Internet Engineering Task Force
MMUSIC Working Group
INTERNET-DRAFT
EXPIRES: April 2004

Flemming Andreasen
Mark Baugher
Dan Wing
Cisco Systems
October 24, 2003

Session Description Protocol Security Descriptions for Media Streams

<draft-ietf-mmusic-sdescriptions-02.txt>

Status of this memo

Copyright Notice

Abstract

This document defines a Session Description Protocol (SDP) cryptographic attribute for media streams. The attribute describes a cryptographic key and other parameters, which serve to configure security for a media stream in either a single message or a roundtrip. The attribute can be used with a variety of SDP media transports and this document defines how to use it for the Secure Real-time Transport Protocol (SRTP) media streams. The SDP crypto attribute requires the services of a data security protocol to secure the SDP message.

Table of Contents

1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. The terminology in this document conforms to [RFC2828], "Internet Security Glossary".

n^r is exponentiation where n is multiplied by itself r times; n and r are integers. 0..k is an integer range of all integers from 0 through k inclusive. The abbreviation "iff" means "if and only if."

2. Introduction

The Session Description Protocol (SDP) describes multimedia sessions, which can be audio, video, whiteboard, fax, modem, and other media sessions. Security services such as data origin authentication, integrity and confidentiality are often needed for media streams. The Secure Real-time Transport Protocol (SRTP) [srtp] provides such security services and is signaled by use of the "RTP/SAVP" transport in an SDP media (m=) line. However, there are no means within SDP itself to configure SRTP beyond using default values. This document specifies a new SDP attribute called "crypto", which is used to signal and negotiate cryptographic parameters for media streams in general, and SRTP in particular.

The crypto attribute is defined in a generic way to enable its use with secure transports besides SRTP that need to signal and negotiate cryptographic parameters, e.g. IPsec [ipsec], S/MIME [s/mime], or TLS [tls], if and only if such parameters can either be advertised in a single message, or negotiated in a single round-trip by use of the offer/answer model [RFC3264]. Such extensions, however, are beyond the scope of this document. Each type of secure SDP media transport needs its own specification for the crypto-attribute parameter. These definitions are frequently unique to the particular type of transport and MUST be specified in an Internet RFC and registered with IANA according to the procedures defined in Section 10. This document defines the security parameters and keying material for SRTP only.

It would be self-defeating not to secure cryptographic keys and other parameters at least as well as SRTP secures RTP packets or

IPsec secures IP packets. Data security protocols such as SRTP rely upon a separate key management system to securely establish encryption and/or authentication keys. Key management protocols

provide authenticated key establishment (AKE) procedures to authenticate the identity of each endpoint and protect against man-in-the-middle, reflection/replay, connection hijacking and some denial of service attacks [skeme]. Along with the key, an AKE protocol such as MIKEY [mikey], GDOI [GDOI], KINK [kink], IKE [ike] or TLS [tls] securely disseminates information describing both the key and the data-security session (for example, whether SRTCP payloads are encrypted or unencrypted in an SRTP session). AKE is needed because it is pointless to provide a key over a medium where an attacker can snoop the key, alter the definition of the key to render it useless, or change the parameters of the security session to gain unauthorized access to session-related information.

SDP, however, was not designed to provide AKE services, and the media security descriptions that follow do not add AKE services to SDP. This specification is no replacement for a key management protocol or for the conveyance of key management messages in SDP [keymgt]. The SDP security descriptions defined here are suitable for restricted cases only where IPsec, TLS, or some other encapsulating data-security protocol (e.g. SIP secure multiparts) protects the SDP message. This document adds security descriptions to those encrypted and/or authenticated SDP messages through the "crypto" attribute, which provides the cryptographic parameters of a media stream. The "crypto" attribute can be adapted to any media transport, but its precise definition is frequently unique to a particular transport. In Section 3, we introduce the general SDP crypto attribute, and in Section 4 we define how it is used with and without the offer/answer model. In Section 5, we define the crypto attribute details needed for SRTP, and in Section 6 we define SRTP-specific use of the attribute with and without the offer/answer model. Section 7 recites security considerations, and Section 8 gives an Augmented-BNF grammar for the general crypto attribute as well as the SRTP-specific use of the crypto attribute. A list of open issues is provided in Section 9 and IANA considerations are provided in Section 10.

3. SDP "Crypto" Attribute and Parameters

A new media-level SDP attribute called "crypto" describes the cryptographic suite, key parameters, and session parameters for the preceding media line. The "crypto" attribute MUST only appear at the SDP media level (not the session level). The "crypto" attribute follows the format (see Section 8.1 for a formal ABNF grammar):

```
a=crypto:<crypto-suite> <key-params> *<session-params>
```

The fields crypto-suite, key-params, and session-param are described in the following sub-sections. Below we show an example of the

crypto attribute for the "RTP/SAVP" transport, i.e. SRTP (newlines included for formatting reasons only):

```
a=crypto:AES_CM_128_HMAC_SHA1_80
 inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:32
 SRC=/721/13
```

The crypto-suite is AES_CM_128_HMAC_SHA1_80, key-params is defined by the line starting with "inline:", and there is a single session-param named "SRC".

3.1 Crypto-suite

The crypto-suite field is an identifier (see Section 8.1 for details) that describes the encryption and authentication algorithms (e.g. AES_CM_128_HMAC_SHA1_80) for the transport in question. The possible values for the crypto-suite parameter are defined within the context of the transport, i.e. each transport defines a separate namespace for the set of crypto-suites. For example, the crypto-suite "AES_CM_128_HMAC_SHA1_80" defined within the context of the "RTP/SAVP" transport applies to Secure RTP only; the string may be reused for another transport, however a separate definition would be needed.

3.2 Key Parameters

The key-params field provides one or more sets of keying material for the crypto-suite in question. The field consists of a method indicator followed by a colon, and the actual keying information as shown below (a formal grammar is provided in Section 8.1):

```
key-params = <key-method> ":" <key-info>
```

Keying material may be provided by different means. One method is defined in this document, namely "inline", which indicates that the keying material is provided in the key-info field itself. There is a single name space for the key-method, i.e. the key-method is transport independent. New key-methods (e.g. use of a URL) may be defined in an IETF RFC in the future, in which case they may be used with any transport, provided the definitions for that transport support use of the new key-method. New key methods MUST be registered with the IANA according to the procedures defined in Section 10.2.1.

Key-info is here just defined as a general character string (see Section 8.1 for details); further transport and key-method specific syntax and semantics MUST be provided in an IETF RFC for each combination of transport and key-method that wants to use it; definitions for SRTP are provided in Section 5. Note that such definitions are provided within the context of both a particular transport (e.g. "RTP/SAVP") and a specific key-method (e.g. "inline").

When multiple keys are included in the key parameters, it MUST be possible to determine which key is being used by a simple inspection of the media packet received; a trial-and-error approach between the possible keys MUST NOT be required.

For SRTP, this could for example be achieved by use of Master Key Identifiers (MKI), or <"From", "To"> values.

3.3 Session Parameters

Session parameters are specific to a given transport and use of them is OPTIONAL in the general framework, where they are just defined as a general character string. If session parameters are to be used for a given transport, then key-method and transport-specific syntax and semantics MUST be provided in an IETF RFC for each transport that wants to use it; definitions for SRTP are provided in Section 5. Note that such definitions are provided within the context of both a specific key-method (e.g. "inline") and a particular transport (e.g. "RTP/SAVP").

3.4 Example

The first example shows use of the crypto attribute for the RTP/SAVP media transport type (as defined in Section 4). The a=crypto line is actually one long line, although it is shown as two lines in this document due to page formatting.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=video 51372 RTP/SAVP 31
a=crypto:AES_CM_128_HMAC_SHA1_80
 inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:AES_CM_128_HMAC_SHA1_32
 inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait
```

This SDP message describes three media streams, two of which use the RTP/SAVP transport. Each has a crypto attribute for the RTP/SAVP transport. These RTP/SAVP-specific descriptions are defined in the Section 5.

4. General Use of the crypto Attribute

In this section, we describe the general use of the crypto attribute outside of any transport or key-method specific rules.

4.1 Use With Offer/Answer

In this section, we define the general rules for use of the crypto attribute with the offer/answer [RFC3264] model. These rules are in addition to the rules specified in RFC 3264, which MUST be followed, unless otherwise noted.

4.1.1 Generating the Initial Offer

4.1.1.1 Unicast Streams

When generating an initial offer for a unicast stream, there MUST be one or more crypto attributes present for each media stream for which security is desired. The ordering of multiple "a=crypto" lines is significant: The most-preferred crypto line is listed first. Each crypto attribute describes the crypto-suite, key(s) and possibly session parameters offered for the media stream. In general, a "more preferred" crypto suite SHOULD be stronger cryptographically than a "less preferred" crypto suite.

The crypto-suite always applies to media in all directions supported by the media stream (e.g. send and receive).

The key(s) apply to media in the direction from the offerer to the answerer; if the media stream is marked as "recvonly", a key MUST still be provided.

> This is done for consistency. Also, in the case of for example SRTP, secure RTCP will still be flowing in both the send and receive direction for a unidirectional stream.

There are no general offer/answer rules for the session parameters; instead, specific rules are provided as part of the transport and key-method specific definitions of any session parameters.

When issuing an offer, the offerer MUST be prepared to support media security in accordance with any of the crypto attributes included in the offer. There are however two problems associated with this. First of all, the offerer does not know which key the answerer will be using for media sent to the offerer; the answerer may or may not choose the same key as the offerer chose in his sending direction (in fact, the answerer SHOULD NOT use the same key as explained in Section 4.1.2.1). Since media may arrive prior to the answer, delay or clipping may occur. If this is unacceptable to the offerer, the offerer SHOULD use a mechanism outside the scope of this document to prevent the above problem.

For example, a "security" precondition [RFC3312] could be defined to solve the above problem.

Another problem can occur when the offerer includes multiple crypto attributes, since the offerer may not be able to deduce which of the offered crypto attributes was accepted by the answerer until the answer is received, yet media may arrive before the answer.

If this is unacceptable to the offerer, the offerer either SHOULD NOT include multiple crypto attributes in the offer, or a mechanism outside the scope of this document SHOULD be used to prevent the above problem (e.g. a "security" precondition).

4.1.1.2 Multicast Streams

The rules for multicast streams are similar to those for unicast streams, except as noted below:

* In order to ensure that all participants use the same crypto parameters, there MUST be exactly one crypto attribute per media stream.

* The key(s) provided apply to media in all directions supported by the media stream, as opposed to just the sending direction.

4.1.2 Generating the Initial Answer

4.1.2.1 Unicast Streams

When the answerer receives the initial offer with one or more crypto attributes for a given unicast media stream, the answerer MUST either accept exactly one of the offered crypto attributes, or the offered stream MUST be rejected.

If the answerer wishes to indicate support for other crypto attributes, those can be listed by use of the SDP Simple Capability Declaration [RFC3407] extensions.

Only crypto attributes that are valid, i.e. do not violate any of general rules defined for security descriptions as well as any specific rules defined for the transport and key method in question can be accepted. When selecting one of the valid crypto attributes, the answerer SHOULD select the most preferred crypto attribute it can support, i.e. the first valid supported crypto attribute in the list, considering the answerer's capabilities and security policies.

If there is one or more crypto attributes in the offer, but none of them are valid, or none of the valid ones are supported, the offered media stream MUST be rejected.

The crypto attribute in the answer MUST contain the following:

* The crypto-suite from the accepted crypto attribute in the offer (the same crypto-suite must be used in the send and receive direction).
* The key(s) the answerer will be using for media sent to the offerer.

There are no general offer/answer rules for the session parameters; instead, specific rules are provided as part of the transport and key-method specific definitions of any session parameters.

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

4.1.2.2 Multicast Streams

The rules for multicast streams are similar to those for unicast streams, except as noted below:

* The crypto-suite in the answer MUST be the same as the one in the offer (unless the offered media stream is rejected). Since no more than one crypto attribute can be offered for a multicast stream, this is satisfied trivially.

* The key(s) provided apply to media in all directions supported by the media stream, as opposed to just the sending direction. Consequently, the key(s) in the answer MUST be the same as the key(s) in the offer.

4.1.3 Offerer Processing of the Initial Answer

4.1.3.1 Unicast Streams

When the offerer receives the answer, the offerer MUST verify, that exactly one of the offered crypto attributes was accepted. Otherwise, the offerer MUST consider the offer/answer negotiation to have failed for that stream.

The key(s) included in the answer are the key(s) that will be used for media sent from the answerer to the offerer and hence the offerer MUST use those key(s) to process media received; the key(s) might not be the same as the key(s) used by the offerer for sending media to the answerer.

There are no general offer/answer rules for the session parameters; instead, specific rules are provided as part of the transport and

key-method specific definitions of any session parameters.

4.1.3.2 Multicast Streams

When the offerer receives the answer, the offerer MUST verify, that the offered crypto attribute and key(s) were accepted and echoed in the answer. Otherwise, the offerer MUST consider the offer/answer negotiation to have failed for that stream for *that answerer* and hence the answerer is not considered a participant in that media stream. If there are other participants in the multimedia session, the session may continue unaffected by this particular answerer's failure.

There are no general offer/answer rules for the session parameters; instead, specific rules are provided as part of the transport and key-method specific definitions of any session parameters.

4.1.4 Modifying the Session

Once a media stream has been established, it MAY be modified at any time, as described in RFC 3264, Section 8. Such a modification MAY be triggered by the security service, e.g. in order to perform a re-keying or change the crypto-suite. If media stream security using the general security descriptions defined is still desired, the crypto attribute MUST be included in these new offer/answer exchanges. The procedures are similar to those defined in Section 4.1.1, 4.1.2, 4.1.3 subject to the considerations provided in RFC 3264 Section 8.

4.2 Use Outside Offer/Answer: Advertising

The crypto attribute can also be used outside the context of offer/answer. For example, when using the Session Announcement Protocol (SAP) [RFC2974], there is no negotiation of the media streams described by the SDP; instead media streams are simply advertised.

The crypto attribute defined here can be used in such environments where the crypto parameters are advertised in a single message rather than being negotiated in a roundtrip (an offer and an answer), albeit with certain restrictions:

* There MUST be exactly one crypto attribute.

There are no general rules for the session parameters; instead, specific rules for advertising session parameters are provided as part of the transport and key-method specific definitions of any session parameters.

4.3 General Backwards Compatibility Considerations

It is possible that the answerer supports a given secure transport and accepts the offered media stream, yet the answerer does not support the crypto attribute defined here. The offerer can recognize this situation by seeing an accepted media stream in the answer that does not include a crypto line. In that case, the security negotiation defined here MUST be deemed to have failed.

5. SRTP Security Descriptions

In this section, we provide definitions for security descriptions for SRTP media streams. In the next Section, we define how to use SRTP security descriptions with and without the offer/answer model.

SRTP security descriptions for a media stream MUST only be used for media streams that use the "RTP/SAVP" transport in the media (m=) line and SHALL apply to that media stream only.

There is no assurance that an endpoint is capable of configuring its SRTP service with a particular crypto attribute parameter, but SRTP guarantees minimal interoperability among SRTP endpoints through the default SRTP parameters [srtp]. More capable SRTP endpoints support a variety of parameter values beyond the SRTP defaults and these values can be configured by the SRTP security descriptions defined here. An endpoint that does not support the crypto attribute will ignore it (per [SDPnew]) and hence, if it supports SRTP, it will simply assume use of default SRTP parameters. Such an endpoint will not correctly process the particular media stream. By using the Offer/Answer model, the offerer and answerer can negotiate the crypto parameters to be used before commencement of the multimedia session (see Section 6.1).

There are over twenty cryptographic parameters listed in the SRTP specification. Many of these parameters have fixed values for particular cryptographic transforms. At the time of session establishment, moreover, there is usually no need to provide unique settings for many of the SRTP parameters, such as salt length and pseudo-random function (PRF). Thus, it is possible to simplify the list of parameters by defining "cryptographic suites" that fix a set of SRTP parameter values for the security session. This approach is followed by the SRTP security descriptions, which uses the general security description parameters as follows:

* crypto-suite: Identifies the encryption and authentication transforms
* key parameter: SRTP keying material and parameters
* session parameters: The following parameters are defined:
  - SRC: An <SSRC, ROC, SEQ> triple
  - KDR: The SRTP Key Derivation Rate is the rate that a pseudo-random function is applied to a master key
  - UNENCRYPTED_SRTP: SRTP messages are not encrypted
  - UNENCRYPTED_SRTCP: SRTCP messages are not encrypted
  - UNAUTHENTICATED_SRTP: SRTP messages are not authenticated
  - FEC_ORDER: Order of forward error correction (FEC) relative to SRTP services
  - WSH: Window Size Hint
  - Extensions: Extension parameters can be defined

Please refer to the SRTP specification for a complete list of parameters and their descriptions [Section 8.2, srtp]. The key parameter, the crypto-suite, and the session parameters shown above are described in detail in the following sections.

5.1.1.1 SRTP Key Parameter

SRTP security descriptions define use of the "inline" key method as described in the following. Use of any other keying method for SRTP security descriptions is for further study.

The "inline" type of key contains the keying material and all policy relating to that 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 MUST NOT accept incoming packets that violate the policy (e.g. after the key lifetime has expired).

The key parameter contains a semi-colon separated list of cryptographic master keys, each of which MUST be a unique cryptographically random [RFC1750] value with respect to other master keys in the entire SDP message (i.e. including master keys for other streams). Each key in the list follows the format (a formal definition is provided in Section 8.2):

"inline:" <key salt> "|" [<lifetime] "|" [MKI:length / FromTo]

key||salt concatenated key and salt, base64 encoded
lifetime key lifetime (number of packets)
MKI:length MKI and length of the MKI field in SRTP packets.
FromTo <"From", "To"> values, specifying the lifetime for a master key.

The following definition provides an example for
AES_CM_128_HMAC_SHA1_80:

inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:4

The first field ("d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj") of the parameter is the cryptographic master key appended with the master salt; the two are first concatenated and then base64 encoded. The length of the concatenated key and salt is determined by the crypto-suite for which the key applies. If the length (after being decoded from base64) does not match that specified for the crypto-suite, the entire crypto attribute MUST be considered invalid and an "invalid key/salt" condition SHOULD be logged. Each master key and salt MUST be a cryptographically random number and MUST be unique to the SDP message.

The second field, is the OPTIONAL lifetime of the master key as measured in maximum total number of packets using that key. The lifetime value MAY be written as a non-zero, positive integer or as a power of 2 (see the grammar in Section 8.2 for details). The "lifetime" value MUST NOT exceed the maximum packet lifetime for the crypto-suite. If the lifetime is too large or otherwise invalid then the entire crypto attribute MUST be considered invalid and an "invalid lifetime" condition SHOULD be logged. The default MAY be implicitly signaled by omitting the lifetime value (i.e. "||"). This is convenient when the SRTP cryptographic key lifetime is the default value. As a shortcut to avoid long decimal values, the syntax of the lifetime allows using the literal "2^", which indicates "two to the power of". The example above, shows a case where the lifetime is specified as 2^20. The following example, which is for the AES_CM_128_HMAC_SHA1_80 crypto-suite, has a default for the lifetime field, which means the SRTP's and SRTCP's default values will be used (2^31):

inline: YUJDZGVmZ2hpSktMbW9QUXJzVHVWd3l6MTIzNDU2||1066:4

The example shows a 30-character key and concatenated salt that is base64 encoded: The 30-character key/salt concatenation is expanded to 40 characters by the three-in-four encoding of base64.

The third field, which is also OPTIONAL, is either the Master Key Identifier (MKI) and its byte length, or a <"From", "To"> value.

"MKI" is the master key identifier associated with the SRTP master key. If the MKI is given, then the length of the MKI MUST also be given and separated from the MKI by a colon (":"). 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 if it exceeds 128 (bytes), then the entire crypto attribute MUST be considered invalid and an "invalid MKI length" condition SHOULD be logged. The substring

"1:4" in the first example assigns to the key a master key identifier of 1 that is 4 bytes long, and the second example assigns a 4-byte key identifier of 1066 to the key.

<"From", "To"> specifies the lifetime for a master key, expressed in terms of the ROC and SEQ values inside whose range (including the range end-points) the master key is valid. <"From", "To"> is an alternative to the MKI and assumes that a master key is in one-to-one correspondence with the SRTP session key on which the <"From", "To"> range is defined. The following example illustrates the use of the <"From", "To"> parameter:

```
inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|FT=0:0,1:0
```

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 MUST either include an MKI or a <"From", "To"> value. Note that it is not permissible to mix and match use of the two within a single key parameter (i.e., one crypto attribute); all master keys in a given key parameter must use one or the other.

5.2 Crypto-suites

The SRTP crypto-suites define the encryption and authentication transforms to be used for the SRTP media stream. The SRTP specification has defined three crypto-suites, which below are described in the context of the SRTP security descriptions.

5.2.1 AES_CM_128_HMAC_SHA1_80

AES_CM_128_HMAC_SHA1_80 is the SRTP default AES Counter Mode cipher and HMAC-SHA1 message authentication having an 80-bit authentication tag. The master-key length is 128 bits and has a default lifetime of a maximum of 2^31 SRTP packets or SRTCP packets, whichever comes first [srtp]. The SRTP and SRTCP encryption key lengths are 128 bits. The SRTP and SRTCP authentication key lengths are 160 bits (see Security Considerations in Section 7). The master salt value is 112 bits in length and the session salt value is 112 bits in length. The pseudo-random function (PRF) is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

The length of the base64 decoded key and salt value for this crypto-suite MUST be 30 characters, i.e. 240 bits; otherwise the crypto attribute is considered invalid.

5.2.2 AES_CM_128_HMAC_SHA1_32

This crypto suite is identical to AES_CM_128_HMAC_SHA1_80 except that the SRTP authentication key is 32 bits and the SRTCP authentication key is 80 bits.

The length of the base64-decoded key and salt value for this crypto-suite MUST be 30 characters, i.e. 240 bits; otherwise the crypto attribute is considered invalid.

5.2.3 F8_128_HMAC_SHA1_80

This crypto suite is identical to AES_CM_128_HMAC_SHA1_80 except the cipher is F8 [srtp].

The length of the base64 decoded key and salt value for this crypto-suite MUST be 30 characters, i.e. 240 bits; otherwise the crypto attribute is considered invalid.

5.2.4 Adding new Crypto-suite Definitions

If new transforms are added to SRTP, new definitions for those transforms SHOULD be given for the SRTP security descriptions and published in an IETF RFC. Sections 5.2.1 through 5.2.3 illustrate how to define crypto-suite values for particular cryptographic transforms. Any new crypto suites MUST be registered with IANA following the guidelines in section 10.

5.3 Session Parameters

SRTP security descriptions define a set of "session" parameters, which OPTIONALLY may be used to override SRTP session defaults for the SRTP and SRTCP streams. These parameters configure an RTP session for SRTP services and are described in the following.

5.3.1 SRC=SSRC/ROC/SEQ

The SRTP cryptographic context for a given SRTP session is identified by the synchronization source (SSRC). Furthermore, associated with a cryptographic context is the SRTP packet index which is derived from the RTP sequence number (SEQ) and a rollover counter (ROC). The SSRC and SEQ are included in the SRTP packets, however they are not included in standard SDP (for various reasons). The ROC is neither included in the SRTP packets nor standard SDP but is instead derived algorithmically based on the total number of packets sent. This presents a couple of challenges:

* If the master key is shared between two or more session participants, SSRC collisions MUST be avoided; SSRC collision detection and resolution is not an acceptable alternative as this can lead to the two-time pad problem [srtp].

* If a participant joins an ongoing session (where the ROC is non-zero), the participant needs to learn the ROC somehow.

* If the initial sequence number is close to the maximum sequence number and the initial SRTP packets are lost, the receiver may not update his ROC correctly.

* When joining a multicast RTP session with multiple participants, a separate crypto context needs to be established for each participant (SSRC). Even if the same master key is used by all participants, the ROC for each still needs to be learned somehow.

The SRC session parameter provides information to establish the SRTP cryptographic context. It contains information about one or more of the following:

* SSRC: Synchronization source
* ROC: Roll-over counter
* SEQ: Sequence number

The ROC and sequence number are typically only needed for sessions already in progress (as when rekeying or when joining a multicast session).

Zero or more SRC parameters MAY appear in a crypto attribute. When more than one SRC parameter is present, each of them MUST contain a distinct SSRC value. Each SRC parameter defines a separate SRTP crypto context (see section 3.2 of [srtp]) that SHALL share the master key and salt defined by one or more inline key parameters. The total number of all packets that are encrypted by keys derived from this master key MUST NOT exceed the lifetime of the inline key. The SRTP crypto contexts so defined SHALL also have a common definition for the crypto-suite and all other crypto parameters.

SSRC is the RTP SSRC that is associated with the crypto context, and is an integer in the range of 0..2^32-1. If an SSRC value is invalid, the entire crypto attribute line MUST be considered invalid and an "invalid SSRC" condition SHOULD be logged. If an SSRC value collides with an SSRC for an existing participant in the session, the entire crypto attribute line MUST be considered invalid and an "SSRC collision" condition SHOULD be logged.

OPEN ISSUE: It would be nice to have a way of indicating this condition in an answer SDP, but we quickly end up duplicating the RTP collision detection and resolution, which we don't want to.

ROC is the SRTP rollover counter (ROC) in the range of 0..2^32-1 and is zero by default. Typically the ROC value is specified as a non-zero value for an ongoing SRTP stream in which the ROC has cycled one or more times [srtp]. The receiver of the SDP message SHOULD refresh the ROC value before joining an ongoing session. Depending

on the nature of the session control, the late-joining receiver might need to refresh its ROC value through a unicast exchange or through receipt of a multicast or unicast message containing a ROC

SRTP description. If the ROC is greater than 2^32-1, then the entire crypto attribute line MUST be considered invalid and an "invalid ROC" condition SHOULD be logged.

SEQ is the SRTP sequence number (SEQ), which MUST be in the range of 0..2^16-1. SRTP uses the RTP sequence number and the ROC to compute the packet index [srtp]. For this reason, the initial SEQ SHOULD be in the range of 0..2^15-1 to avoid an ambiguity when packets are lost at the start of the session. At the start of a session, an SSRC source that randomly selects a high sequence-number value can put the receiver in an ambiguous situation: If initial packets are lost in transit up to the point that the sequence number wraps (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 [srtp].

It is not necessary to signal SEQ and ROC at the start of the SRTP session if the receivers do not join the session late, which is typical in IP telephony, multimedia client/server, and similar applications. Large-scale multicast applications, however, will sometimes have late joiners to the session and MAY choose to use the SRC session parameter to set the SEQ and the ROC. The SSRC MAY also be initialized in the SRC parameter; this can for example be useful to establish the crypto contexts (in particular the ROC) for all the session participants.

Like SEQ and ROC, SSRC is OPTIONAL (unless there are multiple SRC parameters in which case it is mandatory) and often need not be signaled. If the master key is not shared among senders for their encryption services, then SSRC uniqueness is NOT REQUIRED (see Section 7.2) and the SSRC need not be signaled. In this way, each master key is used for encryption by exactly one sender and used for decryption by one or more receivers: In this case, there is no risk of keystream reuse for the crypto-suite ciphers of Section 5.2.1, 5.2.2, and 5.2.3.

The SRTP crypto context can be established for the SRTP session address in the connection (c=) line and the port in the media (m=) line (or rtpmap) without having specified an SSRC value in the SRTP security descriptions. This is called "late binding" by this specification. If late binding is used, then when a packet arrives, the SSRC that is contained in it can be bound to the crypto context at the time of session commencement rather than at the time of session signaling. With the arrival of the packet containing the

SSRC, all the data items (except the ROC if it is non-zero) needed for the SRTP crypto context are held by the receiver. In other

words, the crypto context for an RTP/SAVP session using late binding is initially identified by the SDP as:

<*, address, port>

where '*' is a wildcard SSRC, "address" is from the "c=" line, and "port" is from the "m=" line. 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). Endpoints that do not wish to subject themselves to such security risks can either signal the SSRC by out-of-band mechanisms (as defined here), or ensure that only authenticated SRTP is being used.

5.3.2 KDR=n

KDR specifies the Key Derivation Rate, as described in section 4.3.1 of [srtp].

The value n MUST be an integer in the set {0,1,2,...,24}, which denotes a power of 2 from 2^0 to 2^24, inclusive. The SRTP key derivation rate controls how frequently a new session key is derived from an SRTP master key [srtp]. The default value is 0, which causes the key derivation function to be invoked exactly once (since 2^0 is 1).

5.3.3 UNENCRYPTED_SRTCP and UNENCRYPTED_SRTP

SRTP and SRTCP packet payloads are encrypted by default. The UNENCRYPTED_SRTCP and UNENCRYPTED_SRTP session parameters modify the default behavior of the crypto-suites with which they are used:

* UNENCRYPTED_SRTCP signals that the SRTCP packet payloads are not encrypted.

* UNENCRYPTED_SRTP signals that the SRTP packet payloads are not

encrypted.

5.3.4 UNAUTHENTICATED_SRTP

SRTP and SRTCP packet payloads are authenticated by default. The UNAUTHENTICATED_SRTP session parameter signals that SRTP messages are not authenticated. Use of UNAUTHENTICATED_SRTP is NOT RECOMMENDED (see Security Considerations).

The SRTP specification requires use of message authentication for SRTCP, but not for SRTP [srtp].

5.3.5 FEC_ORDER=order

FEC_ORDER signals the use of forward error correction for the RTP packets [rfc2733]. The forward error correction values for "order" are FEC_SRTP, SRTP_FEC, or SPLIT [mikey]. FEC_SRTP 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; FEC_SRTP is the default. SRTP_FEC is the reverse processing. SPLIT signals that the sender performs SRTP encryption, followed by FEC processing, followed by SRTP authentication; processing is reversed on the receiver.

5.3.6 Window Size Hint (WSH)

SRTP defines the SRTP-WINDOW-SIZE [SRTP, section 3.3.2] parameter to protect against replay attacks. The minimum value, per [srtp], is 64, 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 SDP attributes like "a=maxprate" and the bandwidth modifiers to allow a receiver to derive the parameter satisfactorily. Consequently, this value is only considered a hint to the receiver of the SDP which MAY choose to ignore the value provided.

5.3.7 SRTP Extension Session Parameters

New SRTP session parameters for the SRTP security descriptions can be defined in an IETF RFC and registered with IANA according to the registration procedures defined in Section 10.

SRTP extension session parameters are by default mandatory. An SRTP extension session parameter that is prefixed with the dash character ("-") however is considered optional and MAY be ignored. If a SDP is received with an unknown mandatory session parameter in a crypto attribute, that crypto attribute MUST be considered invalid and a

"unknown session parameter" condition SHOULD be logged.

6. SRTP-Specific Use of the crypto Attribute

In this section, we describe the SRTP-specific use of the crypto attribute.

6.1 Use with Offer/Answer

In this section, we describe how the SRTP security descriptions are used with the offer/answer model to negotiate cryptographic capabilities and communicate SRTP master keys. The rules defined below complement the general offer/answer rules defined in Section 4.1, which MUST be followed, unless otherwise specified.

6.1.1 Generating the Initial Offer

6.1.1.1 Unicast Streams

When the initial offer is generated, the offerer MUST follow the steps in Section 4.1.1.1 as well as the following steps.

For each unicast media line (m=) using the "RTP/SAVP" transport where the offerer wants to specify cryptographic parameters, the offerer MUST provide at least one valid SRTP security description ("a=crypto" line), as defined in Section 5.

The offerer MAY include one or more SRTP session parameters as defined in Section 5.3. Note however, that if any extension SRTP session parameters are included, the negotiation will fail if the answerer does not support them.

6.1.1.2 Multicast Streams

When the initial offer is generated, the offerer MUST follow the steps in Section 4.1.1.2 as well as the following steps.

For each multicast media line (m=) using the "RTP/SAVP" transport where the offerer wants to specify cryptographic parameters, the offerer MUST provide at least one valid SRTP security description ("a=crypto" line), as defined in Section 5. Furthermore, the <"From", "To"> parameter in the key parameter MUST NOT be used, unless the media stream is marked as "recvonly".

> The <"From", "To"> value is SSRC specific, and hence will only work when there is a single sender in the multicast case, i.e. all invited participants only receive media.

The offerer MAY include one or more SRTP session parameters as defined in Section 5.3. Note however, that if any extension SRTP session parameters are included, the negotiation will fail if the

answerer does not support them.

6.1.2 Generating the Initial Answer

6.1.2.1 Unicast Streams

When the initial answer is generated, the answerer MUST follow the steps in Section 4.1.2.1 as well as the following steps.

For each unicast media line using the "RTP/SAVP" transport that contains one or more "a=crypto" lines in the offer, the answerer MUST either accept one of the crypto lines for that media stream, or it MUST reject the media stream. Only "a=crypto" lines that are considered valid SRTP security descriptions as defined in Section 5 can be accepted. Furthermore, all parameters (crypto-suite, key parameter, and session parameters) MUST be acceptable to the answerer in order for the offered media stream to be accepted.

When the answerer accepts an "RTP/SAVP" unicast media stream with a crypto line, the answerer indicates acceptance by including its own "a=crypto" line in the answer. The answer crypto line MUST include at least the selected SRTP crypto-suite and one or more master keys appropriate for the selected crypto algorithm; the master key(s) included in the answer SHOULD be different from those in the offer.

> If the master key(s) are not shared between the offerer and answerer, SSRC collisions are acceptable, which simplifies the overall operation.

Session parameters MAY be included in the answer as well; any session parameters included in the answer are independent of session parameters included in the offer. Use of extension SRTP session parameters SHOULD be avoided unless it is known that the offerer supports these.

If the answerer cannot find any valid crypto line that it supports, or its configured policy prohibits any cryptographic key parameter (e.g. key length) or cryptographic session parameter (e.g. KDR, FEC_ORDER), it MUST reject the media stream, unless it is able to successfully negotiate use of "RTP/SAVP" by other means outside the scope of this document (e.g., by use of MIKEY [mikey]).

6.1.2.2 Multicast Streams

When the initial answer is generated, the answerer MUST follow the steps in Section 4.1.2.2 as well as the following steps.

For each multicast media stream using the "RTP/SAVP" transport that contains an "a=crypto" line in the offer, the answerer MUST either accept the first crypto line for that media stream (note that there should only be one crypto line), or it MUST reject the media stream.

The crypto line MUST only be accepted if it is considered a valid SRTP security description as defined in Section 5. Furthermore, all

parameters (crypto-suite, key parameter, and session parameters) MUST be acceptable to the answerer in order for the offered media stream to be accepted.

When the answerer accepts an "RTP/SAVP" multicast media stream with a crypto line, the answerer indicates acceptance by repeating the crypto line from the offer in the answer, except for the session parameters which SHOULD be excluded.

> There is only a single view of a multicast stream (unlike unicast), and hence there is no reason to repeat optional parameters that cannot change anyway.
>
> OPEN ISSUE: It is not clear that all session parameters should be excluded from the answer. In particular, we may want to allow for inclusion of the SRC parameter, as this would enable a new-comer to instantiate crypto-contexts for other participants in a multicast conference, provided the conference is using a shared key. If each sender uses a unique key, something else would be needed (e.g. an offer/answer exchange with each participant or an entirely different mechanism).

If the answerer cannot find any valid crypto line that it supports, or its configured policy prohibits any cryptographic key parameter (e.g. key length) or cryptographic session parameter (e.g. KDR, FEC_ORDER), it MUST reject the media stream.

It should be noted, that multicast streams with more than one sender that are negotiated by use of this mechanism will be using the same master key for sending and receiving and hence SSRC collisions must be avoided. The mechanism defined here does not provide a way to avoid such SSRC collisions for multicast streams, and hence means outside of the scope of this document are needed to ensure that SSRC collisions are avoided. Examples of how this can be achieved include a centralized controller supplying unique SSRCs to the session participants or a separate protocol that can ensure SSRC uniqueness prior to sending any SRTP packets.

6.1.3 Offerer Processing of the Initial Answer

6.1.3.1 Unicast Streams

When the offerer receives the answer, it MUST perform the steps in Section 4.1.3.1 as well as the following steps for each "RTP/SAVP" media stream it offered with one or more crypto lines in it.

If the media stream was accepted and it contains a crypto line, it MUST be checked that the crypto line is valid according to the constraints specified in Section 5.

If the crypto line contains any SRTP session parameters, those parameters define SRTP behavior for media sent from the answerer to the offerer. If the offerer either does not support or is not willing to honor one or more of the SRTP session parameters in the answer, the offerer MUST consider the crypto line invalid.

If the crypto line is not valid, or the offerer's configured policy prohibits any cryptographic key parameter (e.g. key length) or cryptographic session parameter, the SRTP security negotiation MUST be deemed to have failed.

6.1.3.2 Multicast Streams

When the offerer receives the answer, it MUST perform the steps in Section 4.1.3.2 as well as the following steps for each "RTP/SAVP" media stream it offered with a crypto line in it.

If the media stream was accepted and it contains a crypto line, it MUST be checked that the crypto line is valid according to the constraints specified in Section 5. If the crypto line includes any session parameters, those are simply ignored.

OPEN ISSUE: As noted in Section 6.1.2.2, it may make sense to allow for some session parameters, e.g. SRC, to be included.

If the crypto line is not valid, the SRTP security negotiation MUST be deemed to have failed for that particular answerer.

6.1.4 Modifying the Session

When a media stream using the SRTP security descriptions has been established, and a new offer/answer exchange is performed, the offerer and answerer MUST follow the steps in Section 4.1.4 as well as the following steps.

Unicast Streams:

* The offerer SHOULD include the ROC and SEQ (unless both are made available to the answerer by other means); this enables the answerer to establish the complete crypto context in case he currently does not have the ROC.

Multicast Streams:

* When the media stream is "recvonly", the offerer SHOULD include the ROC and SEQ (unless both are made available to the answerer by other means); this enables the answerer to establish the complete crypto context in case he currently does not have the ROC.

It should be noted, that the mechanism defined here does not provide a way to communicate the ROC for multiple senders, which may be

needed in some multicast scenarios, e.g. conferencing. If renegotiation is needed, a separate mechanism, such as [GDOI], will

be needed for this. These methods are beyond the scope of this document.

OPEN ISSUE: As noted in Section 6.1.2.2, it is not clear that we couldn't do that with the SRC parameter.

When modifying the session, all negotiated aspects of the SRTP media stream can be modified. For example, a new crypto suite can be used or a new master key can be established. As described in RFC 3264, when doing a new offer/answer exchange there will be a window of time, where the offerer and the answerer must be prepared to receive media according to both the old and the new offer/answer exchange. This requirement applies here as well, however the following should be noted:

* When authentication is not being used, it may not be possible for either the offerer or the answerer to determine if a given packet is encrypted according to the old or new offer/answer exchange. RFC 3264 defines a couple of techniques to address this problem, e.g. changing the payload types used and/or the transport addresses. Note however that a change in transport addresses may have an impact on Quality of Service as well as firewall and NAT traversal. The SRTP security descriptions offers two other ways of dealing with this; use the MKI (which adds a few bytes to each SRTP packet) or the <"From","To"> mechanism (which doesn't add bytes to each SRTP packet) as described in Section 5.1.1.1. For further details on MKI and "<"From","To">, please refer to [srtp].

* If the answerer changes its master key, the offerer will not be able to process packets secured via this master key until the answer is received.

  As noted in Section 4.1.1.1, this could for example be addressed by defining a security "precondition" [RFC3312]

Finally note, that if the new offer is rejected, the old crypto parameters remain in place.

6.1.5 Offer/Answer Example

In this example, the offerer supports two crypto suites (F8 and AES). The a=crypto line is actually one long line, although it is shown as two lines in this document due to page formatting.

```
Offerer sends:
  v=0
  o=sam 2890844526 2890842807 IN IP4 10.47.16.5
  s=SRTP Discussion
  i=A discussion of Secure RTP
  u=http://www.example.com/seminars/srtp.pdf
  e=marge@example.com (Marge Simpson)
  c=IN IP4 168.2.17.12
  t=2873397496 2873404696
  m=audio 49170 RTP/SAVP 0
  a=crypto:AES_CM_128_HMAC_SHA1_80
   inline:WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz|2^20|1:4
   FEC_ORDER=FEC_SRTP SRC=//49126
  a=crypto:F8_128_HMAC_SHA1_80
   inline:MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm|2^20|1:4
   FEC_ORDER=FEC_SRTP SRC=//49126

Answerer replies:
  v=0
  o=jill 25690844 8070842634 IN IP4 10.47.16.5
  s=SRTP Discussion
  i=A discussion of Secure RTP
  u=http://www.example.com/seminars/srtp.pdf
  e=homer@example.com (Homer Simpson)
  c=IN IP4 168.2.17.11
  t=2873397526 2873405696
  m=audio 32640 RTP/SAVP 0
  a=crypto:AES_CM_128_HMAC_SHA1_80
   inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
   SRC=/721/13
```

In this case, the session would use the AES_CM_128_HMAC_SHA1_80 crypto suite for the RTP and RTCP traffic. The answerer is also specifying both its current rollover counter (721), and sequence number (13).

6.2 SRTP-Specific Use Outside Offer/Answer: Advertising

The SRTP security descriptions can be used outside the context of offer/answer as described in Section 4.2. In those cases, the general rules defined in Section 4.2 as well as the SRTP-specific rule defined below MUST be followed:

* If any SRTP session parameters are included, they MUST be supported by the recipient of the SDP; otherwise, the recipient MUST NOT join the SRTP session.

6.3 SRTP-Specific Backwards Compatibility Considerations

It is possible that the answerer supports the "RTP/SAVP" transport and accepts the offered media stream, yet it does not support the

crypto attribute defined here. The offerer can recognize this situation by seeing an accepted "RTP/SAVP" media stream in the answer that does not include a crypto line. In that case, the security negotiation defined here MUST be deemed to have failed.

Also, if a media stream with transport set to "RTP/SAVP" is sent to a device that does not support "RTP/SAVP", that media stream will be rejected.

6.4 Operation with KEYMGT= and k= lines

An offer MAY include both "a=crypto" and "a=keymgt" lines [keymgt]. Per SDP rules, the answerer will ignore attribute lines it does not understand. If the answerer supports both "a=crypto" and "a=keymgt", the answer MUST include either "a=crypto" or "a=keymgt" but not both, as including both is undefined.

An offer MAY include both "a=crypto" and "k=" lines [SDPnew]. Per SDP rules, the answerer will ignore attribute lines it does not understand. If the answerer supports both "a=crypto" and "k=", the answer MUST include either "a=crypto" or "k=" but not both, as including both is undefined.

6.5 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 can not automatically be recovered (e.g. ROC and SEQ), 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 once the crypt context is removed.

We resolve this problem as follows. When SRTP security descriptions are being used, crypto contexts removal MUST follow the same rules as SSRC removal from the member table [RFC 3550]; 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.

7. Security Considerations

Like all SDP messages, SDP messages containing security descriptions, are conveyed in an encapsulating application protocol

(e.g. SIP, MGCP, RTSP, SAP, etc.). It is the responsibility of the encapsulating protocol to ensure the protection of the SDP security

descriptions. Therefore, the application protocol SHOULD either invoke its own security mechanisms to do so, or alternatively utilize a lower-layer security service (e.g. TLS, IPSEC). This security service SHOULD provide strong message authentication and packet-payload encryption as well as effective replay protection.

7.1 Authentication of packets

Security descriptions as defined herein signal security services for RTP packets. RTP messages are vulnerable to a variety of attacks such as replay and forging. To limit these attacks, SRTP message integrity mechanisms SHOULD be used (SRTP replay protection is always enabled). Source authentication (i.e. data-origin authentication) of unicast SRTP messages SHOULD be performed [srtp].

7.2 Keystream Reuse

Security descriptions as defined herein signal configuration parameters for SRTP sessions. Misconfigured SRTP sessions are vulnerable to attacks on their encryption services when running the crypto suites defined in Sections 5.2.1, 5.2.2, and 5.2.3. An SRTP encryption service is "misconfigured" when two or more media streams are encrypted using the same AES keystream. When senders and receivers share derived session keys, SRTP requires that the SSRCs of session participants serve to make their corresponding keystreams unique, which is violated in the case of SSRC collision: SRTP SSRC collision drastically weakens SRTP or SRTCP payload encryption during the time that identical keystreams were used [srtp]. An attacker, for example, might collect SRTP and SRTCP messages and await a collision. This attack on the AES-CM and AES-f8 encryption is avoided entirely when each media stream has its own unique master key in both the send and receive direction, as this document RECOMMENDS (see Section 6.1.2.1), i.e. keys are not shared between multiple media streams, and the keys used in the send and receive direction for a given media stream are unique.

SRTP multicast operation requires that each host-sender have a unique SRTP keystream. This can be accomplished by ensuring that each sender be allocated a unique key or by ensuring that the SSRC of each sender will not collide. Since SSRC collision might occur, the latter condition is avoided when all SSRCs are assigned by a central authority such as a 3rd-party key server [srtp]. The RECOMMENDED approach of this document is to allocate a different master key for each host-participant of an SRTP session.

7.3 Signaling Authentication and Signaling Encryption

There is no reason to incur the complexity and computational expense

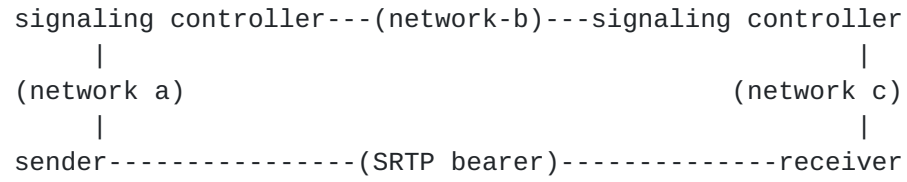
of SRTP, however, when its key establishment is exposed to unauthorized parties. In most cases, the SRTP crypto attribute and its parameters are vulnerable to denial of service attacks when they

are carried in an unauthenticated SDP message. In some cases, the integrity or confidentiality of the RTP stream can be compromised. For example, if an attacker sets UNENCRYPTED for the SRTP stream in an offer, this could result in the answerer not decrypting the encrypted SRTP messages. In the worst case, the answerer might itself send unencrypted SRTP and leave its data exposed to snooping.

Thus, IPsec, TLS, or some other data security service SHOULD be used to provide message authentication for the encapsulating protocol that carries the SDP messages having a crypto attribute (a=crypto). Furthermore, encryption of the encapsulating payload SHOULD be used because a master key parameter (inline) appears in the message. Failure to encrypt the SDP message containing an inline SRTP master key renders the SRTP authentication or encryption service useless in practically all circumstances. Failure to authenticate an SDP message that carries SRTP parameters renders the SRTP authentication or encryption service useless in most practical applications.

When the SDP parameters cannot be carried in an encrypted and authenticated SDP message, it is RECOMMENDED that a key management protocol be used instead of the security descriptions defined here (a=crypto). The proposed SDP key-mgmt extension [keymgt] allows authentication and encryption of the key management protocol data independently of the SDP message that carries it. The security of the SDP SRTP attribute, however, is as good as the data security protocol that protects the SDP message. For example, if an IPsec security association exists between the source and destination endpoints, then this solution is more secure than use of the key-mgmt statement in an unauthenticated SDP message, which is vulnerable to tampering.

There are practical cases, however, where SDP security is not end-to-end: If there is a third-party provider between the sender and receiver, then the data-security session might not be end-to-end. That is, one possible configuration might have an IPsec or TLS connection between the sender of the SDP message and the provider, such as a VoIP service provider, with a second secure connection between the provider and the receiver:



where all of link a, b, and c are encrypted with TLS or IPsec.

In this case, the third-party provider has access to the contents of

the SRTP descriptions in the SDP message. SDP key-mgmt statement, however, allows true end-to-end security that is independent of the service provider, who often needs access to some parts of the SDP

message to render its services. The SRTP attribute SHOULD NOT be used when end-to-end authentication or confidentiality is needed but the SDP message is not secured end-to-end (such as the above example where a third-party provider maintains the security associations with the endpoints for the SDP message).

8. Grammar

8.1 Generic "Crypto" Attribute Grammar

The ABNF grammar for the crypto attribute is defined below:

```
"a=crypto:" crypto-suite 1*WSP key-params *(1*WSP session-param)

crypto-suite     = 1*(ALPHA / DIGIT / "_")

key-params       = key-param *(";" key-params)
key-param        = key-method ":" key-info
key-method       = "inline" | key-method-ext
key-method-ext   = 1*(ALPHA / DIGIT / "_")
key-info         = %x21-3A / %x3B-7E ; visible (printing) characters
                                     ; except semi-colon
session-param    = VCHAR   ; visible (printing) characters
```

where WSP, ALPHA, DIGIT, and VCHAR are defined in [RFC2234].

8.2 SRTP "Crypto" Attribute Grammar

This section provides an Augmented BNF [RFC2234] grammar for the SRTP-specific use of the SDP crypto attribute:

```
  crypto-suite   = srtp-crypto-suite
  key-method     = srtp-key-method
  key-info       = srtp-key-info
  session-param  = srtp-session-param

  srtp-crypto-suite   = "AES_CM_128_HMAC_SHA1_32" /
                        "F8_128_HMAC_SHA1_32" /
                        "AES_CM_128_HMAC_SHA1_80" /
                        srtp-crypto-suite-ext

  srtp-key-method     = "inline"
  srtp-key-info       = key-salt "|" [lifetime] "|" [mki / FromTo]

  key-salt       = 1*(base64)   ; binary key and salt values
                                ; concatenated together, and then
                                ; base64 encoded [section 6.8 of
                                ; RFC2046]
```

```
lifetime      = ["2^"] 1*(DIGIT)   ; see section 5.1.1.1 for "2^"
mki            = mki-value ":" mki-length
```

```
mki-value      = 1*DIGIT
mki-length     = 1*3DIGIT   ; range 1..128.
FromTo         = "FT=" ftval "," ftval
ftval          = roc ":" seq  ; packet index expressed in terms
                              ; of ROC and SEQ.

srtp-session-param  = src /
                      kdr /
                      "UNENCRYPTED_SRTP" /
                      "UNENCRYPTED_SRTCP" /
                      "UNAUTHENTICATED_SRTP" /
                      fec-order /
                      wsh /
                      srtp-session-extension

src  = "SRC=" [ssrc] "/" [roc] "/" [seq]

ssrc = 1*DIGIT                ; range 0..2^32-1
roc  = 1*DIGIT                ; range 0..2^32-1
seq  = 1*DIGIT                ; range 0..2^16-1

kdr  = "KDR=" 1*2(DIGIT)  ; range 0..24, power of two

fec-order = "FEC_ORDER=" fec-type
fec-type  = "FEC_SRTP" / "SRTP_FEC" / "SPLIT"

wsh       = "WSH=" 2*DIGIT    ; minimum value is 64
base64    =  ALPHA / DIGIT / "+" / "/" / "="

srtp-crypto-suite-ext  = 1*(ALPHA / DIGIT / "_")
srtp-session-extension = ["-"] 1*(VCHAR)  ;visible chars [RFC2234]
                         ; first character must not be dash ("-")
```

9. Open Issues

The following is a list of open issues in this document:

* The use of security descriptions, and in particular SRTP security descriptions, with multicast streams where offer/answer is being used is not well understood and requires further consideration.

* The security descriptions do not deal with hierarchically encoded streams (or at least they have not been considered).

* The current mechanism does not allow for a key to be specified as being an encryption or decryption key or both; instead this is inferred from the context (e.g. unicast offer). Should there be a mechanism to allow a key to be tagged as an encryption, decryption or both key ?

10. IANA Considerations

10.1 Registration of the "crypto" attribute

The IANA is hereby requested to register a new SDP attribute as follows:

Attribute name:	crypto
Long form name:	Security description cryptographic attribute for media streams
Type of attribute:	Media-level
Subject to charset:	No
Purpose:	Security descriptions
Appropriate values:	See Section 3

10.2 New IANA Registries and Registration Procedures

The following sub-sections define several new IANA registries to be used for the security descriptions. It is suggested that the following registry structure be used for these:

```
Security Descriptions
  |
  +- Key Methods (described in 10.2.1)
  |
  +- Media Stream Transports
       |
       +- SRTP
            |
            +- SRTP crypto suites (described in Section 10.2.2)
            |
            +- SRTP session parameters (described in Section 10.2.3)
```

10.2.1 Security Descriptions Key Method Registry and Registration

The IANA is hereby requested to create a new registry for SDP security description key methods. An IANA key method registration MUST be documented in an IETF RFC and it MUST provide the name of the key method in accordance with the grammar for key-method-ext defined in Section 8.1.

10.2.2 SRTP Crypto Suite Registry and Registration

The IANA is hereby requested to create a new registry for SRTP crypto suites. An IANA crypto suite registration MUST indicate the crypto suite name in accordance with the grammar for srtp-crypto-suite-ext defined in Section 8.2.

The semantics of the crypto suite MUST be described in an IETF RFC, including the semantics of the "inline" key-method and any special semantics of parameters.

10.2.3 SRTP Session Parameter Registration

The IANA is hereby requested to create a new registry for SRTP session parameters. An IANA SRTP session parameter registration MUST indicate the session parameter name (srtp-session-extension as defined in Section 8.2); the name MUST NOT begin with the dash character ("-").

The semantics of the parameter MUST be described in an IETF RFC. If values can be assigned to the parameter, then the format and possible values that can be assigned MUST be described in the IETF RFC as well.

10.3 Initial Registrations

The following security descriptions key methods are hereby registered:

inline

The following SRTP crypto suites are hereby registered:

AES_CM_128_HMAC_SHA1_80
AES_CM_128_HMAC_SHA1_32
F8_128_HMAC_SHA1_80

The following SRTP session parameters are hereby registered:

SRC
KDR
UNENCRYPTED_SRTP
UNENCRYPTED_SRTCP
UNAUTHENTICATED_SRTP
FEC_ORDER
WSH

The ABNF for all of the above is already included in the ABNF section of this document.

11. Acknowledgements

This document is a product of the IETF MMUSIC working group and has benefited from comments from its participants. This document also benefited from discussions with David McGrew, Mats Naslund, Mike Thomas, Elisabetta Cararra, Brian Weis, Dave Oran, Bill Foster, Earl

Carter, Matt Hammer and Dave Singer. These people shared

observations, identified errors and made suggestions for improving the specification. Mats made several valuable suggestions on parameters and syntax that are in the current draft. Dave Oran and Mike Thomas encouraged us to bring this work to the IETF for standardization. David McGrew suggested the conservative approach of requiring unique master keys for each unicast SDP media stream as followed in this document. Jonathan Rosenberg suggested reducing the complexity by specifying only one security parameter for each media stream.

12. Authors' Addresses

Flemming Andreasen
Cisco Systems, Inc.
499 Thornall Street, 8th Floor
Edison, New Jersey 08837 USA
fandreas@cisco.com

Mark Baugher
5510 SW Orchid Street
Portland, Oregon 97219 USA
mbaugher@cisco.com
+1-408-853-4418

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134 USA
dwing@cisco.com
+1-408-902-3348

13. Normative References

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003, http://www.ietf.org/rfc/rfc3550.txt.

[RFC2234] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF," RFC 2234, November 1997, http://www.ietf.org/rfc/rfc2234.txt.

[SDPnew] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", Work in Progress.

[RFC2733] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999, http://www.ietf.org/rfc/rfc2733.txt.

[RFC2828] R. Shirey, "Internet Security Glossary", RFC 2828, May

2000, http://www.ietf.org/rfc/rfc2828.txt.

[RFC3264] J. Rosenberg, H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2202, http://www.ietf.org/rfc/rfc3264.txt.

[srtp] M. Baugher, R. Blom, E. Carrara, D. McGrew, M. Naslund, K. Norrman, D. Oran, "The Secure Real-time Transport Protocol", Work in Progress.

[RFC1750] D. Eastlake 3rd, S. Crocker, J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994, http://www.ietf.org/rfc/rfc1750.txt.

14. Informative References

[RFC3407] F. Andreasen, "Session Description Protocol (SDP) Simple Capability Declaration", RFC 3407, October 2002, http://www.ietf.org/rfc/rfc3407.txt.

[Bellovin] Steven M. Bellovin, "Problem Areas for the IP Security Protocols," in Proceedings of the Sixth Usenix Unix Security Symposium, pp. 1-16, San Jose, CA, July 1996.

[GDOI] M. Baugher, B. Weis, T. Hardjono, H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003, http://www.ietf.org/rfc/rfc3547.txt.

[kink] M. Thomas, J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", Work in Progress.

[ike] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998, http://www.ietf.org/rfc/rfc2409.txt.

[ipsec] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998, http://www.ietf.org/rfc/rfc2401.txt.

[s/mime] Ramsdell B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999, http://www.ietf.org/rfc/rfc2633.txt.

[tls] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.txt.

[keymgt] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, "Key Management Extensions for SDP and RTSP", Work in Progress.

[mikey] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, "MIKEY: Multimedia Internet KEYing", Work in Progress.

[RFC2045] N. Freed, N. Borenstein, "Multipurpose Internet Mail

Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://www.ietf.org/rfc/rfc2045.txt.

[RFC2104] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2014, November 1997, http://www.ietf.org/rfc/rfc2104.txt.

[skeme] H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for the Internet", ISOC Secure Networks and Distributed Systems Symposium, San Diego, 1996.

[RFC3312] G. Camarillo, W. Marshall, J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, October 2002, http://www.ietf.org/rfc/rfc3312.txt.

[RFC2974] M. Handley, C. Perkins, E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000, http://www.ietf.org/rfc/rfc2974.txt .

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.