Network Working Group
Internet-Draft
Expires: November 26, 2006

F. Audet
Nortel
D. Wing
Cisco Systems

May 25, 2006

Evaluation of SRTP Keying with SIP
draft-wing-srtp-keying-eval-00

Status of this Memo

Copyright Notice

Abstract

Over a dozen incompatible mechanisms have been defined to key an Secure RTP (SRTP) media stream. This document evaluates the keying mechanisms, concentrating on their interaction with SIP features and their security properties.

This document is discussed on the rtpsec mailing list, <http://www.imc.org/ietf-rtpsec>.

Table of Contents

1. Introduction

SIP needs to operate across the world-wide public Internet and thus needs a single, mandatory-to-implement mechanism for strongly authenticating an endpoint. It is likely that the mechanism will be based on RSA, Diffie-Hellman, or Digital Signature Standard (DSS) but cannot rely on an X.509 PKI or pre-shared keys.

There are currently 13 mechanisms defined or under consideration by the IETF to establish SRTP [RFC3711] keys between endpoints. Although an endpoint can implement several mechanisms, these 13 mechanisms are not interoperable with each other. The mechanisms can be broken into three general categories for exchanging SRTP keying: exchanging keys in signaling, media, or both.

The goals of an SRTP key exchange mechanism are, in rough order:

1. Ability to deploy the mechanism across administrative boundaries, such as on the Internet,
2. Cryptographically authenticate the endpoints,
3. Securely exchange SRTP keys,
4. Support SIP features such as retargeting and forking.

Existing key exchange mechanisms fail to meet all of these requirements.

Two mechanisms, MIKEY and Security Descriptions, have been standardized for SRTP key exchange. Both of these mechanisms perform key exchange in the signaling path (SIP or RTSP).

All MIKEY modes share a common syntax (a=key-mgmt, defined in Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP) [I-D.ietf-mmusic-kmgmt-ext]). The base MIKEY specification [RFC3830] defines four MIKEY modes and additional modes are defined in other specifications. MIKEY modes are not compatible with each other.

The other standard mechanism, Security Descriptions, uses a different syntax (a=crypto, defined in Security Descriptions [I-D.ietf-mmusic-sdescriptions]).

Several extensions to MIKEY have been proposed and several techniques which perform some, or all, keying in the media path have been proposed. These new techniques are also discussed in this document.

Out of scope of this document is how SIP, RTSP, and SDP messages themselves are encrypted.

Call signaling (new call, end of call, call transfer, etc.) is done in SIP, and media is sent in RTP. In the following diagram, Alice is calling Bob. This causes Alice to emit a SIP message to her SIP proxy, which processes the message and routes the message to Bob's proxy which then routes it to Bob.

```
                    +---------+     SIP Invite    +-------|
                    | Alice's +------------------>+ Bob's |
                    |  proxy  |                   | proxy |
                    +----+----+                   +---+---|
                         ^                            |
                SIP Invite |                          | SIP Invite
                         |                            V
                     +---+---+                     +-----+
                     | Alice |<===================>+ Bob |
                     +-------+        SRTP         +-----+
```

Figure 1: Simplified SIP Model

2. Terminology

AOR (Address-of-Record): A SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI where the user might be available. Typically, the location service is populated through registrations. An AOR is frequently thought of as the "public address" of the user.

SSRC: The 32-bit value that defines the synchronization source, used in RTP. These are generally unique, but collisions can occur.

two-time pad: The use of the same key and the same key index to encrypt different data. For SRTP, a two-time pad occurs if two senders are using the same key and the same RTP SSRC value.

3. Overview of Keying Mechanisms

Based on how the SRTP keys are exchanged, each SRTP key exchange mechanism belongs to one general category:

signaling path:
All the keying is carried in the call signaling (SIP or SDP) path.

media path:
All the keying is carried in the SRTP/SRTCP media path, and no signaling whatsoever is carried in the call signaling path.

signaling and media path:
Parts of the keying are carried in the SRTP/SRTCP media path, and parts are carried in the call signaling (SIP or SDP) path.

One of the significant benefits of SRTP over other end-to-end encryption mechanisms, such as for example IPsec, is that SRTP is bandwidth efficient and SRTP retains the header of RTP packets. Bandwidth efficiency is vital for VoIP in many scenarios where access bandwidth is limited or expensive, and retaining the RTP header is important for troubleshooting packet loss, delay, and jitter.

Related to SRTP's characteristics is a goal that any SRTP keying mechanism to also be efficient and not cause additional call setup delay. Contributors to additional call setup delay include network or database operations: retrieval of certificates and additional SIP or media path messages, and computational overhead of establishing keys or validating certificates.

When examining the choice between keying in the signaling path, keying in the media path, or keying in both paths, it is important to realize the media path is generally 'faster' than the SIP signaling path. The SIP signaling path has computational elements involved which parse and route SIP messages. The media path, on the other hand, does not normally have computational elements involved, and even when computational elements such as firewalls are involved, they cause very little additional delay. Thus, the media path can be useful for exchanging several messages to establish SRTP keys.

3.1. Signaling Path Keying Techniques

3.1.1. MIKEY-NULL

MIKEY-NULL [RFC3830] has the offerer indicate the SRTP keys for both directions. The key is sent unencrypted in SDP, which means the SDP must be encrypted hop-by-hop (e.g., by using TLS (SIPS)) or end-to-end (e.g., by using S/MIME).

MIKEY-NULL requires one message from offerer to answerer (half a round trip), and does not add additional media path messages.

3.1.2. MIKEY-PSK

MIKEY-PSK (pre-shared key) [RFC3830] requires that all endpoints share one common key. MIKEY-PSK has the offerer encrypt the SRTP keys for both directions using this pre-shared key.

MIKEY-PSK requires one message from offerer to answerer (half a round trip), and does not add additional media path messages.

3.1.3. MIKEY-RSA

MIKEY-RSA [RFC3830] has the offerer encrypt the keys for both directions using the intended answerer's public key, which is obtained from a PKI.

MIKEY-RSA requires one message from offerer to answerer (half a round trip), and does not add additional media path messages. MIKEY-RSA requires the offerer to obtain the intended answerer's certificate.

3.1.4. MIKEY-RSA-R

MIKEY-RSA-R An additional mode of key distribution in MIKEY: MIKEY-RSA-R [I-D.ietf-msec-mikey-rsa-r] is essentially the same as MIKEY-RSA-R but reverses the role of the offerer and the answerer with regards to providing the keys. That is, the answerer encrypts the keys for both directions using the offerer's public key. Both the offerer and answerer validate each other's public keys using a PKI. MIKEY-RSA-R also enables sending certificates in the MIKEY message.

MIKEY-RSA-R requires one message from offerer to answer, and one message from answerer to offerer (full round trip), and does not add additional media path messages. MIKEY-RSA-R requires the offerer validate the answerer's certificate.

3.1.5. MIKEY-DHSIGN

In MIKEY-DHSIGN [RFC3830] the offerer and answerer derive the key from a Diffie-Hellman exchange. In order to prevent an active man-in-the-middle the DH exchange itself is signed using each endpoint's private key and the associated public keys are validated using a PKI.

MIKEY-DHSIGN requires one message from offerer to answerer, and one message from answerer to offerer (full round trip), and does not add additional media path messages. MIKEY-DHSIGN requires the offerer and answerer to validate each other's certificates. MIKEY-DHSIGN also enables sending the answerer's certificate in the MIKEY message.

3.1.6. MIKEY-DHHMAC

MIKEY-DHHMAC [I-D.ietf-msec-mikey-dhhmac] uses a pre-shared secret to HMAC the Diffie-Hellman exchange, essentially combining aspects of MIKEY-PSK with MIKEY-DHSIGN, but without MIKEY-DHSIGN's need for a PKI to authenticate the Diffie-Hellman exchange.

MIKEY-DHHMAC requires one message from offerer to answerer, and one message from answerer to offerer (full round trip), and does not add additional media path messages.

3.1.7. MIKEY-ECIES and MIKEY-ECMQV (MIKEY-ECC)

ECC Algorithms For MIKEY [I-D.ietf-msec-mikey-ecc] describes how ECC can be used with MIKEY-RSA (using ECDSA signature) and with MIKEY-DHSIGN (using a new DH-Group code), and also defines two new ECC-based algorithms, Elliptic Curve Integrated Encryption Scheme (ECIES) and Elliptic Curve Menezes-Qu-Vanstone (ECMQV) .

For the purposes of this paper, the ECDSA signature, MIKEY-ECIES, and MIKEY-ECMQV function exactly like MIKEY-RSA, and the new DH-Group code function exactly like MIKEY-DHSIGN. Therefore these ECC mechanisms aren't discussed separately in this paper.

3.1.8. Security Descriptions

Security Descriptions [I-D.ietf-mmusic-sdescriptions] has each side indicate the key it will use for transmitting SRTP media, and the keys are sent in the clear in SDP. Security Descriptions relies on hop-by-hop (TLS via SIPS) or end-to-end (S/MIME) encryption to protect the keys exchanged in signaling.

Security Descriptions requires one message from offerer to answerer, and one message from answerer to offerer (full round trip), and does not add additional media path messages.

3.1.9. SDP-DH

SDP Diffie-Hellman [I-D.baugher-mmusic-sdp-dh] exchanges Diffie-Hellman messages in the signaling path to establish session keys.

SDP-DH requires one message from offerer to answerer, and one message from answerer to offerer (full round trip), and does not add additional media path messages.

3.2. Media Path Keying Technique

3.2.1. ZRTP

ZRTP [I-D.zimmermann-avt-zrtp] does not exchange information in the signaling path (although it's possible for endpoints to do so if they desire). In ZRTP the keys are exchanged entirely in the media path. The advantage to this mechanism is that the signaling channel is used only for call setup and the media channel is used to establish an encrypted channel -- much like encryption devices on the PSTN. ZRTP uses voice authentication of the DH exchange by having each person read digits to the other person. Subsequent sessions with the same peer can be authenticated using a hash of the previously negotiated key rather than voice authentication.

ZRTP uses 4 media path messages (Hello, Commit, DHPart1, and DHPart2) to establish the SRTP key, and 3 media path confirmation messages. The first 4 are sent as RTP packets (using RTP header extensions), and the last 3 are sent in conjunction with SRTP media packets (again as SRTP header extensions). Note that unencrypted RTP is being exchanged until the SRTP keys are established.

3.3. Signaling and Media Path Keying Techniques

3.3.1. EKT

EKT [I-D.mcgrew-srtp-ekt] relies on another SRTP key exchange protocol, such as Security Descriptions or MIKEY, for bootstrapping. In the initial phase, each member of a conference uses an SRTP key exchange protocol to establish a common key encryption key (KEK). Each member may use the KEK to securely transport its SRTP master key and current SRTP rollover counter (ROC), via RTCP, to the other participants in the session.

EKT requires the offerer to send some parameters (EKT_Cipher, KEK, and security parameter index (SPI)) via the bootstrapping protocol such as Security Descriptions or MIKEY. Each answerer sends an SRTCP message which contains the answerer's SRTP Master Key, rollover counter, and the SRTP sequence number. Rekeying is done by sending a new SRTCP message. For reliable transport, multiple RTCP messages need to be sent.

3.3.2. RTP-DTLS

RTP-DTLS [I-D.fischl-mmusic-sdp-dtls] exchanges public key fingerprints in SDP and then establishes a DTLS session over the media channel [I-D.fischl-sipping-media-dtls]. The endpoints use the DTLS handshake to agree on crypto suites and establish DTLS session keys. Once established, the endpoints use a modified DTLS mode to exchange encrypted media packets on the wire. These encrypted media

packets closely resemble SRTP's on-the-wire format, most importantly by retaining the same RTP header as RTP packets so that header compression and RTP analysis tools can be used. However, these packets are not compatible with SRTP [RFC3711].

The authors of this mechanism have deprecated the mechanism in favor of DTLS-SRTP (Section 3.3.3).

3.3.3. DTLS-SRTP

DTLS-SRTP [I-D.draft-mcgrew-dtls-srtp] exchanges public key fingerprints in SDP and then establishes a DTLS session over the media channel. The endpoints use the DTLS handshake to agree on crypto suites and establish SRTP session keys. SRTP packets are then exchanged between the endpoints.

DTLS-SRTP requires one message from offerer to answerer (half round trip), and, if the offerer wishes to correlate the SDP answer with the endpoint, requires one message from answer to offerer (full round trip). DTLS-SRTP uses 4 media path messages to establish the SRTP key.

This paper assumes DTLS will use TLS_RSA_WITH_3DES_EDE_CBC_SHA as its cipher suite, which is the mandatory-to-implement cipher suite in TLS [RFC4346].

4. Evaluation Criteria - SIP

This section considers how each keying mechanism interacts with SIP features.

4.1. Secure Retargeting and Secure Forking

In SIP, a request sent to a specific AOR but delivered to a different AOR is called a "retarget". A typical scenario is a "call forwarding" feature. In the figure below, Alice sends an Invite in step 1 which is sent to Bob in step 2. Bob responds with a redirect (SIP response code 3xx) pointing to Carol in step 3. This redirect typically does not propagate back to Alice but only goes to a proxy (i.e., the retargeting proxy) which sends the original Invite to Carol in step 4.

```
                          +-----+
                          |Alice|
                          +--+--+
                             |
                             | Invite (1)
                             V
                        +----+----+
                        |  proxy  |
                        ++-+-----++
                         | ^     |
               Invite (2) | |     | Invite (4)
           & redirect (3) | |     |
                         V |     V
                        ++-++   ++----+
                        |Bob|   |Carol|
                        +---+   +-----+
```

Figure 2: Retargeting

Successful use of SRTP requires strongly identifying both calling party and the called party. The mechanism used by SIP for identifying the calling party is SIP Identity [I-D.ietf-sip-identity]. However, due to SIP retargeting issues [I-D.peterson-sipping-retarget], SIP Identity can only identify the calling party (that is, the party that initiated the SIP request). Some key exchange mechanisms predate SIP Identity and include their own identity mechanism. However, those built-in identity mechanism suffer from the same SIP retargeting problem described in the above draft. Going forward, it is anticipated that Connected Identity [I-D.ietf-sip-connected-identity] may allow identifying the called party. In the list below, this is described as the 'retargeting identity' problem.

In SIP, 'forking' is the delivery of a request to multiple locations. This happens when a single AOR is registered more than once. An example of forking is when a user has a desk phone, PC client, and mobile handset all registered with the same AOR.

```
                          +-----+
                          |Alice|
                          +--+--+
                             |
                             | Invite
                             V
                       +-----+-----+
                       |   proxy   |
                       ++---------++
                        |         |
                 Invite |         | Invite
                        V         V
                     +--+--+   +--+--+
                     |Bob-1|   |Bob-2|
                     +-----+   +-----+
```

Figure 3: Forking

With forking, both Bob-1 and Bob-2 might send back SDP answers in SIP responses. Alice will see those intermediate (18x) and final (200) responses. It is useful for Alice to be able to associate the SIP response with the incoming media stream. Although this association can be done with ICE [I-D.ietf-mmusic-ice], and ICE is useful to make this association with RTP, it isn't desirable to require ICE to accomplish this association. The table below analyzes if it is possible for an offerer to associate the media stream with each SDP answer, without using ICE.

Forking and retargeting are often used together. For example, a boss and secretary might have both phones ring and rollover to voice mail if neither phone is answered.

To maintain media security, only the endpoint that answers the call should know the SRTP keys for the session. For key exchange mechanisms that don't provide secure forking or secure retargeting, one workaround is to rekey immediately after forking or retargeting. However, because the originator may not be aware that the call forked this mechanism requires rekeying immediately after every session is established which causes additional signaling messages.

Retargeting securely introduces a more significant problem. With retargeting, the actual recipient of the request is not the original recipient. This means that if the offerer encrypted material (such

as the session key or the SDP) using the original recipient's public key, the recipient will not be able to decrypt that material because the actual recipient won't have the original recipient's private key. In some cases, this is the intended behavior, i.e., you wanted to establish a secure connection with a specific individual. In other cases, it is not intended behavior (you want all voice media to be encrypted, regardless of who answers).

Further compounding this problem is a particularity of SIP that when forking is used, there is always only one final error response delivered to the sender of the request: the forking proxy is responsible for choosing which final response to choose in the event where forking results in multiple final error responses being received by the forking proxy. This means that if a request is rejected, say with information that the keying information was rejected and providing the far end-end's credentials, it is very possible that the rejection will never reach the sender. This problem, called the Heterogeneous Error Response Forking Problem (HERFP) [I-D.mahy-sipping-herfp-fix] is a complicated problem to solve in SIP.

The following list compares the behavior of secure forking, answering association, two-time pads, and secure retargeting for each keying mechanism.

MIKEY-NULL

Secure Forking: No, all AORs see offerer's and answerer's keys. Answer is associated with media by the SSRC in MIKEY. Additionally, a two-time pad occurs if two branches choose the same 32-bit SSRC and transmit SRTP packets.

Secure Retargeting: No, all targets see offerer's and answerer's keys. Suffers from retargeting identity problem.

MIKEY-PSK

Secure Forking: No, all AORs see offerer's and answerer's keys. Answer is associated with media by the SSRC in MIKEY. Note that all AORs must share the same pre-shared key in order for forking to work at all with MIKEY-PSK. Additionally, a two-time pad occurs if two branches choose the same 32-bit SSRC and transmit SRTP packets.

Secure Retargeting: Not secure. For retargeting to work, the final target must possess the correct PSK. As this is likely in scenarios were the call is targeted to another device belonging to the same user (forking), it is very unlikely that other users will possess that PSK and be able to successfully

answer that call.

MIKEY-RSA

Secure Forking: No, all AORs see offerer's and answerer's keys. Answer is associated with media by the SSRC in MIKEY. Note that all AORs must share the same private key in order for forking to work at all with MIKEY-RSA. Additionally, a two-time pad occurs if two branches choose the same 32-bit SSRC and transmit SRTP packets.

Secure Retargeting: No.

MIKEY-RSA-R

Secure Forking: Yes. Answer is associated with media by the SSRC in MIKEY.

Secure Retargeting: Yes.

MIKEY-DHSIGN

Secure Forking: Yes, each forked endpoint negotiates unique keys with the offerer for both directions. Answer is associated with media by the SSRC in MIKEY.

Secure Retargeting: Yes, each target negotiates unique keys with the offerer for both directions.

MIKEY-DHHMAC

Secure Forking: Yes, each forked endpoint negotiates unique keys with the offerer for both directions. Answer is associated with media by the SSRC in MIKEY.

Secure Retargeting: Yes, each target negotiates unique keys with the offerer for both directions. Note that for the keys to be meaningful, it would require the PSK to be the same for all the potential intermediaries, which would only happen within a single domain.

Security Descriptions

Secure Forking: No. Each forked endpoint sees the offerer's key. Answer is not associated with media.

Secure Retargeting: No. Each target sees the offerer's key.

SDP-DH

Secure Forking: Yes. Each forked endpoint calculates a unique SRTP key. Answer is not associated with media.

Secure Retargeting: Yes. The final target calculates a unique

SRTP key.

ZRTP

Secure Forking: Yes. Each forked endpoint calculates a unique SRTP key. As ZRTP isn't signaled in SDP, there is no association of the answer with media.

Secure Retargeting: Yes. The final target calculates a unique SRTP key.

EKT

Secure Forking: Inherited from the bootstrapping mechanism (the specific MIKEY mode or Security Descriptions). Answer is associated with media by the SPI in the EKT protocol. Answer is associated with media by the SPI in the EKT protocol.

Secure Retargeting: Inherited from the bootstrapping mechanism (the specific MIKEY mode or Security Descriptions).

RTP-DTLS

Secure Forking: Yes. Each forked endpoint calculates a unique SRTP key. Answer is associated with media by the certificate fingerprint in signaling and certificate in the media path.

Secure Retargeting: Yes. The final target calculates a unique SRTP key.

DTLS-SRTP

Secure Forking: Yes. Each forked endpoint calculates a unique SRTP key. Answer is associated with media by the certificate fingerprint in signaling and certificate in the media path.

Secure Retargeting: Yes. The final target calculates a unique SRTP key.

4.2. Clipping Media Before SDP Answer

With RTP, the offerer is able to play out any media that arrives prior to the SDP answer arriving if the answerer uses the same payload types as the offerer, as is common practice. To avoid clipping, the offerer immediately plays out media as soon as it is received, even if it hasn't yet received the answer. With SRTP, however, the offerer needs to know the SRTP key in order to decrypt the media before it can play the media. If the SRTP media arrives before the associated SRTP key, the offerer cannot play the media -- causing clipping.

For key exchange mechanisms which send the answerer's key in SDP, a

SIP provisional response [RFC3261] such as 183 (session progress) is useful. However the 183 messages aren't reliable unless both the calling and called endpoint support PRACK [RFC3262], use TCP across all SIP proxies, implement Security Preconditions [I-D.ietf-mmusic-securityprecondition], or the both ends implement ICE [I-D.ietf-mmusic-ice] and the answerer implements the reliable provisional response mechanism described in ICE. However, there is not wide deployment of any of these techniques and there is industry reluctance to requiring these techniques as solutions to avoid the problem described in this section.

Furthermore, the problem gets compounded when forking is used. For example, if using a Diffie-Hellman keying technique with security preconditions that forks to 20 endpoints, the call initiator would get 20 provisional responses containing 20 signed Diffie-Hellman half keys. Calculating 20 DH secrets and validating signatures can be a difficult task depending on the device capabilities.

The following list compares the behavior of clipping before SDP answer for each keying mechanism.

MIKEY-NULL
Not clipped. The offerer provides the answerer's keys.

MIKEY-PSK
Not clipped. The offerer provides the answerer's keys.

MIKEY-RSA
Not clipped. The offerer provides the answerer's keys.

MIKEY-RSA-R
Clipped. The answer contains the answerer's encryption key.

MIKEY-DHSIGN
Clipped. The answer contains the answerer's Diffie-Hellman response.

MIKEY-DHHMAC
Clipped. The answer contains the answerer's Diffie-Hellman response.

Security Descriptions
Clipped. The answer contains the answerer's encryption key.

SDP-DH
Clipped. The answer contains the answerer's Diffie-Hellman response.

ZRTP
Not clipped because the session intially uses RTP. While RTP is flowing, both ends negotiate SRTP keys in the media path and then switch to using SRTP.

EKT
Not clipped. The answerer sends its encryption key in RTCP, which arrives at the same time (or before) the first SRTP packet encrypted with that key.

Note: RTCP needs to work, in the answerer-to-offerer direction, before the offerer can decrypt SRTP media.

RTP-DTLS
Not clipped. Media keys are exchanged in the media path without relying on the signaling path.

DTLS-SRTP
Not clipped. Keys are exchanged in the media path without relying on the signaling path.

4.3. Centralized Keying

For efficient scaling, large audio and video conference bridges operate most efficiently by encrypting the current speaker once and distributing that stream to the conference attendees. Typically, inactive participants receive the same streams -- they hear (or see) the active speaker(s), and the active speakers receive distinct streams that don't include themselves. In order to maintain confidentiality of such conferences where listeners share a common key, all listeners must rekeyed when a listener joins or leaves a conference.

An important use case for mixers/translators is a conference bridge:

```
                                         +----+
                            A --- 1 --->|    |
                              <-- 2 ----| M  |
                                        | I  |
                            B --- 3 --->| X  |
                              <-- 4 ----| E  |
                                        | R  |
                            C --- 5 --->|    |
                              <-- 6 ----|    |
                                        +----+
```

Figure 4: Centralized Keying

In the figure above, 1, 3, and 5 are RTP media contributions from Alice, Bob, and Carol, and 2, 4, and 6 are the RTP flows to those devices carrying the 'mixed' media.

Several scenarios are possible:

a. Multiple inbound sessions: 1, 3, and 5 are distinct RTP sessions,
b. Multiple outbound sessions: 2, 4, and 6 are distinct RTP sessions,
c. Single inbound session: 1, 3, and 5 are just different sources within the same RTP session,
d. Single outbound session: 2, 4, and 6 are different flows of the same (multi-unicast) RTP session

If there are multiple inbound sessions and multiple outbound sessions (scenarios a and b), then every keying mechanism behaves as if the mixer were an endpoint and can set up a point-to-point secure session between the participant and the mixer. This is the simplest situation, but is computationally wasteful, since SRTP processing has to be done independently for each participant. The use of multiple inbound sessions (scenario a) doesn't waste computational resources, though it does consume additional cryptographic context on the mixer for each participant and has the advantage of non-repudiation of the originator of the incoming stream.

To support a single outbound session (scenario d), the mixer has to dictate its encryption key to the participants. Some keying mechanisms allow the transmitter to determine its own key, and others allow the offerer to determine the key for the offerer and answerer. Depending on how the call is established, the offerer might be a participant (such as a participant dialing into a conference bridge) or the offerer might be the mixer (such as a conference bridge

calling a participant).

The use of offerless Invites may help some keying mechanisms reverse the role of offerer/answerer. A difficulty, however, is knowing a priori if the role should be reversed for a particular call.

The following list describes how each keying mechanism behaves with centralized keying (scenario d) and rekeying.

MIKEY-NULL

Keying: Yes, if offerer is the mixer. No, if offerer is the participant (end user).

Rekeying: Yes, via re-Invite

MIKEY-PSK

Keying: Yes, if offerer is the mixer. No, if offerer is the participant (end user).

Rekeying: Yes, with a re-Invite

MIKEY-RSA

Keying: Yes, if offerer is the mixer. No, if offerer is the participant (end user).

Rekeying: Yes, with a re-Invite

MIKEY-RSA-R

Keying: No, if offerer is the mixer. Yes, if offerer is the participant (end user).

Rekeying: n/a

MIKEY-DHSIGN

Keying: No; a group-key Diffie-Hellman protocol is not supported.

Rekeying: n/a

MIKEY-DHHMAC

Keying: No; a group-key Diffie-Hellman protocol is not supported.

Rekeying: n/a

Security Descriptions

Keying: Yes, if offerer is the mixer. Yes, if offerer is the participant.

Rekeying: Yes, with a Re-Invite

SDP-DH

Keying: No; a group-key Diffie-Hellman protocol is not supported.

Rekeying: n/a

ZRTP

Keying: No; a group-key Diffie-Hellman protocol is not supported.

Rekeying: n/a

EKT

Keying: Yes. After bootstrapping a KEK using SDES or MIKEY, each member originating an SRTP stream can send its SRTP master key, sequence number and ROC via RTCP.

Rekeying: Yes. EKT supports each sender to transmit its SRTP master key to the group via RTCP packets. Thus, EKT supports each originator of an SRTP stream to rekey at any time.

RTP-DTLS

Keying: Yes, because with the assumed cipher suite, TLS_RSA_WITH_3DES_EDE_CBC_SHA, each end indicates its SRTP key.

Rekeying: via DTLS in the media path.

DTLS-SRTP

Keying: Yes, because with the assumed cipher suite, TLS_RSA_WITH_3DES_EDE_CBC_SHA, each end indicates its SRTP key.

Rekeying: via DTLS in the media path.

4.4. SSRC and ROC

In SRTP, a cryptographic context is defined as the SSRC, destination network address, and destination transport port number. Whereas RTP, a flow is defined as the destination network address and destination transport port number. This results in a problem -- how to communicate the SSRC so that the SSRC can be used for the cryptographic context.

Two approaches have emerged for this communication. One, used by all MIKEY modes, is to communicate the SSRCs to the peer in the MIKEY exchange. Another, used by Security Descriptions, is to use "late bindng" -- that is, any new packet containing a previously-unseen SSRC (which arrives at the same destination network address and destination transport port number) will create a new cryptographic context. Another approach, common amongst techniques with media-path SRTP key establishment, is to require a handshake over that media path before SRTP packets are sent. MIKEY's approach changes RTP's SSRC collision detection behavior by requiring RTP to pre-establish the SSRC values for each session.

Another related issue is that SRTP introduces a rollover counter (ROC), which records how many times the SRTP sequence number has rolled over. As the sequence number is used for SRTP's default ciphers, it is important that all endpoints know the value of the ROC. The ROC starts at 0 at the beginning of a session.

Some keying mechanisms cause a two-time pad to occur if two endpoints of a forked call have an SSRC collision.

Note: A proposal has been made to send the ROC value on every Nth SRTP packet[I-D.lehtovirta-srtp-rcc]. This proposal has not yet been incorporated into this document.

The following list examines handling of SSRC and ROC:

MIKEY-NULL
: Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

MIKEY-PSK
: Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

MIKEY-RSA
: Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

MIKEY-RSA-R
: Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

MIKEY-DHSIGN
Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

MIKEY-DHHMAC
Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

Security Descriptions
Neither SSRC nor ROC are signaled. SSRC 'late binding' is used.

SDP-DH
Neither SSRC nor ROC are signaled. SSRC 'late binding' is used.

ZRTP
Neither SSRC nor ROC are signaled. SSRC 'late binding' is used.

EKT
The SSRC of the SRTCP packet containing an EKT update corresponds to the SRTP master key and other parameters within that packet.

RTP-DTLS
Neither SSRC nor ROC are signaled. SSRC 'late binding' is used.

DTLS-SRTP
Neither SSRC nor ROC are signaled. SSRC 'late binding' is used.

5. Evaluation Criteria - Security

This section evaluates each keying mechanism on the basis of their security properties.

5.1. Public Key Infrastructure

There are two aspects of PKI requirements -- one aspect is if PKI is necessary in order for the mechanism to function at all, the other is if PKI is used to authenticate a certificate. With interactive communications it is desirable to avoid fetching certificates that delay call setup; rather it is preferable to fetch or validate certificates in such a way that call setup isn't delayed. For example, a certificate can be validated while the phone is ringing or

can be validated while ring-back tones are being played or even while the called party is answering the phone and saying "hello".

SRTP key exchange mechanisms that require a global PKI to operate are gated on the deployment of a common PKI available to both endpoints. This means that no media security is achievable until such a PKI exists. For SIP, something like sipping-certs [I-D.ietf-sipping-certs] might be used to obtain the certificate of a peer.

> Note: Even if Sipping-certs was deployed, the retargeting problem (Section 4.1) would still prevent successful deployment of keying techniques which require the offerer to obtain the actual target's public key.

The following list compares the PKI requirements of each keying mechanism, both if a PKI is required for the key exchange itself, and if PKI is only used to authenticate the certificate supplied in signaling.

MIKEY-NULL
: PKI not used.

MIKEY-PSK
: PKI not used; rather, all endpoints must have some way to exchange per-endpoint or per-system pre-shared keys.

MIKEY-RSA
: The offerer obtains the intended answerer's public key before initiating the call. This public key is used to encrypt the SRTP keys. There is no defined mechanism for the offerer to obtain the answerer's public key, although [I-D.ietf-sipping-certs] might be viable in the future.

MIKEY-RSA-R
: The offer contains the offerer's public key. The answerer uses that public key to encrypt the SRTP keys that will be used by the offerer and the answerer. A PKI is necessary to validate the certificates.

MIKEY-DHSIGN
: PKI is used to authenticate the public key that is included in the MIKEY message, by walking the CA trust chain.

MIKEY-DHHMAC
PKI not used; rather, all endpoints must have some way to exchange per-endpoint or per-system pre-shared keys.

Security Descriptions
PKI not used.

SDP-DH
PKI not used.

ZRTP
PKI not used.

EKT
PKI not used.

RTP-DTLS
Remote party's certificate is sent in media path, and a fingerprint of the same certificate is sent in the signaling path. PKI is used to authenticate the remote party's certificate, by walking the CA trust chain.

DTLS-SRTP
Remote party's certificate is sent in media path, and a fingerprint of the same certificate is sent in the signaling path. PKI is used to authenticate the remote party's certificate, by walking the CA trust chain..

5.2. Perfect Forward Secrecy

In the context of SRTP, Perfect Forward Secrecy is the property that SRTP session keys that protected a previous session are not compromised if the static keys belonging to the endpoints are compromised. That is, if someone were to record your encrypted session content and later acquires either party's private key, that encrypted session content would be safe from decryption if your key exchange mechanism had perfect forward secrecy.

The following list describes how each key exchange mechanism provides PFS.

MIKEY-NULL
No PFS.

MIKEY-PSK
No PFS.

MIKEY-RSA
No PFS.

MIKEY-RSA-R
No PFS.

MIKEY-DHSIGN
PFS is provided with the Diffie-Hellman exchange.

MIKEY-DHHMAC
PFS is provided with the Diffie-Hellman exchange.

Security Descriptions
No PFS.

SDP-DH
PFS is provided with the Diffie-Hellman exchange.

ZRTP
PFS is provided with the Diffie-Hellman exchange.

EKT
No PFS.

RTP-DTLS
PFS is achieved if the negotiated cipher suite includes an exponential or discrete-logarithmic key exchange (such as Diffie-Hellman or Elliptic Curve Diffie-Hellman [I-D.ietf-tls-ecc]).

DTLS-SRTP
PFS is achieved if the negotiated cipher suite includes an exponential or discrete-logarithmic key exchange (such as Diffie-Hellman or Elliptic Curve Diffie-Hellman [I-D.ietf-tls-ecc]).

5.3. Opportunistic Encryption

With opportunistic encryption, SRTP is used if possible but otherwise RTP is used.

SIP needs a backwards-compatible opportunistic encryption in order for SRTP to work successfully with SIP retargeting and forking.

Consider the case of Bob, with a phone that only does RTP and a voice mail system that supports SRTP and RTP. If Alice calls Bob with an SRTP offer, Bob's RTP-only phone will reject the media stream (with an empty "m=" line) because Bob's phone doesn't understand SRTP (RTP/SAVP). Alice's phone will see this rejected media stream and may terminate the entire call (BYE) and re-initiate the call as RTP-only, or Alice's phone may decide to continue with call setup with the SRTP-capable leg (the voice mail system). If Alice's phone decided to re-initiate the call as RTP-only, and Bob doesn't answer his phone, Alice will then leave voice mail using only RTP, rather than SRTP as expected.

Currently, several techniques are commonly considered as candidates to provide opportunistic encryption:

multipart/alternative
: [I-D.jennings-sipping-multipart] describes how to form a multipart/alternative body part in SIP. The significant issues with this technique are (1) that multipart MIME is incompatible with existing SIP proxies, firewalls, Session Border Controllers, and endpoints and (2) when forking, the Heterogeneous Error Response Forking Problem (HERFP) [I-D.mahy-sipping-herfp-fix] causes problems if such non-multipart-capable endpoints were involved in the forking. Retargeting which involves a non-multipart-capable device also causes retargeting to prematurely stop.

SDP Grouping
: A new SDP grouping mechanism (following the idea introduced in [RFC3388]) has been discussed which would allow a media line to indicate RTP/AVP and another media line to indicate RTP/SAVP, allowing non-SRTP-aware endpoints to choose RTP/AVP and SRTP-aware endpoints to choose RTP/SAVP. As of this writing, this SDP grouping mechanism has not been published as an Internet Draft.

session attribute
: With this technique, the endpoints signal their desire to do SRTP by signaling RTP (RTP/AVP), and using an attribute ("a=") in the SDP. This technique is entirely backwards compatible with non-SRTP-aware endpoints, but doesn't use the RTP/SAVP protocol registered by SRTP [RFC3711].

Probing
: With this technique, the endpoints first establish an RTP session using RTP (RTP/AVP). The endpoints send probe messages, over the media path, to determine if the remote endpoint supports their keying technique.

The following list compares the availability of opportunistic encryption for each keying mechanism.

MIKEY-NULL
No opportunistic encryption.

MIKEY-PSK
No opportunistic encryption.

MIKEY-RSA
No opportunistic encryption.

MIKEY-RSA-R
No opportunistic encryption.

MIKEY-DHSIGN
No opportunistic encryption.

MIKEY-DHHMAC
No opportunistic encryption.

Security Descriptions
No opportunistic encryption.

SDP-DH
No opportunistic encryption.

ZRTP
Opportunistic encryption is done by probing (sending RTP messages with header extensions) or by session attribute (see "a=zrtp", defined in section 10 of [I-D.zimmermann-avt-zrtp]). Current implementations of ZRTP use probing.

EKT
No opportunistic encryption.

RTP-DTLS
No opportunistic encryption.

DTLS-SRTP
No opportunistic encryption.

6. Security Considerations

This entire document discusses security.

7. Acknowledgements

Special thanks to Steffen Fries and Dragan Ignjatic for their excellent MIKEY comparison document [I-D.ietf-msec-mikey-applicability].

Thanks also to Cullen Jennings, David Oran, David McGrew, Mark Baugher, Flemming Andreasen, Eric Raymond, Dave Ward, Leo Huang, Eric Rescorla, Lakshminath Dondeti, Steffen Fries, Alan Johnston, and Dragon Ignjatic for their assistance with this document.

8. IANA Considerations

This document does not add new IANA registrations.

9. References

9.1. Normative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[I-D.ietf-sip-identity] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

9.2. Informational References

[I-D.ietf-mmusic-sdescriptions] Andreasen, F., "Session Description Protocol Security Descriptions for Media Streams", draft-ietf-mmusic-sdescriptions-12 (work in progress), September 2005.

[I-D.ietf-mmusic-kmgmt-ext] Arkko, J., "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)", draft-ietf-mmusic-kmgmt-ext-15 (work in

progress), June 2005.

[I-D.ietf-mmusic-securityprecondition]
Andreasen, F. and D. Wing, "Security Preconditions for Session Description Protocol Media Streams", draft-ietf-mmusic-securityprecondition-01 (work in progress), October 2005.

[I-D.ietf-msec-mikey-dhhmac]
Euchner, M., "HMAC-authenticated Diffie-Hellman for MIKEY", draft-ietf-msec-mikey-dhhmac-11 (work in progress), April 2005.

[I-D.ietf-msec-mikey-ecc]
Milne, A., "ECC Algorithms For MIKEY", draft-ietf-msec-mikey-ecc-00 (work in progress), February 2006.

[I-D.ietf-msec-mikey-rsa-r]
Ignjatic, D., "An additional mode of key distribution in MIKEY: MIKEY-RSA-R", draft-ietf-msec-mikey-rsa-r-04 (work in progress), April 2006.

[I-D.ietf-sipping-certs]
Jennings, C. and J. Peterson, "Certificate Management Service for The Session Initiation Protocol (SIP)", draft-ietf-sipping-certs-03 (work in progress), March 2006.

[I-D.mahy-sipping-herfp-fix]
Mahy, R., "A Solution to the Heterogeneous Error Response Forking Problem (HERFP) in the Session Initiation Protocol (SIP)", draft-mahy-sipping-herfp-fix-01 (work in progress), March 2006.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security

(TLS) Protocol Version 1.1", RFC 4346, April 2006.

[I-D.fischl-sipping-media-dtls]
Fischl, J., "Session Initiation Protocol (SIP) for Media Over Datagram Transport Layer Security (DTLS)", draft-fischl-sipping-media-dtls-00 (work in progress), March 2006.

[I-D.fischl-mmusic-sdp-dtls]
Fischl, J. and H. Tschofenig, "Session Description Protocol (SDP) Indicators for Datagram Transport Layer Security (DTLS)", draft-fischl-mmusic-sdp-dtls-00 (work in progress), March 2006.

[I-D.ietf-msec-mikey-applicability]
Fries, S. and D. Ignjatic, "On the applicability of various MIKEY modes and extensions", draft-ietf-msec-mikey-applicability-00 (work in progress), May 2006.

[I-D.zimmermann-avt-zrtp]
Zimmermann, P., "ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP", draft-zimmermann-avt-zrtp-01 (work in progress), March 2006.

[I-D.baugher-mmusic-sdp-dh]
Baugher, M. and D. McGrew, "Diffie-Hellman Exchanges for Multimedia Sessions", draft-baugher-mmusic-sdp-dh-00 (work in progress), February 2006.

[I-D.mcgrew-srtp-ekt]
McGrew, D., "Encrypted Key Transport for Secure RTP", draft-mcgrew-srtp-ekt-00 (work in progress), February 2006.

[I-D.lehtovirta-srtp-rcc]
Lehtovirta, V., "Integrity Transform Carrying Roll-over Counter", draft-lehtovirta-srtp-rcc-01 (work in progress), February 2006.

[I-D.ietf-tls-ecc]
Gupta, V., "ECC Cipher Suites for TLS", draft-ietf-tls-ecc-12 (work in progress), October 2005.

[I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements",

draft-peterson-sipping-retarget-00 (work in progress), February 2005.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-08 (work in progress), March 2006.

[I-D.ietf-sip-connected-identity]
Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", draft-ietf-sip-connected-identity-00 (work in progress), April 2006.

[I-D.jennings-sipping-multipart]
Wing, D. and C. Jennings, "Session Initiation Protocol (SIP) Offer/Answer with Multipart Alternative", draft-jennings-sipping-multipart-02 (work in progress), March 2006.

[I-D.draft-mcgrew-dtls-srtp]
McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)", draft-mcgrew-tls-srtp-00 (work in progress), March 2006, <http://scm.sipfoundry.org/rep/ietf-drafts/ekr/draft-mcgrew-dtls-srtp.txt>.

Authors' Addresses

Francois Audet
Nortel
4655 Great America Parkway
Santa Clara, CA 95054
USA

Email: audet@nortel.com

Dan Wing
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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