MIKEYv2: SRTP Key Management using MIKEY, revisited
draft-dondeti-msec-rtpsec-mikeyv2-00

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 December 21, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

We specify a new version of the Multimedia Internet Keying (MIKEY) protocol, MIKEYv2, tailored for the secure real-time transport protocol (SRTP). MIKEYv2 uses the same payloads and message formats as MIKEY; it supports mode negotiation, uses either SDP in the signaling path or UDP for transport, does not require time synchronization, and does not need SIP support for accessing end-point credentials.

Editor’s note
This is a strawman proposal of the MIKEYv2 protocol. The protocol as specified in this document is broken in several different ways: the security of the protocol, the payload formats of RFC 3830 not quite agreeing with the vision of the author to begin with and perhaps in other ways. This will first be discussed with interested parties to hash out the details and a revision will be submitted at the latest before the Montreal IETF meeting in July 2006. Please feel free to send reviews to ldondeti@qualcomm.com, but it might be better to hold judgement until the revision appears.

Table of Contents

1. Introduction ............................................. 3
2. Terminology ............................................. 4
3. SRTP key management design goals, constraints and use cases . 4
   3.1. SRTP use cases ........................................ 5
   3.2. SRTP key management design goals and constraints ....... 5
   3.3. SRTP cryptographic context ............................ 7
   3.4. SRTCP crypto context .................................. 8
4. MIKEYv2 outline .......................................... 8
5. MIKEYv2 protocol details ................................. 9
   5.1. Initial exchange ...................................... 9
   5.2. Create crypto context exchange ........................ 11
   5.3. Rekeying ............................................. 11
   5.4. Transporting MIKEYv2 Messages ....................... 11
6. Security Considerations .................................. 12
7. IANA Considerations ..................................... 12
8. Acknowledgments ......................................... 12
9. References .............................................. 12
   9.1. Normative References .................................. 12
   9.2. Informative References ............................... 12
Author’s Address ........................................... 14
Intellectual Property and Copyright Statements ............ 15
1. Introduction

The Multimedia Internet Keying (MIKEY) [RFC3830] protocol is a key management protocol for SRTP. It’s a half or one round trip authentication and key delivery/establishment protocol that uses timestamps for replay protection, and asymmetric or symmetric keys for entity authentication. MIKEY is a general purpose efficient key management protocol for real-time applications, but has been designed with SRTP key establishment as the primary application.

MIKEY supports point-to-point as well as group key establishment and is the protocol of choice in other standards development organizations; for instance, the 3GPP Multimedia Broadcast and Multicast Service (MBMS) uses MIKEY for session key establishment via unicast and traffic key establishment and update via broadcast. 3GPP uses the IANA assigned UDP port 2269 for MIKEY transport. The Open Mobile Alliance (OMA)’s Broadcast (BCAST) specification uses MIKEY to transport the long and short term key messages.

However, several shortcomings of MIKEY have been identified, especially on its applicability to general purpose key management for VoIP application.

MIKEY has too many modes and no real support for mode negotiation.

While MIKEY can finish in half or one round trip, it requires time synchronization to do so.

MIKEY, as specified in RFC 3830 [RFC3830] requires SDP for transport. This is disadvantageous for MIKEY modes that require more than one message.

There are two MIKEY modes that require one message:

The RSA mode requires that the Initiator of the protocol know the identity and certificate of the recipient, and the PSK mode requires that the Initiator share a PSK with the Responder. This is simply not a practical assumption; furthermore, in case of SIP forking and call forwarding, MIKEY-RSA and MIKEY-PSK modes do not work.

There are MIKEY modes that can handle SIP forking and call forwarding; however, they cannot handle RTP early media very well. While there are solutions like security preconditions to help alleviate the early media case, those solutions might not be widely deployed or even if deployed will cause unnecessary delay in SRTP context establishment.
The discussion so far provides sufficient motivation to develop alternative protocols for SRTP. To that end several proposals have been made: zRTP, DTLS-SRTP are among them. While these solve some of the problems of MIKEY discussed earlier, introduce issues of their own and leave open several problems. For instance,

zRTP requires a large number of round trips, uses RTP to carry key management messages, and supports SRTP context establishment for unicast communication alone. MIKEY messages can just as easily be carried within RTP, whereas MIKEY requires few round trips and supports unicast and group key management.

DTLS-SRTP uses UDP for transport, but has limited support for SRTP context establishment. For instance, each SRTP stream requires a separate DTLS session. Like zRTP, DTLS-SRTP supports point-to-point communication only.

With this background, we propose MIKEYv2, extending MIKEY along the lines of the IKEv2 [RFC4306], given the lessons learned in the process of implementing and deploying MIKEY.

Below is a list of properties of MIKEYv2:

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In addition, we use the terminology in the MIKEY [RFC3830] and SRTP [RFC3711] specifications. Furthermore, we define the following terms:

3. SRTP key management design goals, constraints and use cases

The primary goal of SRTP key management is to establish the cryptographic context for SRTP encapsulation. In the rest of this document, we refer to this as the SRTP crypto context. The information includes, SRTP encryption and integrity protection keys, cryptographic algorithms used, key lengths, initialization vectors (IVs), salts and identifiers, and replay protection counters and state information. The key management protocol is expected to
bootstrap the SRTP crypto context, and so we delve into the details of these parameters and explore how communicating parties might be able to arrive at sharing the same crypto context. Note that the RTP traffic may be flowing between two parties or from one to two or more parties. In the following, we list SRTP use cases, design goals and constraints, and describe SRTP and SRTCP cryptographic context.

3.1. SRTP use cases

We identify three simple use cases:

Unicast: In the first, there are one or more RTP sessions between two communicating parties and session keys need to be derived for them.

One-to-many Conferencing: In the second, there are one or more one-to-many RTP sessions, all from one sender to two or more receivers. Key distribution for such use cases, where one speaker is delivering a lecture to an audience is of interest.

Many-to-many Conferencing: The final use case is multi-party multimedia conferencing, where two or more speakers are originators of RTP streams (one or more each) and two or more receivers are recipients of those streams.

3.2. SRTP key management design goals and constraints

A key management protocol for SRTP should meet the following design goals and constraints; [I-D.wing-rtpsec-keying-eval] provides a more detailed description of some of these issues and how various proposed key management protocols for SRTP address them.

Identity and Credential Transport: The protocol should allow the Initiator and Responder to send each other’s identities and credentials to the other parties. This was for instance absent in the MIKEY-RSA mode.

Mode Negotiation: The communicating parties should be able to negotiate the mode of authentication.

Algorithm Agility: Cryptographic algorithm negotiation support to help the communicating parties negotiate the use of the most secure algorithms they both support. Note that in group communication algorithm negotiation may not be possible; in that case algorithm agility support is defined as the ability of the group key manager to transition from one suite of algorithms to another.
Support for Multiple RTP Sessions: It is desirable for the key management protocol to establish crypto context for more than one RTP session between the same communicating parties in one run of the protocol.

Unicast and Group Communication Support: It is desirable for the key management protocol to support or be extensible to group security context establishment. Whereas the predominant application in circa 2006 might be point-to-point voice communications, it is unattractive to having to develop a different key management protocol for unicast communication and another for group communication. Since unicast communication is the more popular mode of communication, it makes sense to optimize for that use case.

Rekeying Support: It is desirable for the key management protocol to support rekeying in an efficient manner. In case of unicast communication, rekeying might finish faster than initial crypto context establishment process; more specifically, rekeying might finish in a single round trip, whereas initial run of the protocol might take two or more round trips. In case of group communication, rekeying might allow communication of a new group key with a single message from the group key manager to the members.

Transport Requirements: SRTP key management messages can be transported in one of three different ways: in the first, the messages are sent as part of SDP over SIP. In that case, the return message may arrive later than the media resulting in clipping. This is because the media path is for the most part faster than the signaling path, due to the many intermediate entities involved in the signaling path. The second method of transport is to use the bearer path. While the bearer path transport might be more efficient than signaling path, the first message may have to come from the answerer as opposed to the offerer (this is because the offerer might not always know the address of the answerer and because UDP port control may not allow communications to pass through until the first signaling message has made it to the answerer from the offerer). The third method of transport is to send the key management messages as part of RTP or RTCP packets, for instance in extension headers or with a new transform to the RTP or RTCP packets.

SIP Constraints: In the following we describe some SIP related constraints and design considerations:
Clipping Media Before SDP Answer: RTP packets take a direct route between the communicating parties, whereas the SIP packets pass through various servers and gateways, resulting in RTP packets arriving before SIP packets. Thus it is possible for RTP packets from the answerer to the offerer to arrive before the SDP answer via the signaling path, carrying a key management protocol message required to decipher the RTP packets. This results in the offerer to be not be able to play the media; this is defined as clipping. It is desirable that the key management protocol avoids clipping without the help of external mechanisms.

Secure Retargeting and Secure Forking: In cases such as call forwarding, a SIP message may reach a different party than specified either in the SIP message or the key management message. In most practical scenarios, it is ok to continue secure session establishment with the new peer as long as the peer is correctly identified and has appropriate credentials. In forking the messages reach multiple parties and the multiple parties might respond resulting potentially in multiple sessions.

Perfect-forward Secrecy: In some applications, perfect forward secrecy, defined as the property that past short-term keys or session keys are not compromised due to a compromise of the long-term keys or credentials, may be a desirable property.

Infrastructure Support: For some applications, a center or server may have a PSK with each potential end-user, or there may be a PKI support, but these assumptions are not valid in general. The key management protocol should take this into account.

3.3. SRTP cryptographic context

The SRTP specification [RFC3711] identifies transform dependent and transform independent parameters that comprise the crypto context. The transform-dependent parameters are as follows:

- encryption algorithm, e.g., AES-CTR, AES-f8, and associated key length

- integrity protection transform, e.g., TESLA or algorithm, e.g., HMAC-SHA1, associated key length and output length (e.g., MAC/ICV truncation)

- key derivation parameters
input for IV formation

The transform-independent parameters are listed below:

32-bit unsigned rollover counter (RoC), which records how many times the 16-bit RTP sequence number has been reset to zero after passing through 65,535 ($2^{16}-1$),

for each master key, an SRTP stream MAY use the following associated values:

- a master salt, to be used in the key derivation of session keys. Note that the master salt, MUST be random, but MAY be public

- an integer in the set {1,2,4,...,2^{24}}, the "key derivation rate"; the key management protocol may leave this unspecified and in that case the key derivation rate is assumed to be zero

- a master key identifier (MKI) value to identify the SRTP crypto context

The key management system may also specify the lifetime of the crypto context with a range of SRTP packet indices, From and To. The SRTP packet index is a 48-bit value formed by concatenating the 32-bit RoC with the 16-bit RTP packet index.

3.4. SRTCP crypto context

SRTCP by default shares the crypto context with SRTP, except there is no need to establish the rollover counter via key management as the RTCP index is explicitly carried in each SRTCP packet,

A cryptographic context SHALL be uniquely identified by the triplet context identifier:

context id = < SSRC, destination network address, destination transport port number >

where the destination network address and the destination transport port are the ones in the SRTP packet. It is assumed that, when presented with this information, the key management returns a context with the information as described in Section 3.2.

4. MIKEYv2 outline
The proposal is to revise MIKEY with the intent of adding mode negotiation and removing the time synchronization requirement, and specify MIKEYv2. In addition, we will specify UDP transport for MIKEYv2, borrowing from the original proposal, which at one time did contain the semantics for UDP transport.

MIKEYv2 reuses MIKEY payloads and introduces as few new payloads as possible to facilitate the revised design and the new features. MIKEYv2 messages use version number 0x02 in the common HDR payload specified in RFC3830. Version number 0x02 is reserved for messages described in this specification. Reuse of that version number is allowed only with a revision of this specification.

MIKEYv2 takes two round trips to complete and establishes unicast and optionally group SRTP and/or SRTCP crypto contexts. We reuse the key derivation and traffic key containers defined in RFC3830. The payloads and message structure while retained, are essentially part of a new key management protocol and need a fresh security analysis.

MIKEYv2 is a DH-based key management protocol based on SIGMA. In the first round trip, the communicating parties learn each other’s identities, agree on a MIKEY mode (type of entity authentication primarily), MIKEY crypto algorithms, SRTP policy, and exchange nonces for replay protection. In the second round trip, they negotiate unicast and/or group SRTP crypto context for SRTP and/or SRTCP.

5. MIKEYv2 protocol details

MIKEYv2 has two sets of exchanges. The initial exchange consists of identity establishment, MIKEY mode and algorithm negotiation and the second exchange consists of SRTP and SRTCP crypto context establishment.

5.1. Initial exchange
MIKEYv2_INIT_EXCH message is as follows:

Initiator                       Responder
=========                       =========
HDR, RANDi, M-SPi, IDi,
[IDr], DHi         --->

<---  HDR, RANDr, M-SPr, IDr,
[CERTREQ,] DHR

Figure 1: MIKEYv2 negotiation exchange

MIKEYv2 is closely modeled after IKEv2 [RFC4306] and relies on the SIGMA protocol for its security. The payloads, at least most of them, are reused from the original MIKEY specification, in the interest of code reuse (and potential backward compatibility. This is for further discussion and study).

MIKEYv2_INIT_EXCH is a Diffie-Hellman exchange, which allows the two parties to establish an unauthenticated secure channel.

There is no identity protection as it is specified currently, but that can be added easily. SIGMA provides some identity protection to the Initiator’s or the Responder’s identities.

The M-SPi payloads allow mode and algorithm negotiation for the secure channel. These payloads are intended to be used to negotiate the algorithm used in generating the AUTH and KEMAC payloads of the MIKEYv2 SRTP Cryptographic Context Establishment Exchange or MIKEYv2_SRTP_CCE.

In the second message, the Responder can request that Certificates be used for entity authentication. The proposal is to allow negotiation of this via the M-SPi payload.

The RAND payloads provide replay protection and are used to provide entropy for key derivation in the unicast case.
5.2. Create crypto context exchange

MIKEYv2_SRTP_CCE message is as follows:

Unicast case:
===============

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>=========</td>
<td>=========</td>
</tr>
</tbody>
</table>

HDR, [CERTi,] [CERTREQ,]
SRTP-Spi, AUTH -->

<--- HDR, [CERTr,] SRTP-SPr
AUTH

Group key establishment:
===============

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Group Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>=========</td>
<td>================</td>
</tr>
</tbody>
</table>

HDR, [CERTi,] [CERTREQ,]
GCC-REQ, AUTH -->

<--- HDR, [CERTr,] SRTP-Spg
AUTH, KEMAC

Figure 2: MIKEYv2 SRTTP Crypto Context Establishment

The key material derived in the MIKEYv2_INIT_EXCH is used to protect the messages/payloads of MIKEYv2_SRTP_CCE. The purpose of this exchange is to authenticate the first exchange via the AUTH payloads computed in a manner similar to that in RFC 4306 [RFC4306] and to negotiate or distribute the SRTP crypto context.

5.3. Rekeying

5.4. Transporting MIKEYv2 Messages

MIKEYv2 messages are transported via UDP using IANA assigned port 2269.

6. Security Considerations
7. IANA Considerations

Several IANA registrations may be required, include MIKEY version number and new payload types. Detailed instructions to IANA will be included in a future version.

8. Acknowledgments

This work benefited from discussions with various folks at the IETF, among them are Flemming Andreasen, Francois Audet, Rolf Blom, Ran Canetti, Steffen Fries, Dragan Ignjatic, Cullen Jennings, David McGrew, Karl Norrman, Jon Peterson, Rohan Mahy, and Dan Wing. Note that these folks may not necessarily be endorsing the MIKEYv2 protocol; in fact, it is plausible many of them do not even like the protocol.

9. References

9.1. Normative References


9.2. Informative References


[I-D.ietf-msec-mikey-dhhmac]


Author’s Address

Lakshminath Dondeti
QUALCOMM, Inc.
5775 Morehouse Dr
San Diego, CA
USA

Phone: +1 858-845-1267
Email: ldondeti@qualcomm.com
Intellectual Property Statement

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.

Disclaimer of Validity

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.

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.

Acknowledgment

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