Network Working Group	D. Wing
Internet-Draft	Cisco
Intended status: Informational	S. Fries
Expires: March 23, 2008	Siemens AG
	H. Tschofenig
	Nokia Siemens Networks
	F. Audet
	B. Stucker
	Nortel
	September 20, 2007

TOC

Requirements and Analysis of Media Security Key Management Protocols
draft-wing-media-security-requirements-05

Status of this Memo

Abstract

A number of proposals have been published to address the need of securing media traffic. A summary of the proposals available at that time is available in the appendix of this document. Different assumptions, requirements, and usage environments justify every one of them. This document aims to summarize the discussed media security requirements. A comparison of the requirements against the individual proposals is provided.

This document is discussed on the SIP mailing list, <http://www1.ietf.org/mailman/listinfo/sip>.

Table of Contents

1. Introduction

TOC

The work on media security started a long time ago where the capability of the Session Initiation Protocol (SIP) was still at 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 extend, considered necessary protocol work to ensure that the developed protocols indeed meet the previously envisioned needs for the users in the Internet.

This document aims to summarize the discussed media security requirements, i.e., requirements for mechanisms that negotiate keys for SRTP. The organization of this document is as follows: Section 2 (Terminology) introduces terminology, Section 3 (Call Scenarios) provides an overview about possible call scenarios, Section 4 (Requirements) lists requirements for media security, Section 5 (Requirements Classification) will provide a clustering of requirements to certain deployment environments to address the problem that there might not be a single solution with universal applicability and Appendix D (Out-of-Scope) provides out-of-scope items and aspects for further discussion. The document concludes with a security considerations Section 6 (Security Considerations), IANA considerations Section 7 (IANA Considerations) and an acknowledgement section in Section 8 (Acknowledgements). Appendix A (Overview of Keying Mechanisms) lists the available solution proposals and compares them to the requirements.

2. Terminology

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.), 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.

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 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.
>
> **PKI** Public Key Infrastructure. Throughout this paper, the term PKI refers to a global PKI.

---

3. Call Scenarios

TOC

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

3.1. Clipping Media Before Signaling Answer

TOC

Per the SDP Offer/Answer Model (Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," June 2002.) [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 encrypted SRTP media arrives before the associated SRTP key, the offerer cannot play the media -- causing clipping.

For key exchange mechanisms that send the answerer's key in SDP, a SIP provisional response [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.), such as 183 (session progress), is useful. However, the 183 messages are not reliable unless both the calling and called end point support PRACK (Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)," June 2002.) [RFC3262], use TCP across all SIP proxies, implement Security Preconditions [I-D.ietf-mmusic-securityprecondition] (Andreasen, F. and D. Wing, "Security Preconditions for Session

Description Protocol (SDP) Media Streams," July 2007.), or the both ends implement ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," October 2007.) 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 set requirements regarding these techniques to avoid the problem 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. In this case a solution that makes the key available before the SDP answer arrives will not help.

Requirements are created due to early media, in the sense of pre-offer/answer media, as described in [I-D.barnes-sip-em-ps-req-sol] (Barnes, R. and M. Lepinski, "Early Media in SIP: Problem Statement, Requirements, and Analysis of Solutions," February 2007.). Fixes to early media might make the requirements to become obsolete, but at the time of writing no progress has been accomplished.

3.2. Retargeting and Forking

TOC

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 (Retargeting) 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.

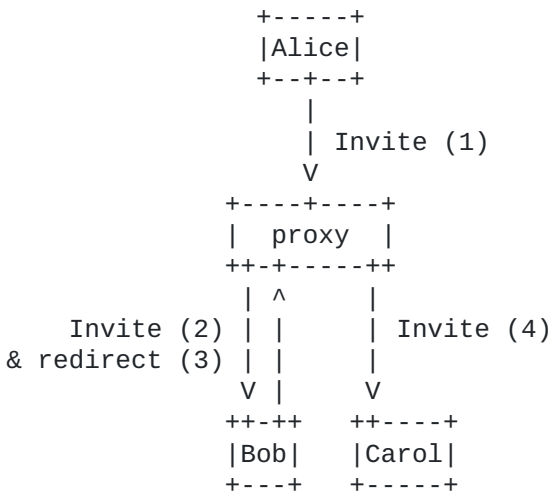

Figure 1: Retargeting

The mechanism used by SIP for identifying the calling party is SIP Identity [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.). However, due to SIP retargeting issues [I-D.peterson-sipping-retarget] (Peterson, J., “Retargeting and Security in SIP: A Framework and Requirements,” February 2005.), 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, Connected Identity (Elwell, J., “Connected Identity in the Session Initiation Protocol (SIP),” June 2007.) [RFC4916] allows identifying the called party. This is also 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.

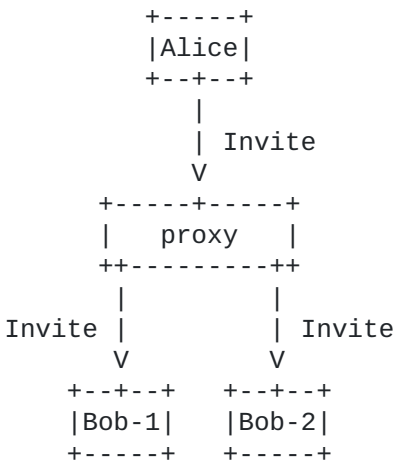

Figure 2: Forking

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] (Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," October 2007.), 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 and rollover to voice mail if neither phone is answered.

To maintain security of the media traffic, only the end point that answers the call should know the SRTP keys for the session. This is only an issue with public key encryption and not with DH-based approaches. For key exchange mechanisms that do not provide secure forking or secure retargeting, one workaround is to re-key immediately after forking or retargeting (that is, perform a re-Invite). 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 Invite messages processed by the network.

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 actual recipient will not be able to decrypt that material because the 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).

For some forms of key management the calling party needs to know in advance the certificate or shared secret of the called party, and retargeting can interfere with this.

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's credentials, it is very possible that the rejection will never reach the sender. This problem, called the Heterogeneous Error Response Forking Problem (HERFP) (Mahy, R., "A Solution to the Heterogeneous Error Response Forking Problem (HERFP) in the Session Initiation Protocol (SIP)," March 2006.) [I-D.mahy-sipping-herfp-fix], is difficult to solve in SIP.

3.3. Shared Key Conferencing

TOC

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:

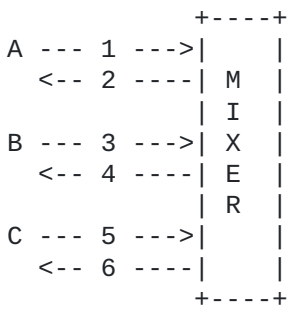

Figure 3: Centralized Keying

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 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.

3.4. B2BUA Signaling Manipulation

TOC

SRTP keying may be impacted due the presence of Back-to-Back User Agents (B2BUA) in the signaling path. Not only does this potentially impact the ability to exchange keying material as part of SIP signaling, but because B2BUAs often limit the exchange of SDP, B2BUAs can impact exchange of keying material in the media path as well. Specifically, a number of scenarios can arise during initial call setup that can interfere with exchanging SRTP keying material between endpoints:

1. UAC indicated support for PRACK [RFC3262] (Rosenberg, J. and H. Schulzrinne, “Reliability of Provisional Responses in Session Initiation Protocol (SIP),” June 2002.) is stripped from signaling,

2. SDP from either endpoint is not exchanged on the same message type or message sequence in which it was sent,

3. UAC reliability extensions, such as PRACK (Rosenberg, J. and H. Schulzrinne, “Reliability of Provisional Responses in Session Initiation Protocol (SIP),” June 2002.) [RFC3262] and Security Preconditions (Andreasen, F. and D. Wing, “Security Preconditions for Session Description Protocol (SDP) Media Streams,” July 2007.) [I-D.ietf-mmusic-securityprecondition] are terminated at the B2BUA itself instead of at the intended recipient,

4. the B2BUA introduces new branches to the call flow (forking) to network media endpoints

B2BUAs may strip support for PRACK from INVITEs in order to simplify the types of signaling scenarios they must support when, usually, trying to trigger network-provided early media. This impacts SRTP keying by preventing the UAS from exchanging keying material in the SDP answer until the next response can be sent. Even UPDATE cannot be used to transport keying material due to limitations in [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) requiring the answer to the offer in an INVITE being limited to a reliable response.

Another not-uncommon manipulation of SIP call setup signaling is to change the ordering in which SDP is exchanged. For example, a B2BUA may hold onto SDP sent to it by a UAS as part of a 18x response or UPDATE exchange and not forward that information back to the UAC until some later point in time (typically the 200 OK to the INVITE). This can delay key exchanges and cause clipping as a result.

A less common, but observed B2BUA tactic for handling signaling interactions during call setup, primarily for network-provided early media, is to "fake-out" the UAC into thinking that reliability extensions such as PRACK [RFC3262] (Rosenberg, J. and H. Schulzrinne, “Reliability of Provisional Responses in Session Initiation Protocol (SIP),” June 2002.) or Resource Management Preconditions [RFC3312] (Camarillo, G., Marshall, W., and J. Rosenberg, “Integration of Resource Management and Session Initiation Protocol (SIP),” October 2002.) are in effect end-to-end when they are not. This manifests itself by sending provisional responses reliably from the perspective of the B2BUA while stripping the extensions from INVITEs sent to the callee's UAS. It is worth noting that such behavior is likely to be applied to Security Preconditions [I-D.ietf-mmusic-securityprecondition] (Andreasen, F. and D. Wing, “Security Preconditions for Session Description Protocol (SDP) Media Streams,” July 2007.) as well for similar reasons.

Finally, B2BUAs may introduce early SIP dialogs to network-provided early media services even though no forking occurs towards the intended callee. The impact of forking of signaling requests is described within section Section 3.2 (Retargeting and Forking).

The impacts of these types of signaling manipulations by B2BUAs is currently left as an OPEN ISSUE.

3.5. Policy and Media Gating Interactions

TOC

Another class of SRTP key exchange interactions that can occur is due to policy policing and media stream gating mechanisms. These functions are often performed by Session Border Controllers or by firewalls. In the case of media stream gating, the flow of RTP packets between endpoints is not authorized until a complete SDP offer/answer exchange has taken place, commonly contingent upon the 200 OK to the INVITE being received by the network entity controlling the media gates. As a result, in-band keying cannot start prior to the flow of packets being authorized. If in-band keying is used it may be possible to detect that the RTP packet in question is part of a key exchange and not part of any data transfer process. However, the firewalls responsible for gating media are typically not inspecting the actual packets received, they are simply dropping them on the floor until the gate is opened.

Policy policing, which is often related to media stream gating, can also cause potential issues. For example, if elements such as a deep-packet inspection element were not expecting in-band SRTP key exchanges these packets may be suppressed according to local policy for not conforming to expected traffic profiles (specifically, not being an SRTP packet).

The impacts of these types of policy and gating related interactions is currently left as an OPEN ISSUE.

---

4. Requirements

TOC

R1: Negotiation of SRTP keys MUST NOT cause the call setup to fail in forked and retargeted calls where all end points are willing to use SRTP, unless the execution of the authentication and key exchange protocol leads to a failure (e.g., an unsuccessful authentication attempt).

R2: Even when some end points of a forked or retargeted call are incapable of using SRTP, the key management protocol MUST allow the establishment of SRTP associations with SRTP-capable

endpoints and / or RTP associations with non-SRTP-capable endpoints.

R3: Forked end points MUST NOT know the SRTP key of any call established with another forked end point.

R4: The media security key management protocol MAY support the ability to utilize an initially established security context that was established as part of the first call setup with a remote end point.
Specialized devices may need to avoid public key operations or Diffie-Hellman operations as much as possible because of the computational cost or because of the additional call setup delay. For example, it can take a second or two to perform a Diffie-Hellman operation in certain devices. Examples of these specialized devices would include some handsets, intelligent SIMs, and PSTN gateways. For the typical case because a phone call has not yet been established, ancillary processing cycles can be utilized to perform the PK or DH operation; for example, in a PSTN gateway the DSP, which is not yet involved with typical DSP operations, could be used to perform the calculation, so as to avoid having the central host processor perform the calculation. Some devices, such as handsets, and intelligent SIMs do not have such ancillary processing capability.

R5: The media security key management protocol SHOULD avoid clipping media before SDP answer without requiring Security Preconditions (Andreasen, F. and D. Wing, “Security Preconditions for Session Description Protocol (SDP) Media Streams,” July 2007.) [I-D.ietf-mmusic-securityprecondition], as Security Preconditions is not widely implemented and requires significant signaling overhead.

R6: The media security key management protocol MUST provide protection against passive attacks on the media path.

R7: The media security key management protocol MUST provide protection against passive attacks of a SIP proxy that is legitimately routing SIP messages.

R8: The media security key management protocol MUST be able to support perfect forward secrecy (PFS). The term PFS is 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.

R9: The media security key management protocol MUST support negotiation of SRTP cipher suites without incurring per-algorithm computational expense. This allows an offer to be built without incurring computational expense for each algorithm.

R10: If SRTP keying is performed over the media path, the keying packets MUST NOT pass the RTP validity check defined in Appendix A.1 of [RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” July 2003.).

R11:
The media security key management protocol that utilizes expensive cryptographic computations SHOULD offer the ability to resume previous sessions in an efficient way.

R12: The media security key management protocol MUST NOT require 3rd parties to sign certificates.
This requirement points to the fact that a global PKI cannot be assumed and opportunistic security approaches should be considered as part of the solution.

R13: The media security key management protocol SHOULD use algorithms that allow FIPS 140-2 (NIST, “Security Requirements for Cryptographic Modules,” June 2005.) [FIPS-140-2] certification.
Note that the United States Government can only purchase and use crypto implementations that have been validated by the FIPS-140 (NIST, “Security Requirements for Cryptographic Modules,” June 2005.) [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."[cryptval] (NIST, “Cryptographic Module Validation Program,” December 2006.)

Some commercial organizations, such as banks and defense contractors, also require or prefer equipment which has validated by the FIPS-140 process.

R14: The media security key management protocol SHOULD be able to associate the signaling exchange with the media traffic. 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 DH secrets and validating signatures can be a difficult task depending on the device capabilities. Hence, in the case of forking, it is not desirable to perform a DH or PK 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 needs to be completed. This might be done by using the transport address indicated in the SDP, although NATs can complicate this association.

Allowing such an association also allows the SDP offerer to avoid performing CPU-consuming operations (e.g., DH or public key operations) with attackers that have not seen the signaling messages.

R15: The media security key management protocol SHOULD allow to start with RTP and then upgrade to SRTP.

R16: The media security key management protocol SHOULD NOT introduce new denial of service vulnerabilities.

R17:
The media security key management protocol SHOULD require the adversary to have access to the data as well as the signaling path for a successful attack to be launched. An adversary that is located only along the data or only along the signaling path MUST NOT be able to successfully mount an attack. A successful attack refers to the ability for the adversary to obtain keying material to decrypt the SRTP encrypted media traffic.

R18: If two parties share an authentication infrastructure that has 3rd parties signing certificates, they SHOULD be able to make use of it.

R19: The media security key management protocol MUST provide crypto-agility.

R20: The media security key management protocol MUST protect cipher suite negotiation against downgrading attacks.

R21: The media security key management protocol MUST allow a SIP User Agent to negotiate media security parameters for each individual session.

R22: The media security key management protocol SHOULD allow establishing SRTP keying between different call signaling protocols (e.g., between Jabber, SIP, H.323, MGCP)

R23: The media security key management protocol SHOULD support recording of decrypted media.
Media recording may be realized by an intermediate nodes. An example for those intermediate nodes are devices, which could be used in banking applications or for quality monitoring in call centers. Here, it must be ensured, that the media security is ensured by the intermediate nodes on the connections to the involved endpoints as originally negotiated. The endpoints need to be informed that there is an intermediate device and need to cooperate. A solution approach for this is described in [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,” October 2008.).

R24: The media security key management protocol SHOULD NOT allow end users to determine whether their end-to-end interaction is subject to lawful interception.

R25: The media security key management protocol MUST work when there are intermediate nodes, terminating or processing media, between the end points.

R26: The media security key management protocol MUST consider termination of media security in a PSTN gateway.
A typical case of using media security is the one 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.

If media security (e.g., SRTP protection) is employed in this kind of gateway-setting, then media security and the related key management needs to be terminated at the 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 end-to-end between the communicating entities.

---

5. Requirements Classification

TOC

An adversary might be located along

1. the media path,
2. the signaling path,
3. the media and the signaling path.

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

Furthermore, it is reasonable to consider the capabilities of the adversary. We also have different types of adversaries, namely

a. active adversary

b. passive adversary

Note that the adversary model for (a) and (b) also assumes the attacker being able to control SIP signaling entities.

With respect to item (a) an adversary may need to be active with regard to the key exchange relevant information traveling along the data or the signaling path.

Some of the deployment variants of the media security key management proposals under considerations do not provide protection against man-in-the-middle adversaries under certain conditions, for example when SIP signaling entities are compromised, when a global PKI is missing or pre-shared secrets are not exchanged between the end points prior to the protocol exchange.

Based on the above-mentioned considerations the following classifications can be made:

Class I:
Passive attack on the signaling and the data path sufficient to reveal the content of the media traffic.

Class II:

> Active attack on the signaling path and passive attack on the data path to reveal the content of the media traffic.

Class III:

> Active attack on the signaling and the data path necessary to reveal the content of the media traffic.

Class IV:

> Active attack is required and will be detected by the end points when adversary tampers with the messages.

For example, Security Descriptions falls into Class I since the adversary needs to learn the Security Descriptions key by processing a signaling message at a SIP proxy (assuming that the adversary is in control of the SIP proxy). Subsequent media traffic can be decrypted with the help of the learned key.

As another example, DTLS-RTP falls into Class III when DTLS is used a public key based ciphersuite with self-signed certificates and without SIP Identity. 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.

An attack is not successful when SIP Identity is used, the adversary is not between the SIP UA and its Authentication Service (or at the Authentication Service), both end points are able to verify the digital signature (of the SIP Identity) and are able to validate the corresponding certificates.

---

6. Security Considerations

TOC

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

---

7. IANA Considerations

TOC

This document does not require actions by IANA.

TOC

8. Acknowledgements

The authors would like to thank the participants of the two RTPSEC BoFs and the members of the RTPSEC mailing list. Further thanks to the following individuals for their specific suggestions which improved this document: 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, Peter Schneider, Eric Rescorla, Srinath Thiruvengadam, Dave Ward, and Dan York.

Thanks also to Dragan Ignjatic (and our co-author, Steffen Fries) for their excellent MIKEY modes (Fries, S. and D. Ignjatic, “On the applicability of various MIKEY modes and extensions,” March 2008.) [I-D.ietf-msec-mikey-applicability] document, which formed the basis for the MIKEY comparisons.

9. References

TOC

9.1. Normative References

TOC

[FIPS-140-2]	NIST, “Security Requirements for Cryptographic Modules,” June 2005.
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[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 (TXT).
[RFC3262]	Rosenberg, J. and H. Schulzrinne, “Reliability of Provisional Responses in Session Initiation Protocol (SIP),” RFC 3262, June 2002 (TXT).
[RFC3264]	Rosenberg, J. and H. Schulzrinne, “An Offer/Answer Model with Session Description Protocol (SDP),” RFC 3264, June 2002 (TXT).
[RFC3711]	Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, “The Secure Real-time Transport Protocol (SRTP),” RFC 3711, March 2004 (TXT).
[cryptval]	NIST, “Cryptographic Module Validation Program,” December 2006.

9.2. Informative References

TOC

[I-D.barnes-sip-em-ps-req-sol]	Barnes, R. and M. Lepinski, "Early Media in SIP: Problem Statement, Requirements, and Analysis of Solutions," draft-barnes-sip-em-ps-req-sol-00 (work in progress), February 2007 (TXT).
[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 (TXT).
[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 (TXT).
[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 (TXT).
[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 (TXT).
[I-D.ietf-mmusic-sdp-capability-negotiation]	Andreasen, F., "SDP Capability Negotiation," draft-ietf-mmusic-sdp-capability-negotiation-13 (work in progress), March 2010 (TXT).
[I-D.ietf-mmusic-securityprecondition]	Andreasen, F. and D. Wing, "Security Preconditions for Session Description Protocol (SDP) Media Streams," draft-ietf-mmusic-securityprecondition-04 (work in progress), July 2007 (TXT).
[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-09 (work in progress), March 2008 (TXT).
[I-D.ietf-msec-mikey-ecc]	Milne, A., "ECC Algorithms for MIKEY," draft-ietf-msec-mikey-ecc-03 (work in progress), June 2007 (TXT).
[I-D.ietf-sip-certs]	Jennings, C. and J. Fischl, "Certificate Management Service for The Session Initiation Protocol (SIP)," draft-ietf-sip-certs-12 (work in progress), March 2010 (TXT).
[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 (TXT).
[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 (TXT).
[I-D.mcgrew-srtp-ekt]	McGrew, D., Andreasen, F., Wing, D., and L. Dondeti, "Encrypted Key Transport for Secure RTP," draft-mcgrew-srtp-ekt-06 (work in progress), October 2009 (TXT).
[I-D.mcgrew-tls-srtp]	Rescorla, E. and D. McGrew, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)," draft-mcgrew-tls-srtp-02 (work in progress), March 2007 (TXT).
[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 (TXT).
[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 (TXT).
[I-D.zimmermann-avt-zrtp]	Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP," draft-zimmermann-avt-zrtp-18 (work in progress), April 2010 (TXT).
[RFC3312]	Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)," RFC 3312, October 2002 (TXT, PS, PDF).
[RFC3388]	Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)," RFC 3388, December 2002 (TXT).
[RFC3550]	Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," STD 64, RFC 3550, July 2003 (TXT, PS, PDF).
[RFC3830]	Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing," RFC 3830, August 2004 (TXT).
[RFC4346]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," RFC 4346, April 2006 (TXT).
[RFC4474]	Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474, August 2006 (TXT).
[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 (TXT).

[RFC4568]	Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams," RFC 4568, July 2006 (TXT).
[RFC4650]	Euchner, M., "HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY)," RFC 4650, September 2006 (TXT).
[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 (TXT).
[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 (TXT).
[RFC4916]	Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)," RFC 4916, June 2007 (TXT).

Appendix A. Overview of Keying Mechanisms

TOC

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

TOC

A.1.1. MIKEY-NULL

TOC

MIKEY-NULL (Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, “MIKEY: Multimedia Internet KEYing,” August 2004.) [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

TOC

MIKEY-PSK (pre-shared key) [RFC3830] (Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, “MIKEY: Multimedia Internet KEYing,” August 2004.) 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

TOC

MIKEY-RSA (Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, “MIKEY: Multimedia Internet KEYing,” August 2004.)

[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.

A.1.4. MIKEY-RSA-R

TOC

MIKEY-RSA-R An additional mode of key distribution in MIKEY: MIKEY-RSA-R (Ignjatic, D., Dondeti, L., Audet, F., and P. Lin, “MIKEY-RSA-R: An Additional Mode of Key Distribution in Multimedia Internet KEYing (MIKEY),” November 2006.) [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 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.

A.1.5. MIKEY-DHSIGN

TOC

In MIKEY-DHSIGN (Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, “MIKEY: Multimedia Internet KEYing,” August 2004.) [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.

A.1.6. MIKEY-DHHMAC

TOC

MIKEY-DHHMAC (Euchner, M., “HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY),” September 2006.) [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 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.

A.1.7. MIKEY-ECIES and MIKEY-ECMQV (MIKEY-ECC)

TOC

ECC Algorithms For MIKEY (Milne, A., “ECC Algorithms for MIKEY,” June 2007.) [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.

A.1.8. Security Descriptions with SIPS

TOC

Security Descriptions (Andreasen, F., Baugher, M., and D. Wing, “Session Description Protocol (SDP) Security Descriptions for Media Streams,” July 2006.) [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

TOC

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

TOC

A.1.10. SDP-DH (expired)

SDP Diffie-Hellman (Baugher, M. and D. McGrew, “Diffie-Hellman Exchanges for Multimedia Sessions,” February 2006.) [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 (Peterson, J. and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP),” August 2006.) [RFC4474] and [RFC4474] (Peterson, J. and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP),” August 2006.).

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)

TOC

MIKEYv2 (Dondeti, L., “MIKEYv2: SRTP Key Management using MIKEY, revisited,” March 2007.) [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 (MIKEYv2 Inband (expired))).

A.2. Media Path Keying Technique

TOC

A.2.1. ZRTP

TOC

ZRTP (Zimmermann, P., Johnston, A., and J. Callas, “ZRTP: Media Path Key Agreement for Unicast Secure RTP,” April 2010.) [I-D.zimmermann-avt-zrtp] does not exchange information in the signaling path (although it's possible for endpoints to indicate support for ZRTP with "a=zrtp" in the initial Offer). 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 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. 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.

A.3. Signaling and Media Path Keying Techniques

TOC

A.3.1. EKT

TOC

EKT (McGrew, D., Andreasen, F., Wing, D., and L. Dondeti, “Encrypted Key Transport for Secure RTP,” October 2009.) [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

TOC

DTLS-SRTP (Rescorla, E. and D. McGrew, “Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP),” March 2007.) [I-D.mcgrew-tls-srtp] exchanges public key fingerprints in SDP [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,” July 2007.) 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 (Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," April 2006.) [RFC4346].

A.3.3. MIKEYv2 Inband (expired)

TOC

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

Appendix B. Evaluation Criteria - SIP

TOC

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

B.1. Secure Retargeting and Secure Forking

TOC

Retargeting and forking of signaling requests is described within section Section 3.2 (Retargeting and Forking). 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.

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 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).

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.

B.2. Clipping Media Before SDP Answer

TOC

Clipping media before receiving the signaling answer is described within section Section 3.1 (Clipping Media Before Signaling Answer). 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 Not clipped. Keys are exchanged in the media path without relying on the signaling path.

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

TOC

B.3. Centralized Keying

Centralized keying is described within section Section 3.3 (Shared Key Conferencing). The following builds upon this description.

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

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

Security Descriptions with SIPS Keying: Yes, if offerer is the mixer. Yes, if offerer is the participant.

Rekeying: Yes, with a Re-Invite.

Security Descriptions with S/MIME 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 Security Descriptions 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.

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.

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

B.4. SSRC and ROC

TOC

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[RFC4771] (Lehtovirta, V., Naslund, M., and K. Norrman,

"Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)," January 2007.). 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.

Appendix C. Evaluation Criteria - Security

TOC

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

C.1. Public Key Infrastructure

TOC

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 sip-certs (Jennings, C. and J. Fischl, "Certificate Management Service for The Session Initiation Protocol (SIP)," March 2010.) [I-D.ietf-sip-certs] might be used to obtain the certificate of a peer.

> Note: Even if SIP CERTs was deployed, the retargeting problem (Secure Retargeting and Secure Forking) 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-sip-certs] (Jennings, C. and J. Fischl, "Certificate Management Service for The Session Initiation Protocol (SIP)," March 2010.) 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.

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

Security Descriptions with SIPS PKI not used.

Security Descriptions with S/MIME PKI is needed for S/MIME. The offerer must obtain the intended target's public key and encrypt their SDP with that key. The answerer must obtain the offerer's public key and encrypt their SDP with that key.

SDP-DH PKI not used.

ZRTP PKI not used.

EKT PKI not used by EKT 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.

C.2. Perfect Forward Secrecy

TOC

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.

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

Security Descriptions with SIPS No PFS.

Security Descriptions with S/MIME No PFS.

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 achieved if the negotiated cipher suite includes an exponential or discrete-logarithmic key exchange (such as Diffie-Hellman or Elliptic Curve Diffie-Hellman (Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)," May 2006.) [RFC4492]).

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

C.3. Best Effort Encryption

TOC

Note: With the ongoing efforts in SDP Capability Negotiation (Andreasen, F., "SDP Capability Negotiation," March 2010.) [I-D.ietf-mmusic-sdp-capability-negotiation], the conclusions reached in this section may no longer be accurate.

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] (Wing, D. and C. Jennings, "Session Initiation Protocol (SIP) Offer/ Answer with Multipart Alternative," March 2006.) 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) (Mahy, R., "A Solution to the Heterogeneous Error Response Forking Problem (HERFP) in the Session Initiation Protocol (SIP)," March 2006.) [I-D.mahy-sipping-herfp-fix] causes problems if such non-multipart-capable endpoints were involved in the forking.
>
> **SDP Grouping** A new SDP grouping mechanism (following the idea introduced in [RFC3388] (Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)," December 2002.)) 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 (Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," March 2004.) [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 best effort encryption for each keying mechanism.

MIKEY-NULL No best effort encryption.

MIKEY-PSK No best effort encryption.

MIKEY-RSA No best effort encryption.

MIKEY-RSA-R No best effort encryption.

MIKEY-DHSIGN No best effort encryption.

MIKEY-DHHMAC No best effort encryption.

MIKEYv2 in SDP No best effort encryption.

Security Descriptions with SIPS No best effort encryption.

Security Descriptions with S/MIME No best effort encryption.

SDP-DH No best effort encryption.

ZRTP Best effort 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] (Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP," April 2010.)). Current implementations of ZRTP use probing.

EKT No best effort encryption.

DTLS-SRTP No best effort encryption.

MIKEY Inband No best effort encryption.

C.4. Upgrading Algorithms

TOC

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 computaional 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".

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 D. Out-of-Scope

TOC

Discussions concluded that key management for shared-key encryption of conferencing is outside the scope of this document. As the

priority is point-to-point unicast SRTP session keying, resolving shared-key SRTP session keying is deferred to later and left as an item for future investigations.

Authors' Addresses

TOC

	Dan Wing
	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.com
	Francois Audet
	Nortel
	4655 Great America Parkway
	Santa Clara, CA 95054
	USA
Email:	audet@nortel.com
	Brian Stucker
	Nortel
	2201 Lakeside
	Richardson, TX 75082
	USA
Email:	bstucker@nortel.com
URI:	http://www.linkedin.com/pub/bstucker

Full Copyright Statement

TOC

Intellectual Property