Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 19, 2007

D. Wing
Cisco
S. Fries
Siemens AG
H. Tschofenig
Siemens Networks GmbH & Co KG

October 16, 2006

Media Security Requirements
draft-wing-media-security-requirements-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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

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

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

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

This Internet-Draft will expire on April 19, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

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.

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 larger deployment and the available 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 behavior and, to some extend, considered necessary protocol work to ensure that the developed protocols indeed meet the previously envisioned needs in the global Internet.

This document aims to summarize the discussed media security requirements. Once a 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, Section 6 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 Section 5 provides out-of-scope items and aspects for further discussion. The document concludes with a security considerations Section 7, IANA considerations Section 8 and an acknowledgement section in Section 9.

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].

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 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.

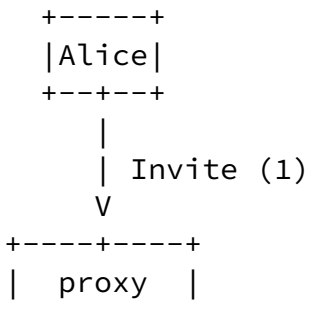
For key exchange mechanisms which send the answerer's key in SDP, a SIP provisional response [RFC3261] such as 183 (session progress) is useful. However the 183 messages aren't reliable unless both the calling and called endpoint support PRACK [RFC3262], use TCP across all SIP proxies, implement Security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

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

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 the figure below, Alice sends an Invite in step 1 which is sent to Bob in step 2. Bob responds with a redirect (SIP response code 3xx) pointing to Carol in step 3. This redirect typically does not propagate back to Alice but only goes to a proxy (i.e., the retargeting proxy) which sends the original Invite to Carol in step 4.



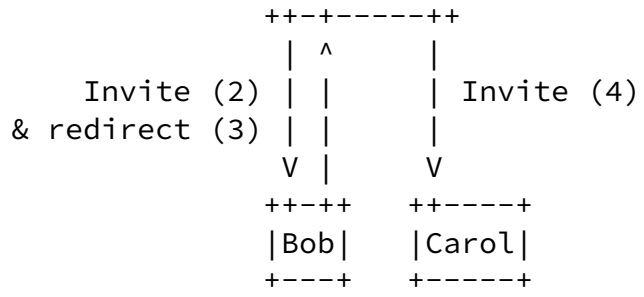
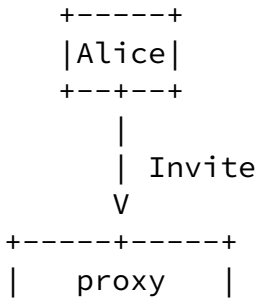


Figure 1: Retargeting

Successful use of SRTP requires strongly identifying both calling party and the called party. The mechanism used by SIP for identifying the calling party is SIP Identity [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.



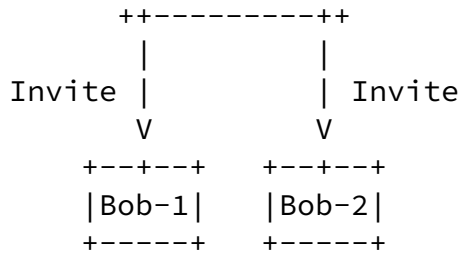


Figure 2: Forking

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 isn't desirable to require ICE to accomplish this association. The table below analyzes if it is possible for an offerer to associate the media stream with each SDP answer, without using ICE.

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

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

Retargeting securely introduces a more significant problem. With retargeting, the actual recipient of the request is not the original recipient. This means that if the offerer encrypted material (such as the session key or the SDP) using the original recipient's public key, the 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-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) [draft-mahy-sipping-herfp-fix] is a complicated problem 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:

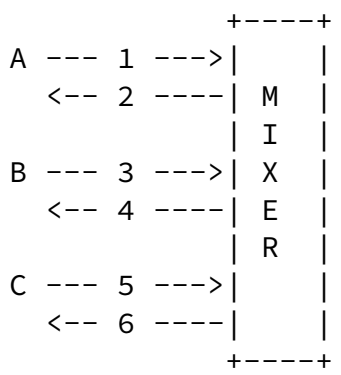


Figure 3: Centralized Keying

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

To support a single outbound session (scenario d), the mixer has to dictate its encryption key to the participants. Some keying mechanisms allow the transmitter to determine its own key, and others allow the offerer to determine the key for the offerer and answerer. Depending on how the call is established, the offerer might be a participant (such as a participant dialing into a conference bridge) or the offerer might be the mixer (such as a conference bridge calling a participant). The use of offerless Invites may help some keying mechanisms reverse the role of offerer/answerer. A difficulty, however, is knowing a priori if the role should be reversed for a particular call.

4. Requirements

R1: Forking and retargeting MUST work with all end-points being SRTP.

R2: Forking and retargeting MUST allow establishing SRTP or RTP with a mixture of SRTP- and RTP-capable targets.

R3: With forking, only the entity to which the call is finally established, MUST get hold of the media encryption keys.

R5: A solution SHOULD avoid clipping media before SDP answer without additional signalling.

R6: A solution MUST provide protection against passive attacks.

R7: A solution MUST be able to support Perfect Forward Secrecy.

R8: A solution MUST support algorithm negotiation without incurring per-algorithm computational expense.

R9: A solution MUST support multiple cipher suites without additional computational expense.

R10: Endpoint identification when forking. The Offerer must be able to associate answer with the appropriate flow endpoint. In case of forking one might not want to perform a DH with every party but instead to associate the SDP response with the right end point. This is a performance related requirement.

R11: A solution MUST NOT require 3rd-party certs. If two parties share an auth infrastructure they should be able to use it.

5. Out-of-Scope and Discussion Items

The following aspects have been voted out-of-scope:

- Shared-key encryption for conferencing (Note: it may be matter of discussion, if shared key conferencing stays as out-of-scope.)

The following items are subject for further study:

- A solution SHOULD allow to start with RTP and then upgrade to SRTP.
- A solution SHOULD consider active attacks.
- From an architectural point of view solutions can exchange key exchange messages along the media path, along the signaling path or on both paths. A solution SHOULD operate along the media path and the signaling path.
- A solution SHOULD support the possibility to protect non-RTP-based data traffic.
- A solution MUST protect cipher suite negotiation against downgrading attacks.
- A solution MUST allow a SIP UE to negotiate media security parameters for each individual session.

The following two requirements are typically raised by other SDOs and might be relevant in this context:

- A solution SHOULD support media recording.

- o A solution SHOULD NOT allow end users to determine whether their end-to-end interaction is subject to lawful interception.

6. Clustering Requirements according to Environments

It is not possible to fulfill all requirements presented in Section 4 to be useful in all environments. This section aims to describe the usage environments and to cluster solutions with respect to these environments to distil a small set of solutions fulfilling these requirements.

[Editor's Note: Text will be provided in a future version of this document.]

7. Security Considerations

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

8. IANA Considerations

This document does not require actions by IANA.

9. Acknowledgements

The authors would like to thank the active participants of the RTPSEC BOF and Wolfgang Buecker, Guenther Horn, Peter Howard, Hans-Heinrich

Grusdt, Srinath Thiruvengadam and Martin Euchner for their feedback to this document.Especially thank to Francois Audet for the work on the key management evaluation.

10. References

10.1. Normative References

[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.

10.2. Informative References

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-11 (work in progress), October 2006.

[I-D.ietf-mmusic-securityprecondition]
Andreasen, F. and D. Wing, "Security Preconditions for Session Description Protocol (SDP) Media Streams", draft-ietf-mmusic-securityprecondition-02 (work in progress), June 2006.

[I-D.ietf-sip-connected-identity]
Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", draft-ietf-sip-connected-identity-02 (work in progress), October 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-01 (work in progress),

June 2006.

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
Siemens Networks GmbH & Co KG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be

found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at 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).