Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 27, 2007

D. Wing
Cisco
S. Fries
Siemens AG
H. Tschofenig
Nokia Siemens Networks

June 25, 2007

Requirements for a Media Security Key Management Protocol
draft-wing-media-security-requirements-04.txt

Status of this Memo

Copyright Notice

Abstract

A number of proposals have been published to address the need of securing media traffic. Different assumptions, requirements, and usage environments justify every one of them. This document aims to summarize the discussed media security requirements in order progress the work on identifying a small subset applicable to a large range of

deployment environments.

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

Table of Contents

1. Introduction

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. Once the list of requirements and architectural aspects have been investigated, the work on the protocol proposals can be progressed by identifying a small number of soltuions and to complete the protocol work.

This document is organized as follows. Section 2 introduces terminology, Section 3 provides an overview about possible call scenarios, Section 4 lists requirements for media security, Section 5 will provide a clustering of requirements to certain deployment environments to adress the problem that there might not be a single solution with universal applicability and Appendix A provides out-of-scope items and aspects for further discussion. The document concludes with a security considerations Section 6, IANA considerations Section 7 and an acknowledgement section in Section 8.

2. Terminology

The keywords "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 (referred as ', not its implementation or application.

3. Discussion of Call Scenarios

The following subsections describe call scenarios, which have been discussed elaborately. These call scenarios pose the most challenge to the key management for media data in cooperation with SIP signaling. The scenarios have already been described as part of the key management evaluation draft [I-D.wing-rtpsec-keying-eval], and are stated here to give a better insight in these discussion.

3.1. Clipping Media Before Signaling Answer

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 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], 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 [I-D.ietf-mmusic-securityprecondition], or the both ends implement ICE [I-D.ietf-mmusic-ice] and the answerer implements the reliable provisional response mechanism described in ICE. 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]. Fixes to early media might make the requirements to become obsolete.

3.2. Retargeting and Forking

In SIP, a request sent to a specific AOR but delivered to a different AOR is called a "retarget". A typical scenario is a "call forwarding" feature. In 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.

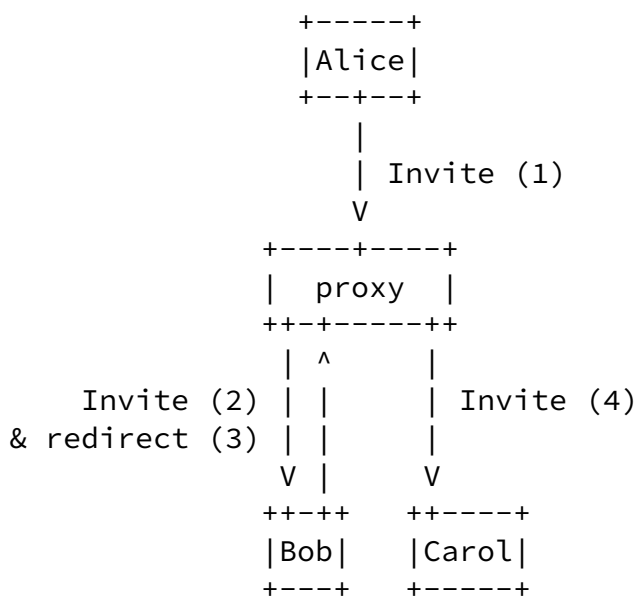

Figure 1: Retargeting

The mechanism used by SIP for identifying the calling party is SIP Identity [RFC3261]. However, due to SIP retargeting issues [I-D.peterson-sipping-retarget], SIP Identity can only identify the

calling party (that is, the party that initiated the SIP request). Some key exchange mechanisms predate SIP Identity and include their own identity mechanism. However, those built-in identity mechanism suffer from the same SIP retargeting problem described in the above draft. Going forward, it is anticipated that Connected Identity [I-D.ietf-sip-connected-identity] may allow identifying the called party. 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.

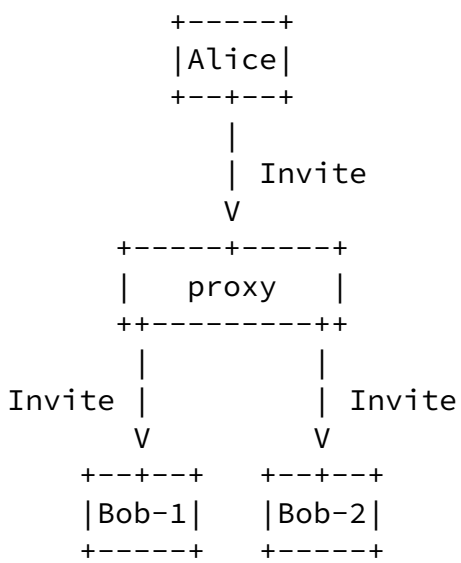

Figure 2: Forking

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], 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) [I-D.mahy-sipping-herfp-fix], is difficult to solve in SIP.

3.3. Shared Key Conferencing

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:

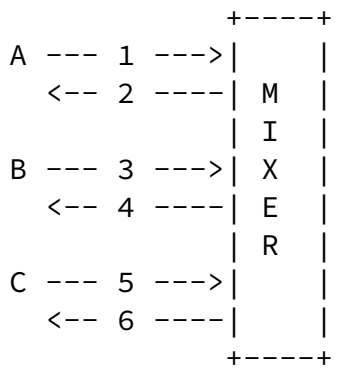

Figure 3: Centralized Keying

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.

4. Requirements

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 PRACK [RFC3262].

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 (or 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].

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 [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 [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]

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.

R15: 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.

R14: The media security key management protocol SHOULD allow to start with RTP and then upgrade to SRTP.

R15: The media security key management protocol SHOULD NOT introduce new denial of service vulnerabilities.

R16: 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.

R17: If two parties share an authentication infrastructure that has 3rd parties signing certificates, they SHOULD be able to make use of it.

R18: The media security key management protocol MUST provide crypto-agility.

R19: The media security key management protocol MUST protect cipher suite negotiation against downgrading attacks.

R20: The media security key management protocol MUST allow a SIP User Agent to negotiate media security parameters for each individual session.

R21: The media security key management protocol SHOULD allow establishing SRTP keying between different call signaling protocols (e.g., between Jabber, SIP, H.323, MGCP)

R22: 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].

R23: The media security key management protocol SHOULD NOT allow end users to determine whether their end-to-end interaction is subject to lawful interception.

R24: The media security key management protocol MUST work when there are intermediate nodes, terminating or processing media, between the end points.

R25: 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

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, SDES falls into Class I since the adversary needs to

learn the SDES key by progressing 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

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

The authors would like to thank the active participants of the RTPSEC BoF and on the RTPSEC mailing list. The authors would furthermore like to thank Wolfgang Buecker, Guenther Horn, Peter Howard, Hans-Heinrich Grusdt, Srinath Thiruvengadam, Martin Euchner, Eric Rescorla, Matt Lepinski, Dan York, Werner Dittmann, Richard Barnes, Vesa Lehtovirta, Colin Perkins, Peter Schneider, and Christer Holmberg for their feedback to this document. We would like to particularly thank Francois Audet for his work on the evaluation of various media security key exchange proposals.

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.

[cryptval] NIST, "Cryptographic Module Validation Program", December 2006, <http://csrc.nist.gov/cryptval/140-2APP.htm>.

9.2. Informative References

[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.

[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-16 (work in progress), June 2007.

[I-D.ietf-mmusic-securityprecondition] Andreasen, F. and D. Wing, "Security Preconditions for Session Description Protocol (SDP) Media Streams", draft-ietf-mmusic-securityprecondition-03 (work in progress), October 2006.

[I-D.ietf-sip-connected-identity]
Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", draft-ietf-sip-connected-identity-05 (work in progress), February 2007.

[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.

[I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.

[I-D.wing-rtpsec-keying-eval]
Audet, F. and D. Wing, "Evaluation of SRTP Keying with SIP", draft-wing-rtpsec-keying-eval-02 (work in progress), February 2007.

[I-D.wing-sipping-srtp-key]
Wing, D., "Disclosing Secure RTP (SRTP) Session Keys with a SIP Event Package", draft-wing-sipping-srtp-key-00 (work in progress), February 2007.

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

Appendix A. Out-of-Scope

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

Dan Wing
Cisco
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

http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).