Internet Engineering Task Force
MMUSIC Working Group
INTERNET-DRAFT
EXPIRES: March 30, 2003

Mark Baugher
Cisco Systems

September 30, 2002

SDP Security Descriptions for Media Streams

<draft-baugher-mmusic-sdpmediasec-00.txt>

Status of this memo

Abstract

This Internet Draft gives a generic 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, which often include Real-time Transport Protocol (RTP) streams. When run under the RTP/SAVP profile, 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 to configure SRTP beyond using defaults values. This Internet Draft specifies an SDP attribute to signal a cryptographic

key and other parameters for SRTP and other SDP media streams.

Thus, the SDP crypto attribute provides generic security descriptions for SDP media streams. In addition to RTP, the crypto attribute MAY be applied to white board, modem, fax, and other media

that could use various security protocols such as IPsec or SSL. Each SDP media transport, however, needs its own definitions that assign values to crypto-attribute parameters, which SHOULD be specified in an Internet RFC. This Internet Draft is intended to be THE standards-track RFC that defines the parameter values for SRTP. With this I-D, 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 as SRTP secures RTP messages or IPsec secures IP packets. Data security protocols such as SRTP rely upon an external 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. This service 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, S/MIME 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 informs the receiver of 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.

Several a=crypto parameters are generic to all media transports, but their values MAY be unique to a particular transport. Section 3.0 specifies the SDP crypto attribute generically. Section 4.0 defines the crypto attribute for SRTP. Section 5.0 discusses use of the crypto attribute in Offer/Answer exchanges. Section 6.0 recites security considerations.

3.0 SDP Media Security Descriptions

A new 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: key_parameters *<security_session parameters>

The next sections describe the cryptographic "key parameters" and explains the optional "security_session parameters."

3.1 Cryptographic Key Parameters

There are four "key_parameters.

1. transport=transport_descriptor
   Exactly one transport= MUST appear in a=crypto. The "transport_descriptor" is the transport value in an m= line. For example, "transport=RTP/SAVP" describes the crypto attribute as an SRTP crypto attribute.

2. format=format_descriptor
   Zero or more format= parameters MAY appear in a=crypto. The "format descriptor" is the format value in an m= line. For example, "format=97" associates a=crypto with dynamic payload 97 from an a=rtpmap description.

3. crypto_suite=value
   Zero or one crypto_suite= parameter MAY appear in a=crypto. The "value" is the authentication and encryption transforms that are applied to the media stream and is specific to the m= transport type. Section 4.0 lists crypto_suite values for RTP/SAVP, the SRTP media transport type.

4. mkey=(method) value
   Zero or one mkey= MAY appear in a=crypto to install a master key. "method" is either "uri" or "srtp." The latter's "value" is an SRTP master key. And the former "value" is a Uniform Resource Identifier value; the URI is a resource that SHOULD be queried to obtain the master key for the session. As SDP descriptions for new media-stream transports are defined in the future, new methods (e.g. "SRTP") SHOULD be defined in an Internet RFC. The mkey contains a random value that MUST be unique with respect to other mkey lines in the SDP message. Section 4.0 lists mkey values for RTP/SAVP, the SRTP media transport type.

Thus, the crypto attribute describes a cryptographic key and other parameters for a transport type that appears in an m= line.

3.2 Security-Session Parameters

There are no generic security-session parameters; these are specific to a particular transport (see Section 4.0).

3.3 Examples

The first example shows a=crypto for the RTP/SAVP transport type.

```
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: transport=RTP/SAVP crypto_suite=AES_CM_128_HMAC_SHA1_80
  mkey=(srtp)/16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/20/1:32
m=audio 49170 RTP/SAVP 0
a=crypto: transport=RTP/SAVP
  mkey=(srtp)/16/14/NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj/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 has a "AES_CM_128_HMAC_SHA1_80" crypto_suite; its mkey parameter carries the SRTP master key data and metadata. The m=audio media entry uses the default "crypto_suite=AES_CM_128_HMAC_SHA1_32." These are RTP/SAVP (SRTP) specific and defined in the next section.

The a=crypto MAY identify a media stream with a format= in addition to a transport=.

```
m=audio 49230 RTP/SAVP 96 97 98
a=rtpmap:96 L8/8000
a=rtpmap:97 L16/8000
a=rtpmap:98 L16/11025/2
a=crypto: transport=RTP/SAVP format=98 format=97
   mkey=(uri)"https://keyserver.com/SDPSeminar/"
a=crypto: transport=RTP/SAVP format=96 crypto_suite=null
```

This example describes SRTP services for RTP payload types L16/8000 and L16/11025/2. SRTP default values are implicitly signaled by the absence of crypto_suite and mkey parameters. Media format 96, however, does not use SRTP services because the RTP/SAVP crypto_suite is null (see Section 4.0) for this media stream.

4.0 SRTP Media Security Descriptions

The generic SDP media security descriptions of the preceding section

need parameter values to be defined for specific media transports; this section defines needed crypto attribute values and parameters for the RTP/SAVP 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; SRTP supports the addition of new transforms through the publication of a new Internet RFC that specifies default and mandatory values for the SRTP parameters. At the time of multimedia 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 for SDP a=crypto follows.

SDP SRTP Parameter	Description
CRYPTO_SUITE	Encryption and authentication transforms
MKEY	Master key, salt and related parameters
SSRC	Source of data to an RTP session
ROC	Roll-over counter
KEY_DERIVATION_RATE	Rate that the pseudo-random function (PRF) is applied to a key
ENCRYPTED_SRTCP	SRTCP messages are encrypted
UNENCRYPTED_SRTP	SRTP messages are not encrypted
UNAUTHENTICATED_SRTP	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 [p.32, srtp]. The CRYPTO_SUITE and MKEY values belong to the crypto_suite and mkey parameters of the SDP crypto attribute (Section 3.0). These are defined in the next section and are followed by the SRTP security-session parameters. In all cases, if a receiver cannot recognize a parameter or value outside of an Offer/Answer exchange (see Section

5.0), then the receiver MUST NOT participate in the media stream and SHOULD log an "invalid name" condition.

4.1 CRYPTO_SUITE=crypto_suite

The crypto_suite value defaults to AES_CM_128_HMAC_SHA1_32 but MAY be set to other valid crypto suites as defined below. There are no means to set CRYPTO_SUITE to different values for SRTP and SRTCP. If a receiver does not support the particular crypto_suite outside of an Offer/Answer exchange (see Section 5.0), then the receiver MUST NOT participate in the media stream and SHOULD log an "unrecognized crypto_suite" condition.

4.1.1 CRYPTO_SUITE=AES_CM_128_HMAC_SHA1_32

This is the SRTP default AES Counter Mode cipher and 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. 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. Please review the Security Considerations section concerning keystream issues for group keys defined by an SDP direction attribute and multicast issues.

4.1.2 CRYPTO_SUITE=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. 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. Please review the Security Considerations section concerning keystream issues for group keys defined by an SDP direction attribute and multicast issues.

4.1.3 CRYPTO_SUITE=AES_CM_128_HMAC_SHA1_80

The SRTP AES Counter Mode cipher is used with HMAC-SHA1 message authentication having an 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. 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. Please review the Security Considerations section concerning keystream issues for group keys defined by an SDP direction attribute and multicast issues.

4.1.4 CRYPTO_SUITE=NULL

No encryption or authentication are applied to SRTP or SRTCP. This

effectively disables all SRTP services for the RTP/SAVP media stream.

4.1.5 Adding new CRYPTO_SUITE definitions

As new transforms are added to SRTP, new definitions SHOULD be given for the SDP crypto attribute and published in a 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 override defaults used in definitions 4.1.1 through 4.1.4. For example, if an application needed to vary the size of the master or session salt key for any of the defined crypto suites, a new crypto suite SHOULD be defined in a Internet RFC that specifies the chosen size of the master or session salt key length.

4.2 MKEY=srtp_mkey

The "srtp_mkey" has the following structure ("||" is the concatenate operator).

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

The "key_length" is the length of the master key, and "salt_length" is the length of the master salt. 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 mkey value.

The third part of the srtp_mkey structure is the cryptographic master key appended with the master salt. Each (master) key and salt MUST be a random number and MUST be unique to the SDP message. Both are base64 encoded (following concatenation). If the length of the concatenated keys (after being decoded from base64) 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 "mkey too short" condition. The "key||salt" value MUST appear as part of the srtp_mkey.

The fourth part of the srtp_mkey is the OPTIONAL lifetime of the master key as measured in number of packets encrypted or authenticated with that key. The default value is 48, which is 2^48 packets encrypted with a master key according to the SRTP standard [srtp]. Thus, "lifetime" is specified as a power of two when present and MUST NOT exceed the maximum packets lifetime for the crypto_suite (e.g. 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 fields of an SDP session-level mkey; otherwise, the receiver MUST NOT participate in the media stream and SHOULD log an "invalid mkey" condition.

The MKI value is OPTIONAL as is its specified bit length. "MKI" is the master key index associated with the srtp_mkey. 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 and MUST be a positive multiple of 8. If the MKI_length is not given or if it exceeds 128 bits, 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.

4.3 SRTP Security-Session Parameters

SRTP security descriptions apply to sessions that include a pair of RTP and RTCP streams; the "security-session parameters" configure these sessions for SRTP services. The following parameters are OPTIONAL and MAY override SRTP session defaults for the SRTP or SRTCP streams.

4.3.1 SSRC=n

The value n is an integer in the range of 0..2^32-1 for the RTP SSRC parameter. SSRC is undefined by default. If n is invalid, the receiver MUST NOT participate in the media stream but SHOULD log an "invalid SSRC" condition.

4.3.2 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.

4.3.3 ENCRYPTED_SRTCP

This parameter signals that SRTCP messages are encrypted. SRTP does not encrypt SRTCP messages by default.

4.3.4 UNENCRYPTED_SRTP

This parameter signals that SRTP messages are not encrypted. SRTP encrypts SRTP messages by default.

4.3.5 UNAUTHENTICATED_SRTP

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

4.3.6 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.

5.0 Use with Offer/Answer

Apart from an Offer/Answer exchange, a sender of an SDP a=crypto description cannot determine if a receiver correctly processed a=crypto, or if that receiver is likely to fail when receiving an RTP/SAVP media stream that does not use SRTP defaults. An Offer/Answer exchange is the remedy that assures the SDP sender of a receiver's capabilities. Offer/Answer exchange capability is implicitly supported in this I-D since the crypto attribute is associated with a media entry - the subject of the Offer/Answer exchange [RFC3264]. Thus, a receiver implicitly accepts or rejects the crypto description when it accepts or rejects the media description in an Offer/Answer exchange.

It is complex, however, to negotiate cryptographic parameters concomitantly with media codecs or other media parameters: Without special processing of a=crypto, the Offer/Answer complexity is on the order of the cross product of the number of crypto attributes and codecs that are offered. Thus, if a media entry has three possible codecs in a one-of-n codec Offer and has two a=crypto alternatives for each, there MUST be six a=rtpmap lines instead of three.

```
v=0
o=carol 28908764872 28908764872 IN IP4 100.3.6.6
s=-
t=0 0
c=IN IP4 192.0.2.4
a=sendonly
m=audio 62986 RTP/SAVP 0 1 3 97 98 99

a=rtpmap:0 PCMU/8000
a=crypto: transport=RTP/SAVP format=0
  mkey=(srtp)/16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/20/1:32
a=rtpmap:97 PCMU/8000
a=crypto: transport=RTP/SAVP format=97
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/NXZXeik3K2BsV2NJcTtTYl0nYVFheWl2XkdLPnwv/20/1:32

a=rtpmap:1 1016/8000
a=crypto: transport=RTP/SAVP format=1
  mkey=(srtp)/16/14/bFcmQFZOanM7P3olOXpJQndmTzcjXz19WG1xODdi/20/1:32
a=rtpmap:98 1016/8000
a=crypto: transport=RTP/SAVP format=98
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/amlAKWt3KnpqZSR9PzFrRG0kSXNCdmk4ISw+XS1N/20/1:32

a=rtpmap:3 GSM/8000
a=crypto: transport=RTP/SAVP format=3
  mkey=(srtp)/16/14/I21dQClsTndvRDAkP0NBd18rWztKJThnMkJWbS48/20/1:32
a=rtpmap:99 GSM/8000
a=crypto: transport=RTP/SAVP format=99
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/cndOTUBMT0k0aWtCQDBMbmxlIzA5OE4jbEp6PX0u/20/1:32
```

As shown in the example, three distinct formats are offered for the m=audio media entry; the crypto_suite default for a=crypto is replaced for format descriptors 97, 98 and 99. Thus, six a=rtpmap lines are needed to enumerate a pair of a=crypto alternatives for the Offer/Answer exchange.

It's possible to reduce the number of codec offers by having a=crypto be explicitly offered in an Offer/Answer exchange. Multiple a=crypto attributes MAY be offered for a media stream and MUST appear in order of preference in a media entry: The first a=crypto in a media entry is most preferred and the last a=crypto is the least preferred. Like any Offer, a crypto Offer MAY be rejected using the mechanisms of the higher-layer protocol. Thus, zero, one or more a=crypto offers MAY be returned in the Answer. An example Offer in an Offer/Answer capabilities exchange is shown below.

```
v=0
o=carol 28908764872 28908764872 IN IP4 100.3.6.6
s=-
t=0 0
c=IN IP4 192.0.2.4
a=sendonly
m=audio 0 RTP/SAVP 0 1 3
a=rtpmap:0 PCMU/8000
a=crypto: transport=RTP/SAVP format=0
  mkey=(srtp)/16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/20/1:32
a=crypto: transport=RTP/SAVP format=0
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/cndOTUBMT0k0aWtCQDBMbmxlIzA5OE4jbEp6PX0u/20/1:32

a=rtpmap:1 1016/8000
a=crypto: transport=RTP/SAVP format=1
  mkey=(srtp)/16/14/NXZXeik3K2BsV2NJcTtTYl0nYVFheWl2XkdLPnwv/20/1:32
a=crypto: transport=RTP/SAVP format=1
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/amlAKWt3KnpqZSR9PzFrRG0kSXNCdmk4ISw+XS1N/20/1:32

a=rtpmap:3 GSM/8000
a=crypto: transport=RTP/SAVP format=3
  mkey=(srtp)/16/14/I21dQClsTndvRDAkP0NBd18rWztKJThnMkJWbS48/20/1:32
a=crypto: transport=RTP/SAVP format=3
  crypto_suite=aes_cm_128_hmac_sha1_80
  mkey=(srtp)/16/14/bFcmQFZOanM7P3olOXpJQndmTzcjXz19WG1xODdi/20/1:32
```

In this example, the Answerer selects one of two a=crypto lines by returning only one or prioritizing one over the other. Alternatively, by permitting a=crypto to appear at the SDP session level, we get simpler Offers and Answers.

```
v=0
o=carol 28908764872 28908764872 IN IP4 100.3.6.6
s=-
t=0 0
c=IN IP4 192.0.2.4
a=crypto: transport=RTP/SAVP format=1 format=2 format=3
 mkey=(srtp)/16/14/d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj/20/1:32
a=crypto: transport=RTP/SAVP format=1 format=2 format=3
 crypto_suite=aes_cm_128_hmac_sha1_80
 mkey=(srtp)/16/14/bFcmQFZOanM7P3olOXpJQndmTzcjXz19WG1xODdi/20/1:32
m=audio 0 RTP/SAVP 0 1 3
a=rtpmap:0 PCMU/8000
a=rtpmap:1 1016/8000
a=rtpmap:3 GSM/8000
```

SDP session-level crypto is elegant compared to the previous examples. This Offer has two prioritized SDP session-level crypto alternatives for RTP/SAVP streams, which are inoperative in this SDP

message; the session name is "-", the start/duration times are zero and the media stream port is zero. The Answerer selects one and it applies to all media streams that use the default. Thus the mkey is inoperative and can't be used (i.e. the SRTP crypto context [srtp] is not defined). The SDP session-level mkey is merely a template to tell the Answerer the key lengths, lifetime and indexing. It is REQUIRED, however, that the final offer assign an individual mkey to each media stream as REQUIRED by Section 4.2. Thus, there MUST be a crypto attribute with a mkey at the media-entry level for every media stream that gets assigned a mkey.

SDP session-level a=crypto lines MUST NOT appear outside of an Offer/Answer exchange of inactive media streams [RFC3264] owing to the risk of a "two-time pad" situation when a shared, derived session key erroneously produces two identical key streams for two or more media streams. This can happen during RTP SSRC collisions when the unique value used to generate a unique keystream is non-unique among two or more media streams [srtp].

This I-D follows the conservative approach of assigning a unique master key to each media stream (session keys that are derived from distinct master keys will be unique). By prohibiting SDP session-level crypto lines, each media stream is sure to have a unique master key.

An alternative approach to the "two-time pad" problem generates unique labels to ensure unique session keystreams [mikey]. This is for further study and thus is SDP session-level crypto lines outside of an Offer/Answer exchange where the key is inactive and inoperative.

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, which is being proposed as a standards-track RFC.

RTP messages are vulnerable to a variety of attacks such as replay and forging. SRTP message integrity and anti-replay mechanisms, therefore, SHOULD be used. Source authentication of unicast SRTP messages SHOULD be performed. Source authentication of multicast SRTP messages is today non-standard and hence for further study. Use of the UNAUTHENTICATED_SRTP 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_SRTP. Even SRTP confidentiality can be broken in certain circumstances when messages are unauthenticated [Bellovin].

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 "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 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 I-D REQUIRES (Section 4.2). There is risk of attack, however, when an SDP media stream has an "a=sendrecv" direction attribute because this implies that a pair of senders are sharing a master key for their session encryption key; in this case, the SDP message SHOULD also set the a=crypto SSRC parameter (Section 4.3.1) for that media stream. By implication, the SDP message that describes the sendrecv stream MUST NOT be a multicast SDP message, since the crypto SSRC parameter can set an SSRC for only one receiver. For the same reason, the risk recurs when a media stream has an "a=sendonly" direction attribute in an multicast SDP message. Thus, a multicast SDP message MUST NOT use a crypto attribute for a media stream that has a direction attribute of a=sendrecv or a=sendonly. There is no risk of sending SRTP and SRTCP using a single master key for recvonly, sendonly, or sendrecv media streams. These rules are essential for correct configuration and secure operation of SRTP cipher suites 4.1.1, 4.1.2 and 4.1.3.

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_SRTP in an SDP session level Offer, this could result in a receiver not decrypting the encrypted SRTP messages. In the worst case, the receiver 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 mkey parameter appears in the message. Failure to encrypt the SDP message containing an SRTP 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/or authenticated SDP message, it is RECOMMENDED that a key management protocol be used. The proposed SDP key-mgmt statement allows authentication and encryption of the key management protocol data independently of the SDP message that carries it [keymgt]. 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. In this case, the third-party provider is privy to 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 Acknowledgements

This work benefited from discussions with David McGrew, Mats Naslund, Mike Thomas, Elisabetta Cararra, Brian Weis, Dave Oran, Flemming Andreasen, 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 I-D. David McGrew suggested the conservative approach of using unique master keys for each SDP media stream as followed in this I-D.

8.0 Author's Address

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

9.0 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-05.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-03.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

[RFC2104] H.Krawczyk, M.Bellare, R.Canetti, HMAC: Keyed-Hashing for Message Authentication, November 1997, ftp://ftp.isi.edu/in-notes/rfc2104.txt

[RFC2327] M.Handley, V.Jacobson, SDP: Session Description Protocol, April 1998, http://www.ietf.org/rfc/rfc2327.txt

[RFC3264] J.Rosenberg, H.Schulzrinne, An Offer/Answer Model with the Session Description Protocol (SDP), June 2202, ftp://ftp.isi.edu/in-notes/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