTP packets (RFC 3550, RFC 2733) relative to SRTP encryption at the sender. The value for **fecOrder** of **fecBeforeSrtp** signals that FEC is applied before SRTP processing by the sender of the SRTP media and after SRTP processing by the receiver of the SRTP media; **fecBeforeSrtp** is the default. **fecAfterSrtp** is the reverse order of processing. **fecOrder** is a declarative parameter.

4.2.2.6 windowSizeHint

SRTP defines the SRTP-WINDOW-SIZE (RFC 3711, section 3.3.2) parameter to protect against replay attacks. The minimum value is 64 (RFC 3711), however this value may be considered too low for some applications, e.g., video.

The Window Size Hint (WSH) session parameter provides a hint for how big this window should be to work satisfactorily (e.g., based on sender knowledge of number of packets per second). However, there might be enough information given in media packetization descriptors to allow a receiver to derive the parameter satisfactorily. Consequently, this value is only considered a hint to the receiver which may choose to ignore the value provided.

windowSizeHint is a declarative parameter.

4.2.2.7 Defining new SRTP session parameters

New SRTP session parameters are by default mandatory. The **newParameter** field is used as an extension mechanism for new session parameters. If an older H.323 Endpoint receives a **SrtpCryptoInfo** parameter with an unknown session parameter in the **newParameter** field, that new **SrtpCryptoInfo** parameter shall be considered invalid.

4.3 SrtpKeys parameter description

The field **SrtpKeys** contains one or more key parameters **SrtpKeyParameter** that are to be used for the SRTP session. Each **SrtpKeyParameter** contains the keying material (master key and salt) and all policy related to that master key, including how long it can be used (lifetime) and whether or not it uses a master key identifier (MKI) to associate an incoming SRTP packet with a particular master key. Compliant implementations obey the policies associated with a master key, and shall not accept incoming packets that violate the policy (e.g., after the master key lifetime has expired).

4.3.1 masterKey

This is the cryptographic master key to be used for the SRTP session. The length of the key is determined by the crypto suite for which the key applies. If the length does not match that specified for the crypto-suite, the crypto parameter in question shall be considered invalid. Each master key shall be a cryptographically random number and shall be unique to the proposed media flow.

4.3.2 masterSalt

This is the cryptographic master salt to be used for the SRTP session. The length of the salt is determined by the crypto suite for which the key applies. If the length does not match that specified for the crypto-suite, the crypto parameter in question shall be considered invalid. Each master salt shall be a cryptographically random number and shall be unique to the proposed media flow.

4.3.3 lifetime

This field is the optional lifetime of the master key as measured in the maximum number of SRTP or SRTCP packets using that master key (i.e., the number of SRTP packets and the number of SRTCP packets each have to be less than the lifetime). The lifetime value may be written as a non-zero, positive integer or as a power of 2. The "lifetime" value shall not exceed the maximum packet lifetime for the crypto-suite. If the lifetime is too large, or otherwise invalid, then the entire crypto parameter shall be considered invalid. If the lifetime field is not present, the default lifetime is to be used. This is convenient when the SRTP cryptographic key lifetime is the default value.

4.3.4 masterKeyId

This optional field specifies the policy of how keys are to be identified for the SRTP session. MKI is the master key identifier associated with the SRTP master key. If the MKI is given, then the length of the MKI shall also be provided. The MKI length is the size of the MKI field in the SRTP packet, specified in bytes. If the MKI length is not given or its value exceeds 128 (bytes), then the entire crypto parameter shall be considered invalid.

As mentioned above, the key parameter can contain one or more master keys. When the key parameter contains more than one master key, all of the master keys in that key parameter shall include an MKI value. When using the MKI, the MKI length shall be the same for all keys in a given crypto parameter.

4.4 SRTP crypto context initialization

In addition to the various SRTP parameters defined above, there are three pieces of information that are critical to the operation of the default SRTP ciphers:

- SSRC: Synchronization source
- ROC: Roll-over counter for a given SSRC
- SEQ: Sequence number for a given SSRC

In a unicast session, as defined here, there are three constraints on these values. The first constraint is on the SSRC, which makes an SRTP keystream be unique from other participants. As explained in SRTP, the keystream shall not be reused on two or more different pieces of plaintext.

Keystream reuse makes the ciphertext vulnerable to cryptanalysis. One vulnerability is that known-plaintext fields in one stream can expose portions of the reused keystream and this could further expose more plaintext in other streams. Since all current SRTP encryption transforms use keystreams, key sharing is a general problem (RFC 3711). SRTP mitigates this problem by including the SSRC of the sender in the keystream. But SRTP does not solve this problem in its entirety because Real-time Transport Protocol has SSRC collisions, which are very rare (RFC 3550) but quite possible. During a collision, two or more SSRCs that share a master key will have identical keystreams for overlapping portions of the RTP sequence-number space. The SRTP security description avoids keystream reuse by making unique master keys required for the sender and receiver of the security description. Thus, the first constraint is satisfied.

It should also be noted that there is a second problem with SSRC collisions: The SSRC is used to identify the crypto context and thereby the cipher, key, ROC, etc. to process incoming packets. In case of SSRC collisions, crypto context identification becomes ambiguous and correct packet

processing may not occur. Furthermore, if an RTCP BYE packet is to be sent for a colliding SSRC, that packet may also have to be secured.

The second constraint is that the ROC shall be zero at the time that each SSRC commences sending packets. Thus, there is no concept of a "late joiner" in SRTP security descriptions, which are constrained to be unicast and pairwise. The ROC and SEQ form a "packet index" in the default SRTP transforms and the ROC is consistently set to zero at session commencement, according to this Recommendation.

The third constraint is that the initial value of SEQ should be chosen to be within the range of 0..2^15 – 1; this avoids an ambiguity when packets are lost at the start of the session. If, at the start of a session, an SSRC source might randomly select a high sequence-number value and put the receiver in an ambiguous situation: if initial packets are lost in transit up to the point that the sequence number wraps (i.e., exceeds 2^16 – 1), then the receiver might not recognize that its ROC needs to be incremented. By restricting the initial SEQ to the range of 0..2^15 – 1, SRTP packet-index determination will find the correct ROC value, unless all of the first 2^15 packets are lost (which seems, if not impossible, then rather unlikely). See section 3.3.1 of the SRTP specification regarding packet-index determination (RFC 3771).

4.4.1 Late binding of SSRCs to a crypto context

The packet index, therefore, depends on the SSRC, the SEQ of an incoming packet and the ROC, which is an SRTP crypto context variable. Thus, SRTP has a big security dependency on SSRC uniqueness. Given the above constraints, unicast SRTP crypto contexts can be established without the need to negotiate SSRC values in the SRTP security description. Instead, an approach called "late binding" is recommended by this Recommendation. When a packet arrives, the SSRC that is contained in it can be bound to the crypto context at the time of session commencement (i.e., SRTP packet arrival) rather than at the time of session signalling (i.e., receipt of an H.245 message). With the arrival of the packet containing the SSRC, all the data items needed for the SRTP crypto context are held by the receiver (note that the ROC value by definition is zero; if non-zero values were to be supported, additional signalling would be required). In other words, the crypto context for a secure RTP session using late binding is initially identified by the H.245 message as:

<*, address, port>

where '*' is a wildcard SSRC, "address" is the local receive address from **mediaChannel**, and "port" is the local receive port from **portNumber**. When the first packet arrives with ssrcX in its SSRC field, the crypto context

<ssrcX, address, port>

is instantiated subject to the following constraints:

- Media packets are authenticated: Authentication must succeed; otherwise, the crypto context is not instantiated.
- Media packets are not authenticated: Crypto context is automatically instantiated.

It should be noted that use of late binding, when there is no authentication of the SRTP media packets, is subject to numerous security attacks and, consequently, it is not recommended (of course, this can be said for unauthenticated SRTP in general).

Note that use of late binding without authentication will result in local state being created as a result of receiving a packet from any unknown SSRC. Unauthenticated SRTP, therefore, is not recommended because it invites easy denial-of-service attack. In contrast, late binding with authentication does not suffer from this weakness.

4.4.2 Sharing of crypto contexts among sessions or SSRCs

With the constraints and procedures described above, it is not necessary to explicitly signal the SSRC, ROC and SEQ for a unicast RTP session. So there are no SRTP crypto-parameters for

signalling SSRC, ROC or SEQ. Thus, multiple SSRCs from the same entity will share SRTP crypto parameters when late binding is used. Multiple SSRCs from the same entity arises due to either multiple sources (microphones, cameras, etc.), or RTP payloads requiring SSRC multiplexing within that same session.

H.245 allows multiple RTP sessions to be defined in the same media description, these RTP sessions will also share the SRTP crypto parameters. An application that uses the SRTP crypto parameter in this way shares a master key among RTP sessions or SSRCs and shall replace the master key when the aggregate number of packets among all SSRCs approaches 2^31 packets. SSRCs that share a master key shall be unique from one another.

The lifetime of all keys that are derived from a master key are determined by the lifetime of the master key. So if the lifetime of the master key is 2^31 packets and one derived key has sent 2^31 – y packets, then only y packets can be sent by any key derived from that master key. This is because the lifetime is based on the amount of entropy or randomness in the key and no randomness is introduced by deriving a key from a master key, which is all the randomness or entropy that the key has.

4.4.3 Removal of crypto contexts

The mechanism defined above addresses the issue of creating crypto contexts. However, in practice, session participants may want to remove crypto contexts prior to session termination. Since a crypto context contains information that cannot automatically be recovered (e.g., ROC), it is important that the sender and receiver agree on when a crypto context can be removed and, perhaps more importantly, when it cannot.

Even when late binding is used for a unicast stream, the ROC is lost and cannot be recovered automatically (unless it is zero) once the crypto context is removed.

Crypto-contexts shall be removed on receipt of a **CloseLogicalChannel**. In addition, crypto-context removal shall follow the same rules as SSRC removal from the member table (RFC 3711); note that this can happen as the result of an SRTCP BYE packet or a simple time-out due to inactivity. Inactive session participants that wish to ensure their crypto contexts are not timed out must thus send SRTCP packets at regular intervals.

5 Procedures

The SRTP procedures described below shall only be used to negotiate security for two-party unicast media streams in situations where the H.245 signalling channel is protected by an encapsulating data-security protocol, e.g., IPsec (RFC 2401), TLS (RFC 2246). The exchange of SRTP crypto parameters using H.245 messages shall provide the following functions:

1) Exchange and negotiation of SRTP media encryption and integrity capabilities.
2) Negotiation and establishment of the initial encryption and algorithms, keys and session parameters to be used for the SRTP streams in each direction.
3) Modification of encryption and algorithms, keys and session parameters at any time during the SRTP session.

5.1 Security capability exchange

The SRTP crypto-suites, the encryption and integrity algorithms, that an H.323 Endpoint is capable of supporting shall be identified by **SrtpCryptoCapability**.

Security capabilities exchange will be provided by existing terminal capabilities exchange using one or more **h235SecurityCapability** entries in the **capabilityTable** of the H.245 **TerminalCapabilitySet** message. The **mediaCapability** field in the **h235SecurityCapability** entry

of the **capabilityTable** is used to associate the security capability with a particular media capability entry in the **capabilityTable**.

The **encryptionAuthenticationAndIntegrity** field in the **h235SecurityCapability** entry contains the **genericH235SecurityCapability** field which will specify the SRTP crypto-suites identified by the H.235.8 OIDs. If the **standard** field of **capabilityIdentifier** of the **genericH235SecurityCapability** field contains the H.235.8 OID (see Table 1) then the **SrtpCryptoCapability** will contain one or more **SrtpCryptoInfo** parameters that represent the crypto-suites that the H.323 Endpoint supports. The **cryptoSuite** field in the **SrtpCryptoInfo** field contains an OID as defined in Table 2 that identifies a particular crypto-suite. Within **SrtpCryptoInfo** field the **sessionParams** field identifies the session parameters and the **allowMKI** field indicates whether the MKI is supported by the H.323 Endpoint.

5.2 Initial negotiation

5.2.1 Initial crypto offer

Each crypto-offer is transported in a separate **OpenLogicalChannel** message. Each crypto-offer shall contain one **SrtpCryptoInfo** structure in **SrtpCryptoCapability** and one or more **SrtpKeyParameters** structures in **SrtpKeys**.

For normal H.245 (not Fast Connect) procedures the H.323 Endpoint shall include the crypto-offer as described in **SrtpCryptoInfo** and **SrtpKeyParameters** structures in an H.245 **OpenLogicalChannel** message for the forward direction (from the offering H.323 Endpoint to the answering H.323 Endpoint). The H.323 Endpoint should offer the master's most preferred security capability as indicated during the terminal capabilities exchange and for which it itself has capability.

For Fast Connect procedures, the offering H.323 Endpoint shall send each crypto-offer as described in **SrtpCryptoInfo** and **SrtpKeyParameters** structures in separate H.245 **OpenLogicalChannel** messages for the forward direction (from the offering H.323 Endpoint to the answering H.323 Endpoint).

The offered **OpenLogicalChannel** messages shall be ordered by preference, the most preferred crypto-suite being listed first, the more preferred crypto-suites should be cryptographically stronger than less preferred crypto-suites. In general, a more preferred crypto-suite should be cryptographically stronger than a less preferred crypto-suite.

When issuing a crypto-offer, the offerer shall be prepared to support media security in accordance with any of the offered crypto-parameters. There are two problems associated with this. Firstly, the offerer does not know which key the answerer will be using for media sent to the offerer. Since media may arrive prior to the crypto-answer, delay or clipping can occur. If this is unacceptable to the offerer, the offerer should use a mechanism such as H.460.11 delayed call establishment procedures to prevent the above problem.

Another problem can occur when there are multiple offers: The offerer is not able to deduce which of the offers was accepted by the answerer until the crypto-answer is received, yet media may arrive before the crypto-answer. If this is unacceptable to the offerer, the offerer either should not send more than one offer, or a mechanism such as H.460.11 delayed call establishment procedures should be used to prevent the above problem.

The **SrtpCryptoInfo** may include session parameters.

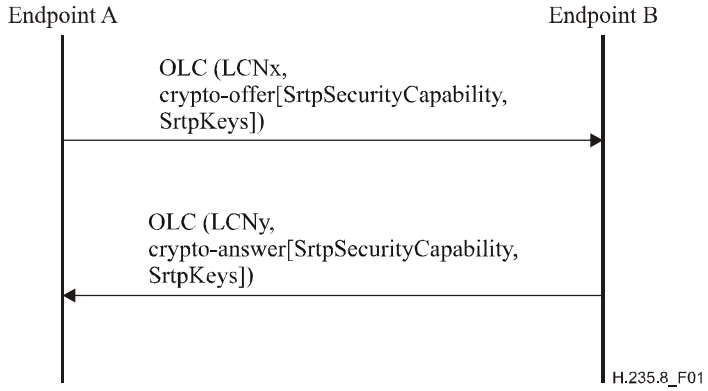

Figure 1/H.235.8 – Fast connect offer-answer exchange

5.2.1.1 Initial crypto-answer

5.2.1.1.1 General

These procedures apply to both Fast Connect and normal H.245 procedures. A crypto-answer shall contain one **SrtpCryptoInfo** structure in **SrtpCryptoCapability** and one or more **SrtpKeyParameters** structures in **SrtpKeys**.

The answering H.323 Endpoint shall apply the crypto-suite selected from the crypto-offer to the corresponding unidirectional SRTP channel in the reverse direction and shall generate the key(s) to be used for that SRTP channel in the reverse direction.

In addition, the answering H.323 Endpoint shall include one or more keys in **SrtpKeys** that are to be used for the SRTP stream from the answering H.323 Endpoint to the offering H.323 Endpoint. The answering H.323 Endpoint may also include any session parameters from the crypto-offer that it wishes to negotiate.

Only parameters that are valid can be accepted; valid parameters do not violate any of the general rules defined for security descriptions as well as any specific rules defined for the transport and key-method in question.

For Fast Connect, when selecting one of the valid crypto-offers, the answerer should select the most preferred crypto-offer it can support, i.e., the first valid supported parameter in the list, considering the answerer's capabilities and security policies. If none of the offers are valid, or none of the valid ones are supported, the offered media stream shall be rejected.

When a crypto-offer is accepted, the crypto-answer shall contain the key(s) the answerer will be using for media sent to the offerer. Note that a key shall be provided, irrespective of any direction parameters in the offer or answer.

Furthermore, any session parameters that are negotiated shall be included in the crypto-answer. Declarative session parameters provided by the offerer are not included in the crypto-answer, however the answerer may provide its own set of declarative session parameters.

Once the answerer has accepted one of the offered crypto parameters, the answerer may begin sending media to the offerer in accordance with the selected crypto-offer. Note however, that the offerer may not be able to process such media packets correctly until the crypto-answer has been received.

5.2.1.1.2 Fast connect procedures

For Fast connect procedures, the answering H.323 Endpoint receiving the crypto-offers in one or more H.245 **OpenLogicalChannel** messages shall respond by accepting one of the crypto-offers by sending an H.245 **OpenLogicalChannel** containing the crypto-answer as shown in Figure 1, or by

rejecting all the crypto-offers by sending a **ReleaseComplete** with **ReleaseCompleteReason** set to **securityDenied**, or by sending a **FastConnectRefused** element in an H.225.0 message. If the answering H.323 Endpoint does not support this Recommendation or any of the proposals in the crypto-offer it shall reject the crypto-offer by sending a **ReleaseComplete** with **ReleaseCompleteReason** set to **securityDenied**, or by sending a **FastConnectRefused** element in an H.225.0 message.

5.2.1.1.3 Normal H.245 procedures

For normal H.245 (not Fast Connect) procedures, the following procedure applies. If the H.323 Endpoint has not already sent an **OpenLogicalChannel** containing a crypto-offer prior to the receipt of **OpenLogicalChannel** containing a crypto-offer it shall send an **OpenLogicalChannelAck** followed by an **OpenLogicalChannel** containing the crypto-answer as shown in Figure 2.

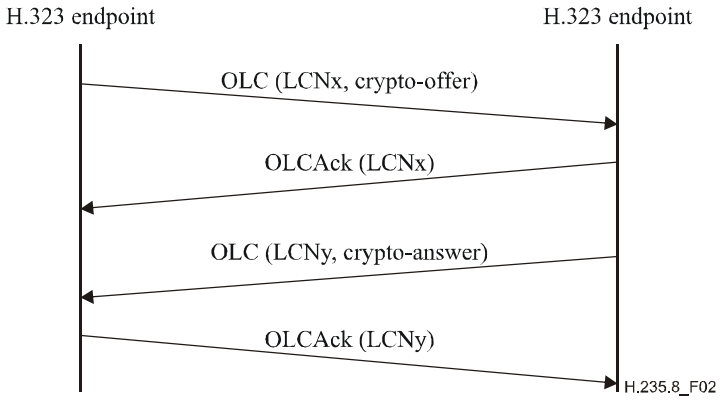

Figure 2/H.235.8 – Offer-answer exchange

If the H.323 Endpoint has already sent an **OpenLogicalChannel** containing a crypto-offer prior to the receipt of **OpenLogicalChannel** containing a crypto-offer the action taken by the master and slave H.323 Endpoints is as follows:

1) A master H.323 Endpoint shall process the received crypto-offer and if it is compatible with the crypto-offer that it has already sent, it shall accept the received crypto-offer as a crypto-answer by sending an **OpenLogicalChannelAck** as shown in Figure 3. If the received crypto-offer is not compatible with the crypto-offer that it has already sent, it shall reject the received crypto-offer by sending an **OpenLogicalChannelReject** with **cause** value of **securityDenied** as shown in Figure 4. The term compatible means that the following parameters in the crypto-offer shall match the corresponding parameters in the crypto-answer: **cryptoSuite** and the negotiated session parameters.

2) A slave H.323 Endpoint shall process the received crypto-offer and, if it is compatible with the crypto-offer that it has already sent, it shall accept the received crypto-offer as a crypto-answer by sending an **OpenLogicalChannelAck** as shown in Figure 3. If the received crypto-offer is not compatible with the crypto-offer that it has already sent, and if it wishes to accept the crypto-offer, it shall do so by sending the following messages shown in Figure 4:

   a) **OpenLogicalChannelAck** to accept the initial crypto-offer from the master;

   b) **CloseLogicalChannel** to terminate its own initial crypto-offer if the **OpenLogicalChannelReject** has not already been received from the master;

   c) **OpenLogicalChannel** with a crypto-answer that matches the crypto-offer from the master.

If the slave H.323 Endpoint does not support the proposal in the offer or does not wish to accept the crypto-offer it shall reject the crypto-offer by sending an **OpenLogicalChannelReject** with **cause** set to **securityDenied**.

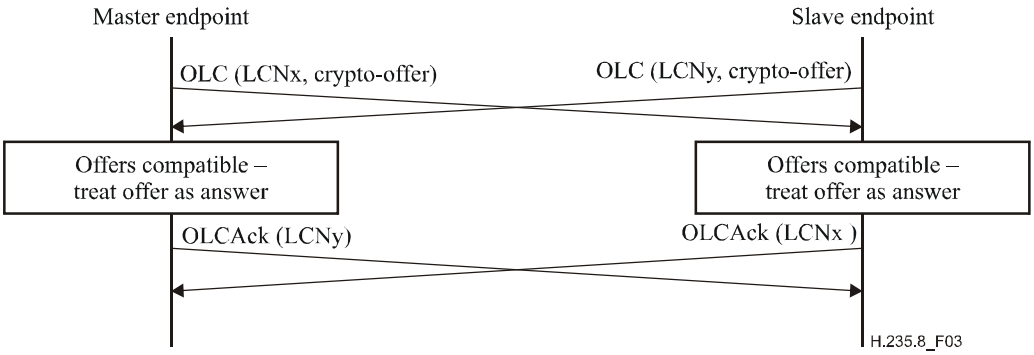

Figure 3/H.235.8 – Simultaneous compatible offer-answer exchange

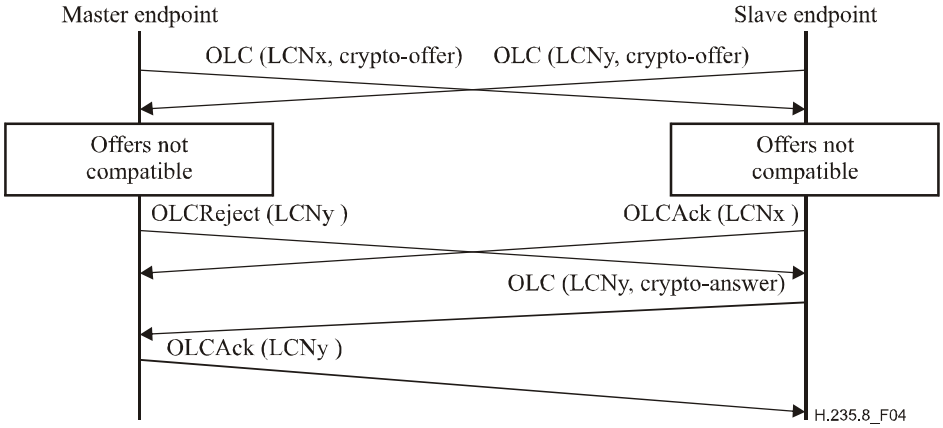

Figure 4/H.235.8 – Simultaneous incompatible offer-answer exchange

5.2.1.2 Offerer processing of initial answer

When the offerer receives the crypto-answer, the offerer shall verify that one of the initial crypto-offers was accepted and echoed in the crypto-answer. Also, the crypto-answer shall include one or more keys, which will be used for media sent from the answerer to the offerer.

The offerer shall verify that the keys in the crypto-answer do not match any of the keys in the crypto-offer. If the crypto-offer contained any mandatory negotiated session parameters, the offerer shall verify that said parameters are included in the crypto-answer and match the corresponding parameters in the crypto-offer. If the crypto-answer contains any mandatory declarative session parameters, the offerer shall be able to support those.

If any of the above fails, the negotiation shall be deemed to have failed.

5.3 Session modification

Once the SRTP media stream has been established, it may be modified at any time using new offer-answer exchanges to perform re-keying or change the crypto-suite. The new crypto-offer and crypto-answer shall be transported in **SrtpCryptoCapability** and **SrtpKeys** parameters of an H.245 **OpenLogicalChannel** to open a new logical channel that will replace the existing logical channel using **replacementFor** procedures. The offering H.323 Endpoint shall include the crypto-offers in one or more H.245 **OpenLogicalChannel** messages within an H.225.0 message.

The answering H.323 Endpoint receiving the crypto-offers shall respond by accepting one of the offers by sending an H.245 **OpenLogicalChannel** within an H.225.0 message or by rejecting the offers with an **OpenLogicalChannelReject** message with **cause** set to **securityDenied**. If the crypto-offer is rejected, the old crypto parameters remain in place.

When establishing a new master key, there will be a window of time during which the H.323 Endpoint must receive media encrypted according to the old and the new offer-answer exchange. The MKI from the incoming SRTP packet shall be used to associate that packet with either the old master key or the new master key. For this reason, if it is anticipated that keys will be changed during a session that does not change the source/destination addresses and ports, the use of MKI is mandatory to allow the receiver to identify the associated key material during the key change.

5.4 No negotiation

In the case of no negotiation of the crypto suite, cryptographic key or session parameters the sender determines the security parameters for the stream. Since there is no negotiation mechanism, the sender shall include exactly one crypto-offer and the receiver shall either accept it or else should reject the offer by sending a **ReleaseComplete** with **ReleaseCompleteReason** set to **securityDenied** or an **OpenLogicalChannelReject** with **cause** set to **securityDenied**. The sender should select the security description that it deems most secure for its purposes.

5.5 Forward error correction

A different master key shall be specified to protect a FEC stream that is sent to a different IP address and/or port pair than the SRTP media stream to which it applies as described in RFC 2733, section 11.1. This FEC stream shall be established using a separate H.245 **OpenLogicalChannel** with **dataType** of **fec**. The master key for the FEC stream shall be transported in the **genericKeyMaterial** field of the **secureSharedSecret** (**V3KeySyncMaterial**) parameter contained within the **h235Key** container in the **encryptionSync** parameter of the H.245 **OpenLogicalChannel** message. The master key shall be different from all other master keys offered for the associated media stream.

6 Public key cryptography to secure key exchange for SRTP

Public key cryptography procedures may be added to provide end-to-end confidentiality and authentication of the SRTP session key material exchanged between H.323 Endpoints by encrypting and then signing the SRTP key material. Public key cryptography may be used in the case where the encapsulating security protocol, e.g., IPsec, TLS, terminates on an intermediate device and, therefore, does not provide end-to-end security.

The SRTP session key that encrypts the SRTP media from the calling Endpoint to the called Endpoint is encrypted using the public key of the called Endpoint and signed with the private key of the calling Endpoint. Likewise, another SRTP session key that encrypts the SRTP media from the called Endpoint to the calling Endpoint shall be encrypted using the public key of the calling Endpoint and signed with the private key of the called Endpoint. The procedure described in this clause may terminate on a Gateway or Gatekeeper as well as an Endpoint.

The SRTP session key shall be transported using Cryptographic Message Syntax (CMS) bodies within H.245 messages. The Cryptographic Message Syntax (RFC 3852) is used to digitally sign and encrypt arbitrary message content. The CMS syntax allows multiple encapsulations which lets one encapsulation envelope be nested within another. In particular, the SRTP session key material shall be transported within a CMS **EnvelopedData** body which is signed using a CMS **SignedData** body.

6.1 Endpoint identification

The following shall be used to identify an Endpoint, Gateway or Gatekeeper in a public key certificate:

- H.323 URL;
- Non-H.323 standard URL, e.g., *tel*;
- Device identification/certificate (FFS).

A public key certificate shall be used to assert the association of the identity of the Endpoint with its public key. The H.323 URL or non-H.323 standard URL shall be stored in the **subjectAltName** field of the certificate.

Endpoints may maintain a local key-store that contains the public key certificates of other Endpoints with which it wishes to establish secure end-to-end communications. An Endpoint that sends signed content to provide end-to-end authentication shall include a public key certificate bearing the public key necessary to verify the signature. A receiving Endpoint shall either:

a) verify that sender's certificate is signed by a recognized certification authority (CA); or

b) trust a security assertion upon the certificate given by a third party. The assertion must be signed by globally verifiable key material.

NOTE – This may be advantageous in scenarios where a global user PKI is not available and self-signed certificates or device certificates are being used.

6.2 SRTP key exchange procedures

If the calling and called Endpoints wish to ensure end-to-end confidentiality and authentication of their SRTP session key material in the case that the call establishment traverses one or more intermediate signalling devices, they should use public key cryptography and X.509 (RFC 3280) public key certificate exchange.

The offer-answer procedures described in the previous clauses are unchanged except as stated below.

6.2.1 Capability exchange

To negotiate the use of public key certificates for SRTP key exchange, the H.323 Endpoint shall set the **genericH235SecurityCapability** within **encryptionAuthenticationAndIntegrity** in an **h235SecurityCapability** entry in the **capabilityTable** of a H.245 **TerminalCapabilitySet** message as follows:

- **capabilityIdentifier** shall contain the H.235.8 CMS Object Identifier (see Table 4) in **standard** field;
- **maxbitRate, collapsing, nonCollapsing**, and **transport** shall be unused;
- **nonCollapsingRaw** shall contain the **SrtpCryptoCapability** parameter.

6.2.2 Key exchange

If the SRTP session key is to be encrypted using public keys, then the encrypted SRTP session key is transported within Cryptographic Message Syntax (CMS) bodies in H.245 messages. The CMS **EnvelopedData** body and the CMS **SignedData** body shall be transported instead of **SrtpKeys** in the **genericKeyMaterial** field of the **secureSharedSecret** (**V3KeySyncMaterial**) parameter contained within the **h235Key** container in the **encryptionSync** parameter of H.245 **OpenLogicalChannel** messages. The CMS **EnvelopedData** body shall be placed into **genericKeyMaterial** field immediately followed by the CMS **SignedData** body.

The structure **SrtpKeys** shall be encrypted using the CMS Content Encryption Key (CEK) and transported in the **EncryptedContentInfo** structure of a CMS **EnvelopedData** body.

The presence of a CMS body containing SRTP session key material in the **genericKeyMaterial** container shall be identified using the H.235.8 CMS Object Identifier value (see Table 4) in the **standard** field of **capabilityIdentifier** within the **genericH235SecurityCapability** field of **encryptionAuthenticationAndIntegrity** in **h235Media** of the OLC **dataType**.

Table 4/H.235.8 – H.235.8 CMS object identifier

OID value
{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 94 }

6.3 CMS body usage

The Endpoint that generates the SRTP session key material **SrtpKeys**, the sending Endpoint, shall encrypt it using the CMS Content Encryption Key (CEK) which is itself encrypted by the public key of the other Endpoint, the recipient Endpoint, and place the encrypted SRTP session key material into a CMS **EnvelopedData** body. The sending Endpoint shall then digitally sign the **EnvelopedData** body with its private key and create a "detached signature" CMS **SignedData** body. The sending Endpoint shall include the certificate with its public key in the CMS **SignedData** body. The sending Endpoint shall send the **EnvelopedData** body with the "detached signature" **SignedData** body to the recipient Endpoint. The creation of the **EnvelopedData** and the **SignedData** bodies by the sending Endpoint are described in more detail in the following clauses.

The **EnvelopedData** body and the "detached signature" **SignedData** body are shown in Figure 5.

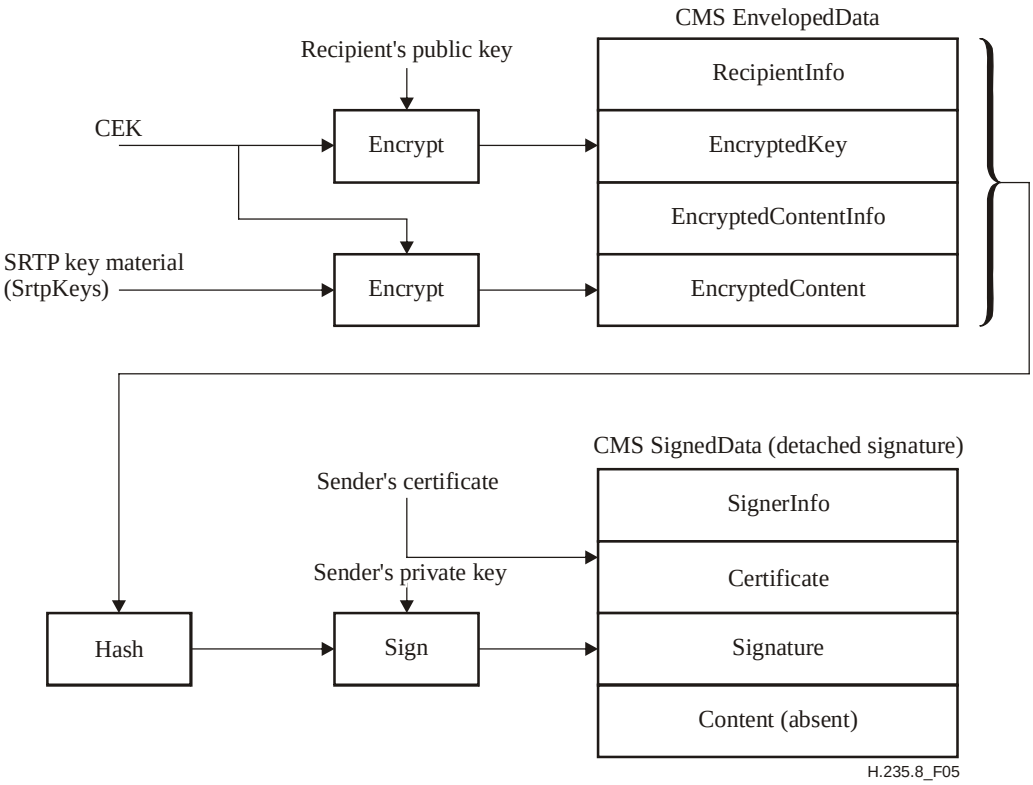

Figure 5/H.235.8 – CMS EnvelopedData and SignedData bodies

6.3.1 Sending endpoint procedures

The sending Endpoint shall perform the following procedures to generate, encrypt and sign the SRTP session key material.

6.3.1.1 EnvelopedData body

The sending Endpoint shall construct the **EnvelopedData** body as follows:

1) Generate the SRTP session key material **SrtpKeys** for the crypto-suite.

2) Generate a random Content Encryption Key (CEK).

3) Encrypt the CEK using the public key of the recipient Endpoint. It is assumed that sending Endpoint already has the public key and certificate of the recipient Endpoint. Place the identifier of the algorithm used in the encryption of the CEK in the **keyEncryptionAlgorithm** field of the **RecipientInfo.ktri** structure.

4) Place the encrypted CEK in the **encryptedKey** field of a **RecipientInfo** structure of an **EnvelopedData** body. The **rid** field of the **RecipientInfo.ktri** structure is used to identify the certificate and public key of the recipient Endpoint that was used to encrypt the CEK.

5) Encrypt the SRTP key material **SrtpKeys** using the CEK and place the identifier of the algorithm used in the encryption in the **contentEncryptionAlgorithm** field of the **EncryptedContentInfo** structure.

6) Place the encrypted SRTP key material in the **encryptedContent** field of the **EncryptedContentInfo** structure.

6.3.1.2 SignedData body

The sending Endpoint shall construct the "detached signature" **SignedData** body as follows:

1) Compute a message digest, or hash value, over the **EnvelopedData** body. The message digest algorithm identifier is placed in the **digestAlgorithm** field of the **SignerInfo** structure.

2) Sign the message digest using the sending Endpoint's private key and place the signature value in the **signature** field of the **SignerInfo** structure. The signature algorithm identifier is placed in the **signatureAlgorithm** field of the **SignerInfo** structure.

3) Place the certificate containing the public key of the sending Endpoint in the **certificates** structure of the **SignerData** structure. The **sid** field of the **SignerInfo** structure shall be set to identify the certificate using either the issuer's distinguished name and certificate serial number, or the X.509 subjectKeyIdentifier extension value.

4) The **eContentType** field of the **encapContentInfo** structure in the **SignedData** body shall contain the Object Identifier id-envelopedData. The **eContent** field of the **encapContentInfo** structure in the **SignedData** body shall be absent as this is a detached signature and the actual signed content is the **EnvelopedData** body.

6.3.2 Recipient endpoint procedures

The recipient Endpoint shall perform the following procedures to verify and decrypt the SRTP session key material.

If the recipient Endpoint encounters any validation failure in the procedures described below, the call is rejected by sending a **ReleaseComplete** with **ReleaseCompleteReason** set to **securityDenied**, or by sending a **FastConnectRefused** element in an H.225.0 message.

6.3.2.1 SignedData body

The recipient Endpoint shall verify the received "detached signature" **SignedData** body as follows:

1) Obtain the sending Endpoint's certificate from the **certificates** structure of the **SignerData** structure.
2) Validate the sending Endpoint's certificate. Details of certificate path validation are beyond the scope of this Recommendation. If the recipient is unable to authenticate the sending Endpoint, it may reject the call.
3) The recipient Endpoint may then add the validated certificate to its key-store.
4) Verify the signature value in the **signature** field of the **SignerInfo** structure using the sending Endpoint's public key from the validated certificate. Use the signature algorithm specified in the **signatureAlgorithm** field of the **SignerInfo** structure. The result of the decryption is the message digest over the **EnvelopedData** body computed by the sending Endpoint.
5) Compute the message digest over the received **EnvelopedData** body using the message digest algorithm identifier specified in the **digestAlgorithm** field of the **SignerInfo** structure.
6) Compare the decrypted message digest value with the computed message digest value. If the message digests match, the **EnvelopedData** body may then be processed. If the message digests do not match, the recipient endpoint shall reject the call.

6.3.2.2 EnvelopedData body

The recipient Endpoint shall extract the SRTP session key material from the **EnvelopedData** body as follows:

1) Use the **rid** field of the **RecipientInfo** structure to identify the certificate and corresponding private key of the recipient Endpoint in the recipient Endpoint's key-store. If the recipient Endpoint receives an **EnvelopedData** body that is encrypted with a public key that is unknown to the recipient, it shall reject the call.
2) Extract the encrypted CEK from the **encryptedKey** field of a **RecipientInfo.ktri** structure of an **EnvelopedData** body.
3) Decrypt the encrypted CEK using the private key of the recipient Endpoint and the algorithm specified in the **keyEncryptionAlgorithm** field of the **RecipientInfo.ktri** structure.
4) Extract the encrypted SRTP session key material from the encrypted SRTP session key material in the **encryptedContent** field of the **EncryptedContentInfo** structure.
5) Decrypt the encrypted SRTP session key material using the CEK and the algorithm specified in the **contentEncryptionAlgorithm** field of the **EncryptedContentInfo** structure.

7 H.235 SRTP security descriptions syntax

The ASN.1 syntax is defined below.

```
H235-SRTP DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
     GenericData
FROM H323-MESSAGES;


 SrtpCryptoCapability ::= SEQUENCE OF SrtpCryptoInfo -- used in H.245
 genericH235SecurityCapability


 SrtpCryptoInfo ::= SEQUENCE
 {
    cryptoSuite                     OBJECT IDENTIFIER OPTIONAL ,
    sessionParams                   SrtpSessionParameters OPTIONAL,
    allowMKI                        BOOLEAN OPTIONAL,
    ...
 }

 SrtpKeys ::= SEQUENCE OF SrtpKeyParameters  -- used in H.235 V3KeySyncMaterial

 SrtpKeyParameters ::= SEQUENCE
 {
    masterKey                       OCTET STRING,
    masterSalt                      OCTET STRING,
    lifetime                        CHOICE
    {
       powerOfTwo                   INTEGER,
       specific                     INTEGER,
       …
    } OPTIONAL,
    mki                             SEQUENCE
    {
       length                       INTEGER(1..128),
       value                        OCTET STRING,

     ...
    } OPTIONAL,
    ...
 }


 SrtpSessionParameters ::= SEQUENCE
 {
    kdr                             INTEGER(0..24) OPTIONAL, -- power of 2
    unencryptedSrtp                 BOOLEAN OPTIONAL,
    unencryptedSrtcp                BOOLEAN OPTIONAL,
    unauthenticatedSrtp             BOOLEAN OPTIONAL,
    fecOrder                        FecOrder OPTIONAL,
    windowSizeHint                  INTEGER(64..65535) OPTIONAL,
    newParameter                    SEQUENCE OF GenericData  OPTIONAL,
    ...
 }

 FecOrder ::= SEQUENCE
 {
    fecBeforeSrtp                   NULL OPTIONAL,
    fecAfterSrtp                    NULL OPTIONAL,
    ...
 }

END
```

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2006