SIP Working Group
Internet-Draft
Intended status: Informational
Expires: July 13, 2009

D. Wing, Ed.
Cisco
S. Fries
Siemens AG
H. Tschofenig
Nokia Siemens Networks
F. Audet
Nortel

January 9, 2009

Requirements and Analysis of Media Security Management Protocols
draft-ietf-sip-media-security-requirements-09

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 13, 2009.

Copyright Notice

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes requirements for a protocol to negotiate a security context for SIP-signaled SRTP media. In addition to the natural security requirements, this negotiation protocol must interoperate well with SIP in certain ways. A number of proposals have been published and a summary of these proposals is in the appendix of this document.

Table of Contents

1. Introduction

The work on media security started when the Session Initiation Protocol (SIP) was still in its infancy. With the increased SIP deployment and the availability of new SIP extensions and related protocols, the need for end-to-end security was re-evaluated. The procedure of re-evaluating prior protocol work and design decisions is not an uncommon strategy and, to some extent, considered necessary to ensure that the developed protocols indeed meet the previously envisioned needs for the users on the Internet.

This document summarizes media security requirements, i.e., requirements for mechanisms that negotiate security context such as cryptographic keys and parameters for SRTP.

The organization of this document is as follows: Section 2 introduces terminology, Section 3 describes various attack scenarios against the signaling path and media path, Section 4 provides an overview about possible call scenarios, Section 5 lists requirements for media security. The main part of the document concludes with the security considerations Section 6, IANA considerations Section 7 and an acknowledgement section in Section 8. Appendix A lists and compares available solution proposals. The following Appendix A.4 compares the different approaches regarding their suitability for the SIP signaling scenarios described in Appendix A, while Appendix A.5 provides a comparison regarding security aspects. Appendix B lists non-goals for this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119], with the important qualification that, unless otherwise stated, these terms apply to the design of the media security key management protocol, not its implementation or application.

Furthermore, the terminology described in SIP ([RFC3261]) regarding functions and components are used throughout the document

Additionally, the following items are used in this document:

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 keystream 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.

Perfect Forward Secrecy (PFS): The property that disclosure of the long-term secret keying material that is used to derive an agreed ephemeral key does not compromise the secrecy of agreed keys from earlier runs.

active adversary: An active adversary is able to alter data communication to affect its operation (see also [RFC4949]).

passive adversary: A passive adversary is able to learn information from data communication, but not alter that data communication (see also[RFC4949]).

signaling path: The signaling path is the route taken by SIP signaling messages transmitted between the calling and called user agents. This can be either direct signaling between the calling and called user agents or, more commonly involves the SIP proxy servers that were involved in the call setup.

media path: The media path is the route taken by media packets exchanged by the endpoints. In the simplest case, the endpoints exchange media directly, and the "media path" is defined by a quartet of IP addresses and TCP/UDP ports, along with an IP route. In other cases, this path may include RTP relays, mixers, transcoders, session border controllers, NATs, or media gateways.

Moreover, as this document discusses requirements for media security, the nomenclature R-XXX is used to mark requrements, were XXX is the requirement, which needs to be met.

3. Attack Scenarios

The discussion in this section relates to requirements R-PASS-MEDIA, R-PASS-SIG, R-ASSOC, R-SIG-MEDIA, R-ACT-ACT, and R-ID-BINDING.

This document classifies adversaries according to their access and their capabilities. An adversary might have access:

1. only to the media path,

2. only to the signaling path,

3. to the media path and to the signaling path.

An attacker that can solely be located along the signaling path, and does not have access to media (item 2), is not considered in this document.

There are two different types of adversaries, active and passive. An active adversary may need to be active with regard to the key exchange relevant information traveling along the media path or traveling along the signaling path.

Based on their robustness against the adversary capabilities described above, we can group security mechanisms using the following labels. This list is generally ordered from easiest to compromise (at the top) to more difficult to compromise:

SIP signaling	media	abbreviation
none	passive	no-signaling-passive-media
none	active	no-signaling-active-media
passive	passive	passive-signaling-passive-media
passive	active	passive-signaling-active-media
active	passive	active-signaling-passive-media
active	active	active-signaling-active-media
active	active	active-signaling-active-media-detect

no-signaling-passive-media:
Access to only the media path is sufficient to reveal the content of the media traffic.

passive-signaling-passive-media:
Passive attack on the signaling and passive attack on the media path is necessary to reveal the content of the media traffic.

passive-signaling-active-media:
Passive attack on the signaling and active attack on the media path is necessary to reveal the content of the media traffic.

active-signaling-passive-media:
Active attack on the signaling path and passive attack on the media path is necessary to reveal the content of the media traffic.

no-signaling-active-media:
Active attack on the media path is sufficient to reveal the content of the media traffic.

active-signaling-active-media:
Active attack on both the signaling path and the media path is necessary to reveal the content of the media traffic.

active-signaling-active-media-detect:
Active attack on both signaling and media path is necessary to reveal the content of the media traffic (as with active-signaling-active-media), and the attack is detectable by protocol messages exchanged between the end points.

For example, unencrypted RTP is vulnerable to no-signaling-passive-media.

As another example, Security Descriptions [RFC4568], when protected by TLS (as it is commonly implemented and deployed), belongs in the passive-signaling-passive-media category since the adversary needs to learn the Security Descriptions key by seeing the SIP signaling message at a SIP proxy (assuming that the adversary is in control of the SIP proxy). The media traffic can be decrypted using that learned key.

As another example, DTLS-SRTP falls into active-signaling-active-media category when DTLS-SRTP is used with a public key based ciphersuite with self-signed certificates and without SIP-Identity [RFC4474]. An adversary would have to modify the fingerprint that is sent along the signaling path and subsequently to modify the certificates carried in the DTLS handshake that travel along the media path. If DTLS-SRTP is used with both SIP Identity [RFC4474] and SIP Connected Identity [RFC4916], the RFC4474 signature protects both the offer and the answer, and such a system would then belong to the active-signaling-active-attack-detect category (provided, of course, the signaling path to the RFC4474 authenticator and verifier is secured as per RFC4474 and the RFC4474 authenticator and verifier are behaving as per RFC4474).

The above discussion of DTLS-SRTP demonstrates how a single security protocol can be in different classes depending on the mode in which it is operated. Other protocols can achieve similar effect by adding functions outside of the on-the-wire key management protocol itself. Although it may be appropriate to deploy lower-classed mechanisms in some cases, the ultimate security requirement for a media security negotiation protocol is that it have a mode of operation available in which is detect-attack, which provides protection against the passive and active attacks and provides detection of such attacks. That is, there must be a way to use the protocol so that an active attack is required against both the signaling and media paths, and so that such attacks are detectable by the endpoints.

4. Call Scenarios and Requirements Considerations

The following subsections describe call scenarios that pose the most challenge to the key management system for media data in cooperation with SIP signaling.

Throughout the subsections requirements are stated by using the nomenclature R- to state an explicit requirement. All of the stated requirements are explanied in detail in section Section 5. The requirements in section Section 5 are listed according their association to the key management protocol, to attack scenarios, and requirements which can be met inside the key management protocol or outside of the key management protocol.

4.1. Clipping Media Before Signaling Answer

The discussion in this section relates to requirement R-AVOID-CLIPPING and R-ALLOW-RTP.

Per the SDP Offer/Answer Model [RFC3264],

> "Once the offerer has sent the offer, it MUST be prepared to receive media for any recvonly streams described by that offer. It MUST be prepared to send and receive media for any sendrecv streams in the offer, and send media for any sendonly streams in the offer (of course, it cannot actually send until the peer provides an answer with the needed address and port information)."

To meet this requirement with SRTP, the offerer needs to know the SRTP key for arriving media. If either endpoint receives encrypted media before it has access to the associated SRTP key, it cannot play the media -- causing clipping.

For key exchange mechanisms that send the answerer's key in SDP, a SIP provisional response [RFC3261], such as 183 (session progress), is useful. However, the 183 messages are not reliable unless both the calling and called end point support PRACK [RFC3262], use TCP across all SIP proxies, implement Security Preconditions [RFC5027], or the both ends implement ICE [I-D.ietf-mmusic-ice] and the answerer implements the reliable provisional response mechanism described in ICE. Unfortunately, there is not wide deployment of any of these techniques and there is industry reluctance to require these techniques to avoid the problems described in this section.

Note that the receipt of an SDP answer is not always sufficient to allow media to be played to the offerer. Sometimes, the offerer must send media in order to open up firewall holes or NAT bindings before media can be received (for details see [I-D.ietf-mmusic-media-path-middleboxes]). In this case, even a solution that makes the key available before the SDP answer arrives will not help.

Preventing the arrival of early media (i.e., media that arrives at the SDP offerer before the SDP answer arrives) might obsolete the R-AVOID-CLIPPING requirement, but at the time of writing such early media exists in many normal call scenarios.

4.2. Retargeting and Forking

The discussion in this section relates to requirements R-FORK-RETARGET, R-DISTINCT, R-HERFP, and R-BEST-SECURE.

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 Figure 1 Alice sends an INVITE in step 1 that 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) that 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 1: Retargeting

Using retargeting might lead to situations where the User Agent

Client (UAC) does not know where its request will be going. This might not immediately seem like a serious problem; after all, when one places a telephone call on the PSTN, one never really knows if it will be forwarded to a different number, who will pick up the line when it rings, and so on. However, when considering SIP mechanisms for authenticating the called party, this function can also make it difficult to differentiate an intermediary that is behaving legitimately from an attacker. From this perspective, the main problems with retargeting are:

Not detectable by the caller: The originating user agent has no means of anticipating that the condition will arise, nor any means of determining that it has occurred until the call has already been set up.

Not preventable by the caller: There is no existing mechanism that might be employed by the originating user agent in order to guarantee that the call will not be re-targeted.

The mechanism used by SIP for identifying the calling party is SIP Identity [RFC4474]. However, due to the nature of retargeting 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 (e.g., MIKEY). However, those built-in identity mechanism also suffer from the SIP retargeting problem. While Connected Identity [RFC4916] allows positive identification of the called party, the primary difficulty still remains that the calling party does not know if a mismatched called party is legitimate (i.e., due to authorized retargeting) or illegitimate (i.e., due to unauthorized retargeting by an attacker above to modify SIP signaling).

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 2: 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 is not desirable to require ICE to accomplish this association.

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

To maintain security of the media traffic, only the end point that answers the call should know the SRTP keys for the session. Forked and re-targeted calls only reveal sensitive information to non-responders when the signaling messages contain sensitive information (e.g., SRTP keys) that is accessible by parties that receive the offer, but may not respond (i.e., the original recipients in a retargeted call, or non-answering endpoints in a forked call). For key exchange mechanisms that do not provide secure forking or secure retargeting, one workaround is to re-key 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. This doubles the number of messages processed by the network.

Further compounding this problem is a unique feature 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's credentials, it is very possible that the rejection will never reach the sender. This problem, called the Heterogeneous Error Response Forking Problem (HERFP) [RFC3326], is difficult to solve in SIP. Because we expect the HERFP to continue to be a problem in SIP for the foreseeable future, a media security system should function even in the presence of HERFP behavior.

4.3. Recording

The discussion in this section relates to requirement R-RECORDING.

Some business environments, such as stock brokers, banks, and catalog call centers, require recording calls with customers. This is the familiar "this call is being recorded for quality purposes" heard during calls to these sorts of businesses. In these environments, media recording is typically performed by an intermediate device (with RTP, this is typically implemented in a 'sniffer').

When performing such call recording with SRTP, the end-to-end security is compromised. This is unavoidable, but necessary because the operation of the business requires such recording. It is desirable that the media security is not unduly compromised by the media recording. The endpoint within the organization needs to be informed that there is an intermediate device and needs to cooperate with that intermediate device.

This scenario does not place a requirement directly on the key management protocol. The requirement could be met directly by the key management protocol (e.g., MIKEY-NULL or [RFC4568]) or through an external out-of-band-mechanism (e.g., [I-D.wing-sipping-srtp-key]).

4.4. PSTN gateway

The discussion in this section relates to requirement R-PSTN.

It is desirable, even when one leg of a call is on the PSTN, that the IP leg of the call be protected with SRTP.

A typical case of using media security where two entities are having a VoIP conversation over IP capable networks. However, there are cases where the other end of the communication is not connected to an IP capable network. In this kind of setting, there needs to be some kind of gateway at the edge of the IP network which converts the VoIP conversation to format understood by the other network. An example of such gateway is a PSTN gateway sitting at the edge of IP and PSTN networks (such as the architecture described in [RFC3372]).

If media security (e.g., SRTP protection) is employed in this kind of gateway-setting, then media security and the related key management is terminated at the PSTN gateway. The other network (e.g., PSTN) may have its own measures to protect the communication, but this means that from media security point of view the media security is not employed truely end-to-end between the communicating entities.

4.5. Call Setup Performance

The discussion in this section relates to requirement R-REUSE.

Some devices lack sufficient processing power to perform public key operations or Diffie-Hellman operations for each call, or prefer to avoid performing those operations on every call. The ability to re-use previous public key or Diffie-Hellman operations can vastly decrease the call setup delay and processing requirements for such devices.

In certain devices, it can take a second or two to perform a Diffie-Hellman operation. Examples of these devices include handsets, IP Multimedia Services Identity Module (ISIMs), and PSTN gateways. PSTN gateways typically utilize a Digital Signal Processor (DSP) which is not yet involved with typical DSP operations at the beginning of a call, thus the DSP could be used to perform the calculation, so as to avoid having the central host processor perform the calculation. However, not all PSTN gateways use DSPs (some have only central processors or their DSPs are incapable of performing the necessary public key or Diffie-Hellman operation), and handsets lack a separate, unused processor to perform these operations.

Two scenarios where R-REUSE is useful are calls between an endpoint and its voicemail server or its PSTN gateway. In those scenarios calls are made relatively often and it can be useful for the voicemail server or PSTN gateway to avoid public key operations for subsequent calls.

Storing keys across sessions often interferes with perfect forward secrecy (R-PFS).

4.6. Transcoding

The discussion in this section relates to requirement R-TRANSCODER.

In some environments is is necessary for network equipment to transcode from one codec (e.g., a highly compressed codec which makes

efficient use of wireless bandwidth) to another codec (e.g., a standardized codec to a SIP peering interface). With RTP, a transcoding function can be performed with the combination of a SIP B2BUA (to modify the SDP) and a processor to perform the transcoding between the codecs. However, with end-to-end secured SRTP, a transcoding function implemented the same way is a man in the middle attack, and the key management system prevents its use.

However, such a network-based transcoder can still be realized with the cooperation and approval of the endpoint, and can provide end-to-transcoder and transcoder-to-end security.

4.7. Upgrading to SRTP

The discussion in this section relates to the requirement R-ALLOW-RTP.

Legitimate RTP media can be sent to an endpoint for announcements, colorful ringback tones (e.g., music), advertising, or normal call progress tones. The RTP may be received before an associated SDP answer. For details on various scenarios, see [I-D.stucker-sipping-early-media-coping].

While receiving such RTP exposes the calling party to a risk of receiving malicious RTP from an attacker, SRTP endpoints will need to receive and play out RTP media in order to be compatible with deployed systems that send RTP to calling parties.

4.8. Interworking with Other Signaling Protocols

The discussion in this section relates to the requirement R-OTHER-SIGNALING.

In many environments, some devices are signaled with protocols other than SIP which do not share SIP's offer/answer model (e.g., [H.248.1] or do not utilize SDP (e.g., H.323). In other environments, both endpoints may be SIP, but may use different key management systems (e.g., one uses MIKEY-RSA, the other MIKEY-RSA-R).

In these environments, it is desirable to have SRTP -- rather than RTP -- between the two endpoints. It is always possible, although undesirable, to interwork those disparate signaling systems or disparate key management systems by decrypting and re-encrypting each SRTP packet in a device in the middle of the network (often the same device performing the signaling interworking). This is undesirable due to the cost and increased attack area, as such an SRTP/SRTP interworking device is a valuable attack target.

At the time of this writing, interworking is considered important. Interworking without decryption/encryption of the SRTP, while useful, is not yet deemed critical because the scale of such SRTP deployments is, to date, relatively small.

4.9. Certificates

The discussion in this section relates to R-CERTS.

On the Internet and on some private networks, validating another peer's certificate is often done through a trust anchor -- a list of Certificate Authorities that are trusted. It can be difficult or expensive for a peer to obtain these certificates. In all cases, both parties to the call would need to trust the same trust anchor (i.e., "certificate authority"). For these reasons, it is important that the media plane key management protocol offer a mechanism that allows end-users who have no prior association to authenticate to each other without acquiring credentials from a third party trust point. Note that this does not rule out mechanisms in which servers have certificates and attest to the identities of end-users.

5. Requirements

This section is divided into several parts: requirements specific to the key management protocol (Section 5.1), attack scenarios (Section 5.2), and requirements which can be met inside the key management protocol or outside of the key management protocol (Section 5.3).

5.1. Key Management Protocol Requirements

SIP Forking and Retargeting, from Section 4.2:

R-FORK-RETARGET:
The media security key management protocol MUST securely support forking and retargeting when all endpoints are willing to use SRTP without causing the call setup to fail. This requirement means the endpoints that did not answer the call MUST NOT learn the SRTP keys (in either direction) used by the answering endpoint.

R-DISTINCT:
The media security key management protocol MUST be capable of creating distinct, independent cryptographic contexts for each endpoint in a forked session.

R-HERFP:
The media security key management protocol MUST function securely even in the presence of HERFP behavior, i.e., the rejection of key information does not reach the sender.

Performance considerations:

R-REUSE:
The media security key management protocol MAY support the re-use of a previously established security context.

Note: re-use of the security context does not imply re-use of RTP parameters (e.g., payload type or SSRC).

Media considerations:

R-AVOID-CLIPPING:
The media security key management protocol SHOULD avoid clipping media before SDP answer without requiring Security Preconditions [RFC5027]. This requirement comes from Section 4.1.

R-RTP-CHECK:
If SRTP key negotiation is performed over the media path (i.e., using the same UDP/TCP ports as media packets), the key negotiation packets MUST NOT pass the RTP validity check defined in Appendix A.1 of [RFC3550], so that SRTP negotiation packets can be differentiated from RTP packets.

R-ASSOC:
The media security key management protocol SHOULD include a mechanism for associating key management messages with both the signaling traffic that initiated the session and with protected media traffic. It is useful to associate key management messages with call signaling messages, as this allows the SDP offerer to avoid performing CPU-consuming operations (e.g., Diffie-Hellman or public key operations) with attackers that have not seen the signaling messages.

For example, if using a Diffie-Hellman keying technique with security preconditions that forks to 20 end points, the call initiator would get 20 provisional responses containing 20 signed Diffie-Hellman key pairs. Calculating 20 Diffie-Hellman secrets and validating signatures can be a difficult task for some devices. Hence, in the case of forking, it is not desirable to perform a Diffie-Hellman operation with every party, but rather only with the party that answers the call (and incur some media clipping). To do this, the signaling and

media need to be associated so the calling party knows which key management exchange needs to be completed. This might be done by using the transport address indicated in the SDP, although NATs can complicate this association.

Note: due to RTP's design requirements, it is expected that SRTP receivers will have to perform authentication of any received SRTP packets.

R-NEGOTIATE:
The media security key management protocol MUST allow a SIP User Agent to negotiate media security parameters for each individual session. Such negotiation MUST NOT cause a two-time pad (Section 9.1 of [RFC3711]).

R-PSTN:
The media security key management protocol MUST support termination of media security in a PSTN gateway. This requirement is from Section 4.4.

5.2. Security Requirements

This section describes overall security requirements and specific requirements from the attack scenarios (Section 3).

Overall security requirements:

R-PFS:
The media security key management protocol MUST be able to support perfect forward secrecy.

R-COMPUTE:
The media security key management protocol MUST support offering additional SRTP cipher suites without incurring significant computational expense.

R-CERTS:
The key management protocol MUST NOT require that end-users obtain credentials (certificates or private keys) from a third-party trust anchor.

R-FIPS:
The media security key management protocol SHOULD use algorithms that allow FIPS 140-2 [FIPS-140-2] certification or similar country-specific certification (e.g., [AISITSEC]).

The United States Government can only purchase and use crypto implementations that have been validated by the FIPS-140

[FIPS-140-2] process:

> "The FIPS-140 standard is applicable to all Federal agencies that use cryptographic-based security systems to protect sensitive information in computer and telecommunication systems, including voice systems. The adoption and use of this standard is available to private and commercial organizations."

Some commercial organizations, such as banks and defense contractors, require or prefer equipment which has received the same validation.

R-DOS:
The media security key management protocol MUST NOT introduce any new significant denial of service vulnerabilities (e.g., the protocol should not request the endpoint to perform CPU-intensive operations without the client being able to validate or authorize the request).

R-EXISTING:
The media security key management protocol SHOULD allow endpoints to authenticate using pre-existing cryptographic credentials, e.g., certificates or pre-shared keys.

R-AGILITY:
The media security key management protocol MUST provide crypto-agility, i.e., the ability to adapt to evolving cryptography and security requirements (update of cryptographic algorithms without substantial disruption to deployed implementations)

R-DOWNGRADE:
The media security key management protocol MUST protect cipher suite negotiation against downgrading attacks.

R-PASS-MEDIA:
The media security key management protocol MUST have a mode which prevents a passive adversary with access to the media path from gaining access to keying material used to protect SRTP media packets.

R-PASS-SIG:
The media security key management protocol MUST have a mode in which it prevents a passive adversary with access to the signaling path from gaining access to keying material used to protect SRTP media packets.

R-SIG-MEDIA:
The media security key management protocol MUST have a mode in which it defends itself from an attacker that is solely on the media path and from an attacker that is solely on the signaling path. A successful attack refers to the ability for the adversary to obtain keying material to decrypt the SRTP encrypted media traffic.

R-ID-BINDING:
The media security key management protocol MUST enable the media security keys to be cryptographically bound to an identity of the endpoint.

This allows domains to deploy SIP Identity [RFC4474].

R-ACT-ACT:
The media security key management protocol MUST support a mode of operation that provides active-signaling-active-media-detect robustness, and MAY support modes of operation that provide lower levels of robustness (as described in Section 3).

Failing to meet R-ACT-ACT indicates the protocol can not provide secure end-to-end media.

5.3. Requirements Outside of the Key Management Protocol

The requirements in this section are for an overall VoIP security system. These requirements can be met within the key management protocol itself, or can be solved outside of the key management protocol itself (e.g., solved in SIP or in SDP).

R-BEST-SECURE:
Even when some end points of a forked or retargeted call are incapable of using SRTP, a solution MUST be described which allows the establishment of SRTP associations with SRTP-capable endpoints and / or RTP associations with non-SRTP-capable endpoints.

R-OTHER-SIGNALING:
A solution SHOULD be able to negotiate keys for SRTP sessions created via different call signaling protocols (e.g., between Jabber, SIP, H.323, MGCP).

R-RECORDING:
A solution SHOULD be described which supports recording of decrypted media. This requirement comes from Section 4.3.

R-TRANSCODER:
A solution SHOULD be described which supports intermediate nodes (e.g., transcoders), terminating or processing media, between the end points.

R-ALLOW-RTP: A solution SHOULD be described which allows RTP media to be received by the calling party until SRTP has been negotiated with the answerer, after which SRTP is preferred over RTP.

6. Security Considerations

This document lists requirements for securing media traffic. As such, it addresses security throughout the document.

7. IANA Considerations

This document does not require actions by IANA.

8. Acknowledgements

For contributions to the requirements portion of this document, the authors would like to thank the active participants of the RTPSEC BoF and on the RTPSEC mailing list, and a special thanks to Steffen Fries and Dragan Ignjatic for their excellent MIKEY comparison [RFC5197] document.

The authors would furthermore like to thank the following people for their review, suggestions, and comments: Flemming Andreasen, Richard Barnes, Mark Baugher, Wolfgang Buecker, Werner Dittmann, Lakshminath Dondeti, John Elwell, Martin Euchner, Hans-Heinrich Grusdt, Christer Holmberg, Guenther Horn, Peter Howard, Leo Huang, Dragan Ignjatic, Cullen Jennings, Alan Johnston, Vesa Lehtovirta, Matt Lepinski, David McGrew, David Oran, Colin Perkins, Eric Raymond, Eric Rescorla, Peter Schneider, Srinath Thiruvengadam, Dave Ward, Dan York, and Phil Zimmermann.

9. References

9.1. Normative References

[FIPS-140-2]
NIST, "Security Requirements for Cryptographic Modules", June 2005, <http://csrc.nist.gov/publications/fips/

fips140-2/fips1402.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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.

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

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, 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.

9.2. Informative References

[AISITSEC] "Anwendungshinweise und Interpretationen (AIS) zu ITSEC", January 2002, <http://www.bsi.de/zertifiz/zert/interpr/aisitsec.htm>.

[H.248.1] ITU, "Gateway control protocol", June 2000, <http://www.itu.int/rec/T-REC-H.248/e>.

[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.dondeti-msec-rtpsec-mikeyv2] Dondeti, L., "MIKEYv2: SRTP Key Management using MIKEY, revisited", draft-dondeti-msec-rtpsec-mikeyv2-01 (work in progress), March 2007.

[I-D.fischl-sipping-media-dtls] Fischl, J., "Datagram Transport Layer Security (DTLS) Protocol for Protection of Media Traffic Established with the Session Initiation Protocol", draft-fischl-sipping-media-dtls-03 (work in progress), July 2007.

[I-D.ietf-avt-dtls-srtp]
McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)", draft-ietf-avt-dtls-srtp-06 (work in progress), October 2008.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19 (work in progress), October 2007.

[I-D.ietf-mmusic-media-path-middleboxes]
Stucker, B. and H. Tschofenig, "Analysis of Middlebox Interactions for Signaling Protocol Communication along the Media Path", draft-ietf-mmusic-media-path-middleboxes-01 (work in progress), July 2008.

[I-D.ietf-mmusic-sdp-capability-negotiation]
Andreasen, F., "SDP Capability Negotiation", draft-ietf-mmusic-sdp-capability-negotiation-09 (work in progress), July 2008.

[I-D.ietf-msec-mikey-ecc]
Milne, A., "ECC Algorithms for MIKEY", draft-ietf-msec-mikey-ecc-03 (work in progress), June 2007.

[I-D.ietf-sip-certs]
Jennings, C. and J. Fischl, "Certificate Management Service for The Session Initiation Protocol (SIP)", draft-ietf-sip-certs-07 (work in progress), November 2008.

[I-D.ietf-tls-rfc4346-bis]
Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", draft-ietf-tls-rfc4346-bis-10 (work in progress), March 2008.

[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.mcgrew-srtp-ekt]
McGrew, D., "Encrypted Key Transport for Secure RTP",

draft-mcgrew-srtp-ekt-03 (work in progress), July 2007.

[I-D.stucker-sipping-early-media-coping]
Stucker, B., "Coping with Early Media in the Session Initiation Protocol (SIP)", draft-stucker-sipping-early-media-coping-03 (work in progress), October 2006.

[I-D.wing-sipping-srtp-key]
Wing, D., Audet, F., Fries, S., Tschofenig, H., and A. Johnston, "Secure Media Recording and Transcoding with the Session Initiation Protocol", draft-wing-sipping-srtp-key-04 (work in progress), October 2008.

[I-D.zimmermann-avt-zrtp]
Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-11 (work in progress), November 2008.

[RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, December 2002.

[RFC3372] Vemuri, A. and J. Peterson, "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", BCP 63, RFC 3372, September 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

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

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[RFC4650] Euchner, M., "HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY)", RFC 4650, September 2006.

[RFC4738] Ignjatic, D., Dondeti, L., Audet, F., and P. Lin, "MIKEY-RSA-R: An Additional Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 4738, November 2006.

[RFC4771] Lehtovirta, V., Naslund, M., and K. Norrman, "Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)", RFC 4771, January 2007.

[RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC5027] Andreasen, F. and D. Wing, "Security Preconditions for Session Description Protocol (SDP) Media Streams", RFC 5027, October 2007.

[RFC5197] Fries, S. and D. Ignjatic, "On the Applicability of Various Multimedia Internet KEYing (MIKEY) Modes and Extensions", RFC 5197, June 2008.

Appendix A. Overview and Evaluation of Existing 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. A disadvantage of keying over the media path is that interworking different key exchange requires the interworking function be in the media path, rather than just in the signaling path; in practice this involvement is probably unavoidable anyway.

A.1. Signaling Path Keying Techniques

A.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.

A.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.

A.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 mechanism outside of MIKEY.

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.

A.1.4. MIKEY-RSA-R

MIKEY-RSA-R [RFC4738] is essentially the same as MIKEY-RSA 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 standard X.509 validation techniques. 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.

A.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 standard X.509 validation techniques.

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.

A.1.6. MIKEY-DHHMAC

MIKEY-DHHMAC [RFC4650] 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 certificate authentication.

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.

A.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) .

With this proposal, 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 are not discussed separately in this document.

A.1.8. Security Descriptions with SIPS

Security Descriptions [RFC4568] 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:") 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.

A.1.9. Security Descriptions with S/MIME

This keying mechanism is identical to Appendix A.1.8, except that rather than protecting the signaling with TLS, the entire SDP is encrypted with S/MIME.

A.1.10. SDP-DH (expired)

SDP Diffie-Hellman [I-D.baugher-mmusic-sdp-dh] exchanges Diffie-Hellman messages in the signaling path to establish session keys. To protect against active man-in-the-middle attacks, the Diffie-Hellman exchange needs to be protected with S/MIME, SIPS, or SIP Identity [RFC4474] and SIP Conected Identity [RFC4916].

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.

A.1.11. MIKEYv2 in SDP (expired)

MIKEYv2 [I-D.dondeti-msec-rtpsec-mikeyv2] adds mode negotiation to MIKEYv1 and removes the time synchronization requirement. It

therefore now takes 2 round-trips to complete. In the first round trip, the communicating parties learn each other's identities, agree on a MIKEY mode, crypto algorithm, SRTP policy, and exchanges nonces for replay protection. In the second round trip, they negotiate unicast and/or group SRTP context for SRTP and/or SRTCP.

Furthemore, MIKEYv2 also defines an in-band negotiation mode as an alternative to SDP (see Appendix A.3.3).

A.2. Media Path Keying Technique

A.2.1. ZRTP

ZRTP [I-D.zimmermann-avt-zrtp] does not exchange information in the signaling path (although it's possible for endpoints to exchange a hash of the ZRTP Hello message with "a=zrtp-hash" in the initial Offer if sent over an integrity-protected signaling channel. This provides some useful correlation between the signaling and media layers). In ZRTP the keys are exchanged entirely in the media path using a Diffie-Hellman exchange. 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 its Diffie-Hellman exchange by having each person read digits or words to the other person. Subsequent sessions with the same ZRTP endpoint can be authenticated using the stored 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. These initial messages are all sent as non-RTP packets.

> Note that when ZRTP probing is used, unencrypted RTP can be exchanged until the SRTP keys are established.

A.3. Signaling and Media Path Keying Techniques

A.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.

A.3.2. DTLS-SRTP

DTLS-SRTP [I-D.ietf-avt-dtls-srtp] exchanges public key fingerprints in SDP [I-D.fischl-sipping-media-dtls] 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 one message from the answerer to offerer (full round trip) so the offerer can correlate the SDP answer with the answering endpoint. DTLS-SRTP uses 4 media path messages to establish the SRTP key.

This document assumes DTLS will use TLS_RSA_WITH_AES_128_CBC_SHA as its cipher suite, which is the mandatory-to-implement cipher suite in TLS [I-D.ietf-tls-rfc4346-bis].

A.3.3. MIKEYv2 Inband (expired)

As defined in Appendix A.1.11, MIKEYv2 also defines an in-band negotiation mode as an alternative to SDP (see Appendix A.3.3). The details are not sorted out in the draft yet on what in-band actually means (i.e., UDP, RTP, RTCP, etc.).

A.4. Evaluation Criteria - SIP

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

A.4.1. Secure Retargeting and Secure Forking

Retargeting and forking of signaling requests is described within Section 4.2. The following builds upon this description.

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.

MIKEYv2 in SDP
The behavior will depend on which mode is picked.

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 with SIPS
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.

Security Descriptions with S/MIME
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. Suffers from retargeting identity problem.

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
Yes. Each forked endpoint calculates a unique SRTP key. With the "a=zrtp-hash" attribute, the media can be associated with an answer.

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).

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.

MIKEYv2 Inband
The behavior will depend on which mode is picked.

A.4.2. Clipping Media Before SDP Answer

Clipping media before receiving the signaling answer is described within Section 4.1. The following builds upon this description.

Furthermore, the problem of clipping 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.

MIKEYv2 in SDP
The behavior will depend on which mode is picked.

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

Security Descriptions with S/MIME
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, as long as the first RTCP packet (containing the answerer's key) is not lost in transit. 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.

DTLS-SRTP
No clipping after the DTLS-SRTP handshake has completed. SRTP keys are exchanged in the media path. Need to wait for SDP answer to ensure DTLS-SRTP handshake was done with an authorized party.

If a middlebox interferes with the media path, there can be clipping [I-D.ietf-mmusic-media-path-middleboxes].

MIKEYv2 Inband
Not clipped. Keys are exchanged in the media path without relying on the signaling path.

A.4.3. 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 apply "late binding" -- 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[RFC4771]. 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.

MIKEYv2 in SDP
Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

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

Security Descriptions with S/MIME
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.

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

MIKEYv2 Inband
Each endpoint indicates a set of SSRCs and the ROC for SRTP packets it transmits.

A.5. Evaluation Criteria - Security

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

A.5.1. Distribution and Validation of Persistent Public Keys and Certificates

Using persistent public keys for confidentiality and authentication can introduce requirements for two types of systems, often implemented using certificates: (1) a system to distribute those persistent public keys certificates, and (2) a system for validating those persistent public keys. We refer to the former as a key distribution system and the latter as an authentication infrastructure. In many cases, a monolithic public key infrastructure (PKI) is used for fulfill both of these roles. However, these functions can be provided by many other systems. For instance, key distribution may be accomplished by any public repository of keys. Any system in which the two endpoints have access to trust anchors and intermediate CA certificates that can be used to validate other endpoints' certificates (including a system of self-signed certificates) can be used to support certificate validation in the below schemes.

With real-time communications it is desirable to avoid fetching or validating certificates that delay call setup. Rather, it is preferable to fetch or validate certificates in such a way that call setup is not 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". Even better is to avoid fetching or validating persistent public keys at all.

SRTP key exchange mechanisms that require a particular authentication infrastructure to operate (whether for distribution or validation) are gated on the deployment of a such an infrastructure available to both endpoints. This means that no media security is achievable until such an infrastructure exists. For SIP, something like sip-certs [I-D.ietf-sip-certs] might be used to obtain the certificate of a peer.

> Note: Even if sip-certs [I-D.ietf-sip-certs] was deployed, the retargeting problem (Appendix A.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 requirements introduced by the use of public-key cryptography in each keying mechanism, both for public key distribution and for certificate validation.

MIKEY-NULL
Public-key cryptography is not used.

MIKEY-PSK
Public-key cryptography is 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-sip-certs] might be viable in the future.

The offer may also contain a certificate for the offeror, which would require an authentication infrastructure in order to be validated by the receiver.

MIKEY-RSA-R
The offer contains the offerer's certificate, and the answer contains the answerer's certificate. The answerer uses the public key in the certificate to encrypt the SRTP keys that will be used by the offerer and the answerer. An authentication infrastructure is necessary to validate the certificates.

MIKEY-DHSIGN
An authentication infrastructure is used to authenticate the public key that is included in the MIKEY message.

MIKEY-DHHMAC
Public-key cryptography is not used. Rather, all endpoints must have some way to exchange per-endpoint or per-system pre-shared keys.

MIKEYv2 in SDP
The behavior will depend on which mode is picked.

Security Descriptions with SIPS
Public-key cryptography is not used.

Security Descriptions with S/MIME
Use of S/MIME requires that the endpoints be able to fetch and validate certificates for each other. The offerer must obtain the intended target's certificate and encrypts the SDP offer with the public key contained in target's certificate. The answerer must obtain the offerer's certificate and encrypt the SDP answer with the public key contained in the offerer's certificate.

SDP-DH
Public-key cryptography is not used.

ZRTP
Public-key cryptography is used (Diffie-Hellman), but without dependence on persistent public keys. Thus, certificates are not fetched or validated.

EKT
Public-key cryptography is not used by itself, but might be used by the EKT bootstrapping keying mechanism (such as certain MIKEY modes).

DTLS-SRTP
Remote party's certificate is sent in media path, and a fingerprint of the same certificate is sent in the signaling path.

MIKEYv2 Inband
The behavior will depend on which mode is picked.

A.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
Not applicable; MIKEY-NULL does not have a long-term secret.

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.

MIKEYv2 in SDP
The behavior will depend on which mode is picked.

Security Descriptions with SIPS
Not applicable; Security Descriptions does not have a long-term secret.

Security Descriptions with S/MIME
Not applicable; Security Descriptions does not have a long-term secret.

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

ZRTP
PFS is provided with the Diffie-Hellman exchange.

EKT
No PFS.

DTLS-SRTP
PFS is provided if the negotiated cipher suite uses ephemeral keys (e.g., Diffie-Hellman (DHE_RSA [I-D.ietf-tls-rfc4346-bis]) or Elliptic Curve Diffie-Hellman [RFC4492]).

MIKEYv2 Inband
The behavior will depend on which mode is picked.

A.5.3. Best Effort Encryption

With best effort encryption, SRTP is used with endpoints that support SRTP, otherwise RTP is used.

SIP needs a backwards-compatible best effort encryption in order for SRTP to work successfully with SIP retargeting and forking when there is a mix of forked or retargeted devices that support SRTP and don't support SRTP.

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) [RFC3326] causes problems if such non-multipart-capable endpoints were involved in the forking.

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].

SDP Capability Negotiation
SDP Capability Negotiation [I-D.ietf-mmusic-sdp-capability-negotiation] provides a

backwards-
compatible mechanism to allow offering both SRTP and RTP in a

single offer. This is the preferred technique.

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. A disadvantage of probing is an active attacker can interfere with probes, and until probing completes (and SRTP is established) the media is in the clear.

The preferred technique, SDP Capability Negotiation [I-D.ietf-mmusic-sdp-capability-negotiation], can be used with all key exchange mechanisms. What remains unique is ZRTP, which can also accomplish its best effort encryption by probing (sending ZRTP messages over the media path) or by session attribute (see "a=zrtp-hash" in [I-D.zimmermann-avt-zrtp]). Current implementations of ZRTP use probing.

A.5.4. Upgrading Algorithms

It is necessary to allow upgrading SRTP encryption and hash algorithms, as well as upgrading the cryptographic functions used for the key exchange mechanism. With SIP's offer/answer model, this can be computionally expensive because the offer needs to contain all combinations of the key exchange mechanisms (all MIKEY modes, Security Descriptions) and all SRTP cryptographic suites (AES-128, AES-256) and all SRTP cryptographic hash functions (SHA-1, SHA-256) that the offerer supports. In order to do this, the offerer has to expend CPU resources to build an offer containing all of this information which becomes computationally prohibitive.

Thus, it is important to keep the offerer's CPU impact fixed so that offering multiple new SRTP encryption and hash functions incurs no additional expense.

The following list describes the CPU effort involved in using each key exchange technique.

MIKEY-NULL
No significant computational expense.

MIKEY-PSK
No significant computational expense.

MIKEY-RSA
For each offered SRTP crypto suite, the offerer has to perform RSA operation to encrypt the TGK

MIKEY-RSA-R
For each offered SRTP crypto suite, the offerer has to perform public key operation to sign the MIKEY message.

MIKEY-DHSIGN
For each offered SRTP crypto suite, the offerer has to perform Diffie-Hellman operation, and a public key operation to sign the Diffie-Hellman output.

MIKEY-DHHMAC
For each offered SRTP crypto suite, the offerer has to perform Diffie-Hellman operation.

MIKEYv2 in SDP
The behavior will depend on which mode is picked.

Security Descriptions with SIPS
No significant computational expense.

Security Descriptions with S/MIME
S/MIME requires the offerer and the answerer to encrypt the SDP with the other's public key, and to decrypt the received SDP with their own private key.

SDP-DH
For each offered SRTP crypto suite, the offerer has to perform a Diffie-Hellman operation.

ZRTP
The offerer has no additional computational expense at all, as the offer contains no information about ZRTP or might contain "a=zrtp-hash".

EKT
The offerer's Computational expense depends entirely on the EKT bootstrapping mechanism selected (one or more MIKEY modes or Security Descriptions).

DTLS-SRTP
The offerer has no additional computational expense at all, as the offer contains only a fingerprint of the certificate that will be presented in the DTLS exchange.

MIKEYv2 Inband
The behavior will depend on which mode is picked.

Appendix B. Out-of-Scope

The compromise of an endpoint that has access to decrypted media (e.g., SIP user agent, transcoder, recorder) is out of scope of this document. Such a compromise might be via privilege escalation, installation of a virus or trojan horse, or similar attacks.

B.1. Shared Key Conferencing

The consensus on the RTPSEC mailing list was to concentrate on unicast, point-to-point sessions. Thus, there are no requirements related to shared key conferencing. This section is retained for informational purposes.

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 3: 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 end point 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 data origin authentication.

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 significant advantage of a single outbound session is the number of SRTP encryption operations remains constant even as the number of participants increases. However, a disadvantage is that data origin authentication is lost, allowing any participant to spoof the sender (because all participants know the sender's SRTP key).

Appendix C. Requirement renumbering in -02

[[RFC Editor: Please delete this section prior to publication.]]

Previous versions of this document used requirement numbers, which were changed to mnemonics as follows:

R1 R-FORK-RETARGET

R2 R-BEST-SECURE

R3 R-DISTINCT

R4 R-REUSE; changed from 'MAY' to 'protocol MUST support, and
SHOULD implement'

R5 R-AVOID-CLIPPING

R6 R-PASS-MEDIA

R7 R-PASS-SIG

R8 R-PFS

R9 R-COMPUTE

R10 R-RTP-CHECK

R11 (folded into R4; was reuse previous session)

R12 R-CERTS

R13 R-FIPS

R14 R-ASSOC

R15 R-ALLOW-RTP

R16 R-DOS

R17 R-SIG-MEDIA

R18 R-EXISTING

R19 R-AGILITY

R20 R-DOWNGRADE

R21 R-NEGOTIATE

R23 R-OTHER-SIGNALING

R23 R-RECORDING (R23 was duplicated in previous versions of the document)

R24 (deleted; was lawful intercept)

R25 R-TRANSCODER

R26 R-PSTN

R27 R-ID-BINDING

R28 R-ACT-ACT

Authors' Addresses

Dan Wing (editor)
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com

Steffen Fries
Siemens AG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: steffen.fries@siemens.com

Hannes Tschofenig
Nokia Siemens Networks
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@nsn.com
URI: http://www.tschofenig.priv.at

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

Email: audet@nortel.com