Internet Engineering Task Force
MMUSIC Working Group
INTERNET-DRAFT
EXPIRES: August 2003

Mark Baugher
Dan Wing
Cisco Systems
February 24, 2003

SDP Security Descriptions for Media Streams

<draft-ietf-mmusic-sdescriptions-00.txt>

Status of this memo

Abstract

This Internet Draft gives a cryptographic attribute to Session Description Protocol (SDP) media streams. The attribute describes a cryptographic key and other parameters, which serve to configure security for a media stream. This draft also defines the SRTP parameters for the attribute. The SDP crypto attribute requires the services of a data security protocol to secure the SDP message.

TABLE OF CONTENTS

1.0 Notational Conventions

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

2.0 Introduction

Session Description Protocol (SDP) describes multimedia sessions, such as Real-time Transport Protocol (RTP), white board, fax, modem and other media sessions. Security services such as authentication and confidentiality are often needed for these media streams. When run under the RTP/SAVP profile, for example, an RTP stream uses the Secure Real-time Transport Protocol (SRTP). The "RTP/SAVP" descriptor in an SDP m=line signals the use of SRTP for a media stream, but there are no means within SDP itself to configure SRTP beyond using defaults values. This document specifies an SDP attribute to signal a cryptographic parameters in addition to a key for SRTP and other SDP media streams.

Thus, the SDP crypto attribute provides both generic and specific security descriptions for SDP media streams that can be used for various transports, including SRTP. In this way, the crypto attribute can be extended to non-SRTP transports such as white board, modem, fax, and other transports that could use various security protocols such as IPsec or TLS. Each SDP media stream, however, needs its own definitions that assign values to crypto-attribute parameters. These definitions are unique to the SDP transport and SHOULD be specified in an Internet RFC. This document defines the parameter values for SRTP. With this document, an application developer can describe an SRTP key and its configuration according to application-specific needs.

It would be self-defeating, however, to not secure cryptographic keys and other parameters at least as well as SRTP secures RTP messages 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, GDOI, KINK, IKE or TLS securely disseminates information describing both the key and the data-security session (for example, whether SRTCP is 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 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]. SDP media-stream security descriptions are suitable for restricted cases

where IPsec, TLS, or some other data-security protocol protects the SDP message.

This draft adds security descriptions to SDP messages through a new SDP attribute named "crypto", which provides the cryptographic parameters of a media stream. The crypto attribute MAY contain a cryptographic key and other parameters that describe the key. a=crypto MAY also contain "security session parameters" that are unique to a transport.

The a=crypto parameter is applicable to all media transports, but its value MAY be unique to a particular transport. Section 3 specifies the SDP crypto attribute generically. Section 4 defines the crypto attribute for SRTP. Section 5 discusses use of the crypto attribute in Offer/Answer exchanges. Section 6 recites security considerations, and Section 7 gives an Augmented-BNF grammar for the security descriptions.

3.0 SDP "Crypto" Attribute and Parameters

A new media-level SDP attribute called "crypto" describes the cryptographic and security-session parameters for one or more media entries. "a=crypto" MUST NOT appear at the SDP session level.

a=crypto:<crypto-suite> <application> <key> [<session>]

The ordering of multiple a=crypto lines is significant, and the most-preferred is listed first; see section 5. The next sections describes these fields in more detail.

3.1 Crypto-suite

"Crypto-suite" describes all needed information about the encryption and authentication algorithms for the transport. The "crypto-suite" parameter is unique to the transport.

3.2 Application Parameter

A particular transport can have multiple protocols that are secured differently. For example, when using the RTP/SAVP transport, both the SRTP and SRTCP protocols will be used, but the security for each MAY be different: A longer authentication output tag might be desired for the SRTCP control protocol than for the SRTP media stream. The "application" parameter allows separate security descriptions for separate protocols of a transport.

3.3 Key Parameter

The key parameter can contain an inline key descriptor, or can be a

pointer to a uri which contains the actual key:

```
inline:key-descriptor
uri:absolute-uri
```

If the key parameter starts with the string "uri:", the URI method is used and the value that follows is a Uniform Resource Identifier. The URI is a resource that SHOULD be queried to obtain the cryptographic key for the session. The format or protocols used for the uri are beyond the scope of this document.

The INLINE method is invoked when the key parameter starts with the string "inline:" - and the cryptographic key is encoded according to a transport-specific syntax. Thus, the URI method is transport generic and the INLINE method is transport specific. Section 4 describes the INLINE key-parameter syntax for RTP/SAVP, the SRTP media transport type.

If SDP descriptions for new media-stream transports are defined in the future, new methods MAY be defined in an Internet RFC.

3.4 Session Parameters

"Session" parameters are specific to the SDP transport and optional. Section 4 describes the session parameters for RTP/SAVP.

3.5 Examples

The first example shows a=crypto for the RTP/SAVP transport type (as defined in Section 4).

```
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 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=video 51372 RTP/SAVP 31
a=crypto:AES_CM_128_HMAC_SHA1_80 both
  inline:16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/2^20/1:32
m=audio 49170 RTP/SAVP 0
a=crypto:AES_CM_128_HMAC_SHA1_32 srtp
  inline:16/14/NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj/2^20/1:32
a=crypto:AES_CM_128_HMAC_SHA1_80 srtcp
  inline:16/14/ ZkBkQythOTg3NjU0MSEzMDMyMT01NDg5N2RlRkF/2^20/1:32
m=application 32416 udp wb
a=orient:portrait
```

This SDP message describes three "recvonly" media streams, two of which use the RTP/SAVP transport. The first a=crypto line appears

in the m=video media entry; it is associated with the RTP/SAVP transport of the m=video line and uses the SRTP default crypto-suite, "AES_CM_128_HMAC_SHA1_80," and its key parameter carries the SRTP master key data and descriptors inline. The m=video a=crypto attribute applies to both SRTP and SRTCP. The m=audio media entry uses the "crypto-suite=AES_CM_128_HMAC_SHA1_32," having a short 32-bit tag for SRTP, and it uses AES_CM_128_HMAC_SHA1_80 for SRTCP. The RTP/SAVP-specific descriptions are defined in the next section.

4.0 RTP/SAVP (SRTP) Security Descriptions

The generic security descriptions of the preceding section need parameter values for specific media transports; this section defines the crypto attribute values and parameters for the RTP/SAVP (SRTP) transport. SRTP services for a media stream MUST be signaled through the presence of an RTP/SAVP transport descriptor in the m= line and SHALL apply only to that media entry.

There is no assurance that a receiver is capable of configuring its SRTP service with a particular crypto attribute parameter, but SRTP guarantees minimal interoperability among SRTP systems through the default SRTP parameters [srtp]. More capable SRTP receivers support a variety of parameter values beyond the SRTP defaults and can be configured by the crypto attribute. A receiver that does not recognize a=crypto and assumes default SRTP parameters might receive a stream that uses non-default parameters, which will cause that receiver to fail. An Offer/Answer capabilities exchange, however, allows sender and receiver to agree on parameters before commencement of the multimedia session (see Section 5.0).

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, however, there is usually no need to provide unique settings for many of the SRTP parameters. Thus, it is possible to simplify the list of parameters in "cryptographic suites" that fix a set of SRTP parameter values for the security session. The list of SRTP parameters, including the crypto-suite parameter for SDP a=crypto follows.

SDP SRTP Parameter	Description
CRYPTO-SUITE	Encryption and authentication transforms
SSRC	SSRC of the sender of the SDP message
ROC	Roll-over counter
KEY_DERIVATION_RATE	Rate that the pseudo-random function is applied to a key
UNENCRYPTED	Protocol messages are not encrypted

UNAUTHENTICATED	SRTP messages are not authenticated
FEC_ORDER	Order of forward error correction (FEC) relative to SRTP services

Please refer to the SRTP specification for a complete list of parameters and their descriptions [Section 8.2, srtp]. The CRYPTO-SUITE and the five session parameters shown in the table above are described in the following sections. If a receiver cannot recognize a parameter or value, then the receiver MUST NOT participate in the media stream and SHOULD log an "invalid name" condition unless the receiver is participating in an Offer/Answer exchange (Section 5).

4.1 Crypto-suites

A crypto-suite value appears as the first parameter in a=crypto. The CRYPTO-SUITE value MAY be different for SRTP and SRTCP as described in Section 4.2. If a receiver does not support the particular crypto-suite, then the receiver MUST NOT participate in the media stream and SHOULD log an "unrecognized crypto-suite" condition unless the receiver is participating in an Offer/Answer exchange (Section 5). RTP/SAVP has four crypto-suites as described below.

4.1.1 AES_CM_128_HMAC_SHA1_80

This is the SRTP default AES Counter Mode cipher and HMAC-SHA1 message authentication having a 80-bit authentication tag. The encryption and authentication key lengths are 128 bits. The master salt value is 112 bits and the session salt value is 112 bits. The PRF is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

4.1.2 AES_CM_128_HMAC_SHA1_32

The SRTP AES Counter Mode cipher is used with HMAC-SHA1 message authentication having an 32-bit authentication tag. The encryption and authentication key lengths are 128 bits. The master salt value is 112 bits and the session salt value is 112 bits. These values apply to SRTP and to SRTCP. The PRF is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

4.1.3 F8_128_HMAC_SHA1_80

The SRTP f8 cipher is used with HMAC-SHA1 message authentication having a 80-bit authentication tag. The encryption and authentication key lengths are 128 bits. The master salt value is 112 bits and the session salt value is 112 bits. The PRF is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

4.1.4 F8_128_HMAC_SHA1_32

The SRTP f8 cipher is used with HMAC-SHA1 message authentication having a 32-bit authentication tag. The encryption and

authentication key lengths are 128 bits. The master salt value is 112 bits and the session salt value is 112 bits. The PRF is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

4.1.5 Adding new CRYPTO-SUITE definitions

If new transforms are added to SRTP, new definitions for those transforms SHOULD be given for the SDP crypto attribute and published in an Internet RFC. Sections 4.1.1 through 4.1.4 illustrate how to define CRYPTO-SUITE values for particular cryptographic transforms. New definitions MAY be added to existing transforms, moreover, to augment or modify definitions 4.1.1 through 4.1.4.

4.2 Application Parameter

The "application" parameter indicates if this a=crypto line applies to only secure RTP, only secure RTCP, or to both secure RTP and RTCP. The values for this are "srtp", "srtcp", or "both". If a receiver finds an "srtp" a=crypto without a corresponding "srtcp" a=crypto, or vice versa, it MUST NOT participate in the media stream and SHOULD log an "missing crypto attribute" condition.

4.3 Key Parameter

If the "key" parameter has a "uri:" descriptor, the value is a Uniform Resource Identifier value as described in Section 3. When key-parameter has an "inline:" descriptor, the value contains a cryptographic key that MUST be a unique random value with respect to other "inline:" values in the SDP message.

4.3.1 INLINE Usage

The "inline:" descriptor is applicable to SDP media-entries having a "recvonly," "sendonly" or "sendrecv" direction attribute. In general, the source of data will generate the master key to protect its data, but this is a matter of local policy and application preference. Multicast applications, for example, often will use a third-party provider of a master key. Thus, when the inline key is used, it SHOULD be used for a recvonly media-entry or for the received stream of sendrecv media-entry. The inline key MAY be used for a sendonly media-entry or for streams that are sent and received on a sendrecv media-entry. The following paragraphs add detail to these inline-key recommendations for recvonly, sendonly, and sendrecv media entries.

In the recvonly case, the inline SRTP master key SHOULD be used to derive keys [SRTP] to decrypt/authenticate incoming SRTP messages.

When the a=crypto "application" parameter is set to "both," the receiver also derives keys from the same master key to

decrypt/authenticate incoming SRTCP messages. When that receiver sends RTP Receiver Reports for the incoming SRTP stream, it SHOULD derive keys from the same master key to encrypt/authenticate outgoing SRTCP messages for that SRTP stream.

In the sendonly case, the inline SRTP master key SHOULD be used to derive keys [SRTP] to encrypt and authenticate outgoing SRTP messages. When the a=crypto "application" parameter is set to "both," the sender also derives keys from the same SRTP master key to encrypt and authenticate outgoing SRTCP message. When that sender sends RTP Sender Reports for the outgoing SRTP stream, it SHOULD derive keys from the same master key to encrypt/authenticate outgoing SRTCP messages for the outgoing SRTP stream.

In the sendrecv case, the inline SRTP master key SHOULD be used as in the recvonly case described above but MAY also be used as in the sendonly case.

4.3.2 INLINE Definition

If the identifier is "inline", the key-descriptor MUST have the following format.

key_length/salt_length/BASE64(key||salt)/lifetime/MKI:MKI_length

The "key_length" is the integer length of the SRTP master key in bytes, and "salt_length" is the integer length of the master salt in bytes. If their sum is less than the sum of the lengths of the master key and salt of the crypto suite, then the receiver MUST NOT participate in the media stream and SHOULD log a "key length too short" condition. If their sum is greater than the crypto-suite sum, then bytes are truncated from the right (i.e. "little end"). The key_length and salt_length MUST appear in the "inline" encoding. For example,

inline:16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/2^20/1:32 (1)

has a key_length of 16 and a salt_length of 14.

The third part of the "inline" encoding is the cryptographic master key appended with the master salt ("||" denotes concatenation). Each master key and salt MUST be a random number and MUST be unique to the SDP message. Both are concatenated and then base-64 encoded. If the length of the concatenated keys (after being decoded from base 64) does not equal or exceed the sum of the key_length and salt_length, the receiver MUST NOT participate in the media stream and SHOULD log a "inline encoding too short" condition. For example,

```
   inline:16/8/YUJDZGVmZ2hpSktMbW9QUXJzVHVWd3l6//1066:32            (2)
```

has a key_length of 16, a salt_length of 8, and a 32-character key and concatenated salt that is base-64 encoded: The 24-character key/salt concatenation is expanded to 32 characters by the three-in-four encoding of base 64.

The fourth part of the of the "inline" encoding is the OPTIONAL lifetime of the master key as measured in number of packets encrypted or authenticated with that key. The lifetime value MAY be written as an non-zero, positive integer or as a power of 2, and is indicated with "2^"; see the ABNF in Section 7 for details. The default value is 2^48, which is 2^48 packets encrypted with a master key according to the SRTP standard [srtp]. The "lifetime" value MUST NOT exceed the maximum packets lifetime for the crypto-suite (e.g. 2^48 for AES Counter Mode with a 128-bit key). If lifetime is too large or otherwise invalid, then the receiver MUST NOT participate in the media stream and SHOULD log an "invalid lifetime" condition. The default MAY be implicitly signaled by having no described value for lifetime (i.e. "//"). This is convenient when the srtp crypto_key lifetime is allowed to default. Trailing slashes ("/") MUST follow the master key and lifetime; otherwise, the receiver MUST NOT participate in the media stream and SHOULD log an "invalid inline encoding" condition. Example (1), above, shows a case where the lifetime is specified as 2^20 while example (2) shows an empty lifetime that implicitly uses the SRTP default value of 2^48.

The MKI value is OPTIONAL as is its specified bit length (see Section 7). "MKI" is the master key index 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 bits, and MUST be a positive multiple of 8. If the MKI_length is not given or if the value exceeds 128, then the receiver MUST NOT participate in the media stream and SHOULD log an "invalid MKI_length" condition. If the value of the MKI is larger than allowed by MKI_length, then the receiver MUST NOT participate in the media stream and SHOULD log an "invalid MKI" condition. The substring "1:32" in example (1) assigns to the key a key index of 1 that is 32 bits long, and example (2) assigns a 32-bit key index of 1066 to the key.

4.4 Session Parameters

The "session" parameters are OPTIONAL and MAY override SRTP session defaults for the SRTP and SRTCP streams. These parameters configure an RTP session for SRTP services.

4.4.1 SSRC=n

The value n is an integer in the range of 0..2^32-1 for the RTP SSRC parameter, which is undefined by default. This is the RTP SSRC of

the sender of the SDP message. If n is invalid, the receiver MUST NOT participate in the media stream but SHOULD log an "invalid SSRC" condition.

SSRC MAY be specified when the setting of the "application" parameter is "srtp" or "both." Otherwise the receiver MUST NOT participate in the media stream and SHOULD log an "invalid session parameter" condition.

4.4.2 ROC=n

The value "n" is an integer in the range of 0..2^32-1 for the SRTP rollover counter (ROC), which is zero by default. The ROC MAY be set to a non-zero value for an ongoing RTP/SAVP stream in which the SRTP ROC has cycled one or more times [srtp]. The receiver of the SDP message SHOULD refresh the ROC value before joining a session "late." How "late" is defined depends on the rate of the particular RTP stream and the time that has elapsed since its commencement. 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 SDP message. If n is invalid, then the receiver MUST NOT participate in the media stream but SHOULD log an "invalid ROC" condition.

ROC MAY be specified when the setting of the "application" parameter is "srtp" or "both." Otherwise the receiver MUST NOT participate in the media stream and SHOULD log an "invalid session parameter" condition.

4.4.3 KEY_DERIVATION_RATE=n

The value n may be an integer in the set {1,2,4,...,2^24}, i.e. a power of 2 between 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.

Key_Derivation_Rate MAY be specified when the "application" parameter setting is "srtp" or "both". Otherwise the receiver MUST NOT participate in the media stream and SHOULD log an "invalid session parameter" condition.

4.4.4 UNENCRYPTED

This indicates that the SRTP or SRTCP stream is not encrypted. SRTP and SRTCP messages are encrypted by default.

UNENCRYPTED MAY be specified for "srtp", "srtcp", or "both". If the a=crypto "application" setting is "both," then both the SRTP and

SRTCP streams are unencrypted.

4.4.5 FEC_ORDER=order

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 on the sender and after SRTP processing on the receiver; FEC_SRTP is the default. SRTP_FEC is the reverse processing. SPLIT signals that SRTP encryption occurs on the sender, followed by FEC processing, followed by SRTP authentication; processing is reversed on the receiver. If the receiver cannot recognize the order value, then the receiver MUST NOT participate in the media stream but SHOULD log an "invalid FEC_ORDER" condition.

If specified, it MUST only be specified with "srtp" or "both." Otherwise the receiver MUST NOT participate in the media stream and SHOULD log an "invalid session parameter" condition.

4.4.6 UNAUTHENTICATED

This parameter signals that SRTP messages are not authenticated. SRTP authenticates SRTP messages by default (see Security Considerations).

If specified, it MUST only be specified with "srtp", or "both" since it applies only to the SRTP stream: Authentication is mandatory for secure RTCP. If UNAUTHENTICATED appears in an a=crypto with an "srtcp" application parameter, the receiver MUST NOT participate in the media stream and SHOULD log an "invalid session parameter" condition.

5.0 Use with Offer/Answer

A receiver that does not recognize a=crypto and assumes default SRTP parameters might receive a stream that uses non-default parameters, which will cause that receiver to fail. An Offer/Answer capabilities exchange, however, allows sender and receiver to agree on parameters before commencement of the multimedia session. The sender and receiver use an Offer/Answer exchange [RFC3264] to match cryptographic capabilities.

This section discusses Offer/Answer exchanges only for the RTP/SAVP (SRTP). A future revision of this document will consider Offer/Answer exchanges for security descriptions in general.

5.1 Offerer Processing

On each SDP media line (m=) where the <transport> is "RTP/SAVP", the offerer MAY follow that media line with at least one a=crypto line. The lines are specified in preference order, with the most preferred listed first. The offerer determines the crypto parameters based on

its capabilities and its security policies.

To specify a list of preferred crypto suites for RTP, RTCP, or both, the offerer includes separate a=crypto lines, in preference order. Each offer is grouped. If separate rtp and rtcp keys are wanted, the srtp a=crypto line MUST be sent first, and both the RTP and RTCP keys MUST always be sent, even if the endpoint is recvonly.

The offerer obtains keying material or the uri pointing to a keyserver by means that are outside the scope of this specification (e.g. the offerer could generate the material or request the material from a third party).

5.2 Answerer Processing

For each SDP media line where the <transport> is RTP/SAVP and the stream is bi-directional stream will be created, the answer MUST include a media line with its <transport> set to RTP/SAVP in order to accept the offer; otherwise, the offer is rejected for that media entry.

The answerer examines the a=crypto lines, in order. If the a=crypto line indicates the application is rtp, and the line immediately following indicates the application is rtcp, the receiver groups these two lines together; otherwise, this group is ignored as it is syntactically invalid. If the a=crypto line indicates the application is "both", it is grouped by itself.

After grouping, the answerer selects the first group of a=crypto that it supports, considering the answerer's capabilities and its security policies.

After selecting one group, the answerer obtains keys appropriate for the selected crypto algorithm(s). The key MUST have the same key length and salt length as the offer.

To set up the bi-directional media with <transport> set to RTP/SAVP, the answerer includes one or two a=crypto lines after the media line. If the offer indicated separate keys for RTP and RTCP, the answer MUST do the same.

5.3 Non-RTP/SAVP Answerers

If a media line with <transport> set to RTP/SAVP is sent to a device that doesn't suppport RTP/SAVP, that media line will not be processed.

If an offerer wants to interoperate with such a device, albeit without the benefits of SRTP or SRTCP, the offerer MUST include an additional media line with <transport> set to RTP/AVP, and other values in that line MUST match the values of the associated

"RTP/SAVP" media line. This second media line would be specified after all of the attribute (a=) lines for the RTP/SAVP media

5.4 Offer/Answer Example: Receiver Supports SRTP

In this example, the offerer supports two crypto suites (F8 and AES). When presented with multiple "a=crypto" lines for an "m=" line, the answerer will chose the first crypto suite that it supports, and the answerer MUST reply with only one "a=crypto" line per "m=" line

Offerer transmits:

```
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 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/SAVP 0
a=crypto:AES_CM_128_HMAC_SHA1_80 both
  inline:16/14/WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz/20/1:32
a=crypto:F8_128_HMAC_SHA1_80 both
  inline:16/14/MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm/20/1:32
```

Answerer transmits:

```
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 224.2.17.11/128
t=2873397526 2873405696
a=sendonly
m=audio 32640 RTP/SAVP 0
a=crypto:F8_128_HMAC_SHA1_80 both
  inline:16/14/MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm/20/1:32
```

In this case, the session would use the F8_128_HMAC_SHA1_80 crypto suite for the RTP and RTCP traffic it generates.

5.5 Offer/Answer Example: Different SRTP and SRTCP keys

In this example, the offerer requests use of one crypto suite for SRTP (AES) and a different crypto suite for RTCP.

Offerer transmits:

```
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 224.2.17.12/127
    t=2873397496 2873404696
    a=recvonly
    m=audio 49170 RTP/SAVP 0
    a=crypto:AES_CM_128_HMAC_SHA1_80 rtp
      inline:16/14/WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz/20/1:32
    a=crypto:F8_128_HMAC_SHA1_80 rtcp
      inline:16/14/MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm/20/1:32

  Answerer transmits:
    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 224.2.17.11/128
    t=2873397526 2873405696
    a=sendonly
    m=audio 32640 RTP/SAVP 0
    a=crypto:AES_CM_128_HMAC_SHA1_80 rtp
      inline:16/14/8NxJiu9zZW1jdGwgKCkgewkyMjA7fQp9CnVubTnC/20/1:32
    a=crypto:F8_128_HMAC_SHA1_80 rtcp
      16/14/c2VtZ2V0KSkgewogICAgc3V/20/1:32
```

5.6 Use of a=crypto With Active Media Streams

When an active SRTP session is rekeyed, this is indicated by sending a new SDP. Rekey MUST NOT be done with an Offer/Answer exchange, but rather as a unidirectional SDP transmission.

When the Offerer needs to rekey, the Offerer MUST only send a=crypto lines which match a=crypto lines it had received in the Answer.

When an SRTP or SRTCP transmitter needs to rekey, the only new values permitted in the a=crypto line(s) are the new key and new salt -- other cryptographic parameters MUST NOT be changed.

If the Answerer selected a=crypto lines using the "inline" method, the exact same a=crypto line(s) as agreed to in the Answer MUST be sent and a new new inline key MUST be sent.

If the Answer selected a=crypto lines using the "uri" method, the sender MAY transmit the same uri, and the recipient MUST re-fetch the uri.

6.0 Security Considerations

One needs to define SDP security descriptions for a specific SDP media transport for a=crypto to be useful. The definitions SHOULD be specified in an Internet RFC, which has security implications that MUST be considered in the RFC. This section considers the SRTP descriptions for the RTP/SAVP transport as specified in this Internet Draft.

6.1 Authentication of packets

RTP messages are vulnerable to a variety of attacks such as replay and forging. To limit these attacks, SRTP message integrity and anti-replay mechanisms SHOULD be used. Source authentication of unicast SRTP messages SHOULD be performed [srtp]. Note that SRTP source-message authentication does not authenticate the IP-address of the SRTP source, but ensures that the SRTP message that the SRTP receiver had received is exactly what the SRTP sender had sent. Source authentication of multicast SRTP messages is today non-standard and hence for further study. Use of the UNAUTHENTICATED parameter therefore, is NOT RECOMMENDED. SRTP supports this setting, however, for voice applications where authentication is implicit in the application [srtp]. In general, applications SHOULD NOT set UNAUTHENTICATED.

6.1 Reuse of Keying Material ("Two-Time Pad")

Misconfigured SRTP sessions, moreover, are vulnerable to attacks on their encryption services when running crypto suites of Sections 4.1.1, 4.1.2 and 4.1.3. An SRTP encryption service is "mis-configured" 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 make them unique, which is violated in the case of SSRC collision: RTP SSRC collision reveals SRTP or SRTCP plaintext 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, as this document requires (Section 4.2). There is risk of attack, however, when an SDP media stream has an "a=sendrecv" direction attribute and a pair of senders are sharing a master key for their encryption (i.e. a weaker condition than sharing a master key). It is RECOMMENDED, therefore, that a sendrecv stream have two SRTP master keys, one for each directional stream. By implication, the SDP message that describes the sendrecv stream MUST NOT be a multicast media stream that provides inline keys from multiple receivers. For the same reason, the risk recurs when a media stream

has an "a=sendonly" direction attribute in an multicast SDP message.

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

6.2 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 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 set 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.

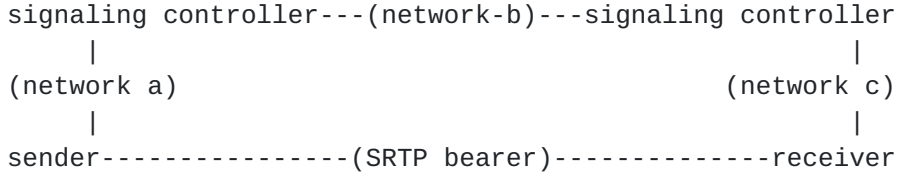
IPsec, TLS, S/MIME or some other data security service SHOULD be used to provide message authentication for SDP messages that carry the SRTP attribute. Message encryption SHOULD be used when a master key parameter appears in the message. Failure to encrypt the SDP message containing an inline SRTP master 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. The proposed SDP key-mgmt statement [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 endpoint, its signaling controller, and the 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 is provided the contents of the SRTP attribute 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 MUST 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).

7.0 Grammar

This section provides an Augmented BNF grammar for the SRTP profile of the SDP crypto-attribute. ABNF is defined in [RFC2234]. The rule names <att-field> and <att-value> are from SDP [RFC2327].

```
att-field = "crypto"
att-value = cipher application

application = cipher-both / cipher-srtp / cipher-srtcp

cipher-both  = key-parameter *[SP sess-par-both]
cipher-srtp  = key-parameter *[SP sess-par-srtp]
cipher-srtcp = key-parameter *[SP sess-par-srtcp]

key-parameter = method-inline / method-uri

cipher  = "AES_CM_128_HMAC_SHA1_32" /
          "F8_128_HMAC_SHA1_32"     /
          "AES_CM_128_HMAC_SHA1_80"

sess-par-both = roc /             ; Roll-Over Counter
                kdr /             ; Key Derivation Rate
                "UNENCRYPTED"

sess-par-srtp  = ssrc / fec-order
```

```
     sess-par-srtcp = "UNAUTHENTICATED"
```

```
method-inline = "inline:" key-info
method-uri = "uri:<" absoluteURI ">"    ; absoluteURI defined in
                                        ; [RFC2396]

key-info = "/" key-length "/" salt-length "/" key-salt
           "/" [lifetime] "/" [mki]

key-length = 1*(DIGIT)          ; length in bytes
salt-length = 1*(DIGIT)         ; length in bytes
key = 1*(base64)
salt = 1*(base64)
key-salt = key salt             ; key and salt concatenated and
                                ; then base64 encoded [section
                                ; 6.8 of RFC2045]
lifetime = ["2^"] 1*(DIGIT)
mki = "/" mki-length ":" mki-value
mki-length = 1*3(DIGIT)         ; MUST be multiple of 8 and
                                ; MUST not exceed 128
mki-value = 1*(base64)

roc = "ROC:" 1*(DIGIT)
kdr = "KDR:" 1*(DIGIT)

ssrc = "SSRC:" ssrc-value
ssrc-value = 1*(DIGIT) *["," ssrc-value]

fec-order = "FEC:" 1*(DIGIT)

base64 =  ALPHA / DIGIT / "+" / "/" / "="
```

8.0 Acknowledgements

This document benefited from discussions with Flemming Andreasen, 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 recommended the generic approach to the SDP media-stream security descriptions that is followed in this draft. Flemming Andreasen suggested some changes to an earlier draft that greatly simplify this document. David McGrew suggested the conservative approach of using unique master keys for each SDP media stream as followed in this document. Jonathan Rosenberg suggested reducing the complexity by specifying only one security parameter for each media stream.

9.0 Authors' Addresses

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

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134 USA
dwing@cisco.com
+1 408 525 5314

10.0 References

[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.

[keymgt] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, "Key Management Extensions for SDP and RTSP", June 2002, http://search.ietf.org/internet-drafts/draft-ietf-mmusic-kmgmt-ext-06.txt, Work in Progress

[mikey] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, "MIKEY: Multimedia Internet KEYing", July 2002, http://search.ietf.org/internet-drafts/draft-ietf-msec-mikey-06.txt, Work in Progress

[RFC1889] H. Schulzrinne, S. Casner, R. Fredrick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", January 1996, http://www.ietf.org/rfc/rfc1889.txt

[RFC2045] N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", November 1996, http://www.ietf.org/rfc/rfc2045.txt

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

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

[RFC2327] M. Handley, V. Jacobson, "SDP: Session Description

Protocol", April 1998, http://www.ietf.org/rfc/rfc2327.txt

[RFC2828] R. Shirey, "Internet Security Glossary", May 2000, http://www.ietf.org/rfc/rfc2828.txt

[RFC2396] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", August 1998, http://www.ietf.org/rfc/rfc2396.txt

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

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

[srtp] M. Baugher, R. Blom, E. Carrara, D. McGrew, M. Naslund, K. Norrman, D. Oran, "The Secure Real-time Transport Protocol", June 2002, http://search.ietf.org/internet-drafts/draft-ietf-avt-srtp-05.txt, Work in Progress

11.0 Full Copyright Statement