Network WG
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2007

L. Dondeti
QUALCOMM, Inc.

March 5, 2007

MIKEYv2: SRTP Key Management using MIKEY, revisited
draft-dondeti-msec-rtpsec-mikeyv2-01

Status of this Memo

Copyright Notice

Abstract

The Multimedia Internet Keying (MIKEY) protocol is a general purpose key management protocol; it is used especially for key management for secure RTP. We specify a couple of variations of that protocol to support mode negotiation, media path key establishment and other assorted requirements.

Table of Contents

1. Introduction 3
1.1. Motivation to Designing MIKEYv2 4
2. Terminology 5
3. SRTP Key Management Design Goals, Constraints and Use Cases . 5
3.1. SRTP Use Cases 5
3.2. SRTP Cryptographic Context 6
3.3. SRTCP Crypto Context 6
4. MIKEYv2 Outline 7
5. MIKEYv2 Protocol Details: Option 1 8
5.1. Initial Exchange 8
5.2. Create Crypto Context Exchange 9
6. MIKEYv2 Protocol Details: Option 2 10
6.1. MIKEY Mode Negotiation Exchange 10
6.2. MIKEY-RSA/PSK Exchange 10
7. Transporting MIKEYv2 Messages 11
8. Security Considerations 11
9. IANA Considerations 11
10. Acknowledgments 11
11. References 12
11.1. Normative References 12
11.2. Informative References 12
Author's Address 13
Intellectual Property and Copyright Statements 14

1. Introduction

The Multimedia Internet Keying (MIKEY) [RFC3830] protocol is a general purpose key management protocol for real-time applications, especially 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 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.

It requires time synchronization for replay protected.

MIKEY, as specified in RFC 3830 [RFC3830] requires SDP for transport.

MIKEY-RSA mode requires that the Initiator of the protocol know the identity and certificate of the recipient. This mode does not handle SIP forking well.

MIKEY-PSK mode requires that the Initiator share a PSK with the Responder. This is simply not a practical assumption. This mode also does not handle SIP forking well.

MIKEY-RSA-R does not handle early media well. Early media may arrive before the SDP answer arrives.

Next, after some debate and discussion at the IETF, there is a consensus on some of the requirements for a common key management protocol for an application such as VoIP so there is a chance for cross-domain interoperability. The idea is to come up with a single protocol for key management for SRTP.

However, given the variety of constraints and use cases, it may not be possible to have a single universal key management protocol for

SRTP. Many of the devices that will need to implement the protocol are resource-constrained and some of them have one of the candidate protocols or their close cousins already implemented. There may be resistance to implement another protocol. Next some of the requirements may force resource-intensive computations, especially when there is SIP forking; a PSTN gateway may not be able to perform several DH computations per session.

1.1. Motivation to Designing MIKEYv2

There are at least two candidates for key management for SRTP, namely DTLS-SRTP and zRTP other than MIKEYv2. Considering the goal of specifying a single protocol, it makes sense to not design a new protocol. So the question is why design MIKEYv2? We consider that question in detail below:

First, MIKEY [RFC3830] was designed with SRTP as the primary application and has taken into account a number of design considerations. It is as good a candidate for reuse as any. None of the shortcomings listed earlier are inherent to MIKEY. In the end, just as any other candidate key management protocol it can be extended to meet the requirements at hand.

Next, MIKEY is the key management protocol for SRTP for other use cases such as broadcast key management. MIKEY is (planned to be) implemented in a smartcard, which is typically the device with which 3GPP and 3GPP2 operators share credentials with. The question may be whether it is difficult to implement TLS or DTLS on the smartcard. Difficult? Yes. Improbable, no! There are known implementations of TLS on a smartcard. It is definitely wasteful to have to implement two different protocols for key management for SRTP on the same device, and especially on a smartcard.

Finally, the current designs of some of the candidate protocols seem to indicate that negotiation of SRTP parameters may have to be split between the key management protocol and SDP. It is not clear whether there is an inherent shortcoming in the key management protocol or not. Furthermore, session reestablishment semantics seem less than optimal. There is also no support for group keying; whereas shared key conferencing is not a consensus requirement, broadcast/multicast using SRTP is a known use case today. Thus, it may be better to explore other options for key management for SRTP.

With this background, we propose MIKEYv2. We consider two possible designs: the first is to extend MIKEY along the lines of the IKEv2 [RFC4306], taking into account the lessons learned in the process of

implementing and deploying MIKEY. The second is to simply add mode negotiation to base MIKEY exchanges limiting modifications to MIKEY to an absolute minimum.

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.

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 group communication: In the second, there are one or more one-to-many RTP sessions, all from one sender to two or more receivers.

Many-to-many group communication: 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 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; integrity algorithm, e.g., HMAC-SHA1, associated key length and output length (e.g., MAC/ICV truncation)

key derivation parameters (e.g., PRF algorithm)

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 cast 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.

3.3. 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

MIKEYv2 supports mode negotiation, allows fast session reestablishment using reduced roundtrip exchanges, but does not require time synchronization.

MIKEYv2 runs over UDP (reusing the port number assigned for MIKEY) or over RTP/RTCP.

It may be plausible to specify starting MIKEYv2 over the signaling path and resume it via the media path (Steffen Fries talked about this at various times).

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/or 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.

For fast session reestablishment, it is plausible to use one of the RFC 3830 MIKEY exchange.

MIKEYv2 can also take advantage of the SAS technique introduced by zRTP or the certificate fingerprint transport via SDP as described in the context of DTLS-SRTP.

We explore two possible approaches to designing MIKEYv2:

MIKEYv2 is an authenticated DH 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, 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.

The second option is to simply add mode negotiation to MIKEY exchanges.

5. MIKEYv2 Protocol Details: Option 1

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:

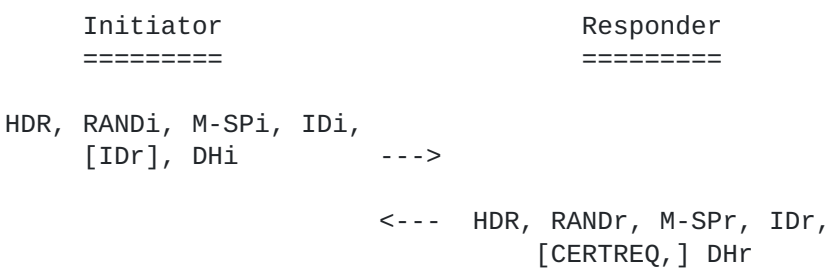

Figure 1: MIKEYv2 Policy Negotiation Exchange

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 MIKEY 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 paylods 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 paylods 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:

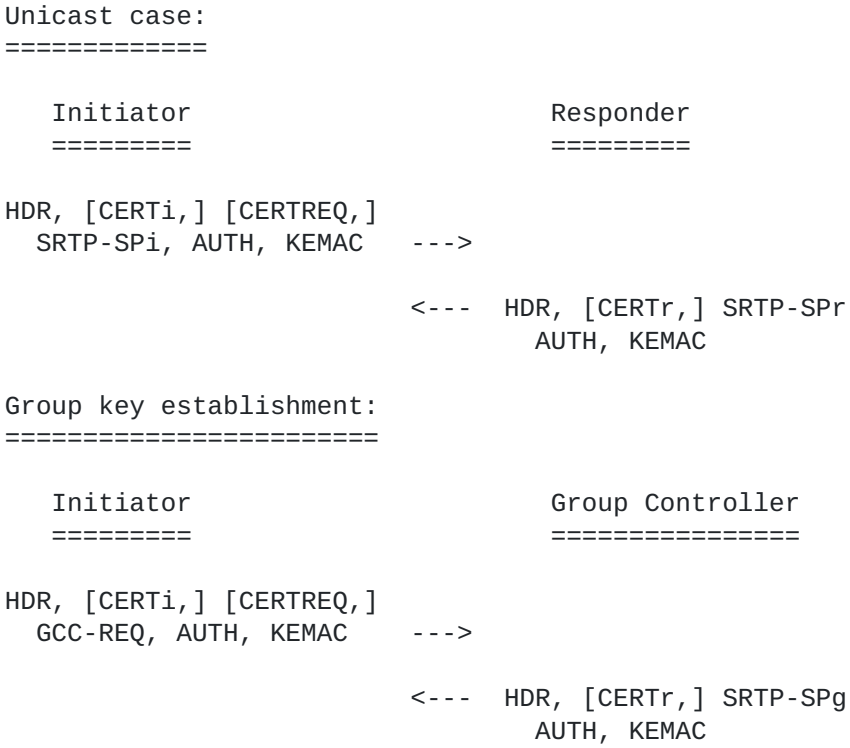

Figure 2: MIKEYv2 SRTP Crypto Context Establishment

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 via the SRTP-SP payloads. The KEMAC payloads in the unicast case do not necessarily contain keys, but the MAC portion of KEMAC integrity protects the entire message. The KEMAC payload sent by the Group Controller MUST contain keys.

6. MIKEYv2 Protocol Details: Option 2

6.1. MIKEY Mode Negotiation Exchange

MIKEYv2_MODE_NEG_EXCH message is as follows:

```
     Initiator                      Responder
     =========                      =========

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

                         <---  HDR, RANDr, M-SPr, IDr,
                                   [CERTREQ,] DHr
```

Figure 3: MIKEYv2 Mode Negotiation Exchange

6.2. MIKEY-RSA/PSK Exchange

MIKEYv2_CCE2 message is as follows:

```
     Initiator                      Responder
     =========                      =========
HDR, T, RAND, {SP}, KEMAC,
    [CHASH], [PKE, SIGNi]        --->

                                 <---    HDR, T, IDr, [V],
                                             [PKE, SIGNr]
```

Figure 4: MIKEYv2 SRTP Crypto Context Establishment Exchange

The idea here is to authenticate the initial exchange as part of the SIGNx payload and provide a proof of possession of the key derived using the DH exchange via the KEMAC and the V payload. These processing semantics are slightly different from that of RFC 3830. Note that these exchanges are shown to given an idea on how MIKEY may be reused; the details are TBD. The T payload contains a sequence number instead of a timestamp (note that the 3GPP MBMS specification also uses a sequence number instead of a timestamp in the exchange).

The CCE exchange may be used for 1 RT rekeying. The timestamp field contains a monotonically increasing sequence number (and serves a similar purpose as the message-id field of IKEv2 [RFC4306]).

7. Transporting MIKEYv2 Messages

MIKEYv2 messages may be transported via UDP using IANA assigned port 2269. Alternatively, MIKEYv2 messages may share the RTP/RTCP port with media/control packets. In the end, we may allow one option of this based on consensus.

8. Security Considerations

Security considerations of RFC 3830 apply. For Option 1, security considerations of RFC 4306 also apply.

9. 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.

10. Acknowledgments

This work benefited from discussions with various folks at the IETF, among them are Flemming Andreasan, 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.

11. References

11.1. Normative References

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

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

11.2. Informative References

[I-D.ietf-msec-mikey-ecc] Milne, A., "ECC Algorithms for MIKEY", draft-ietf-msec-mikey-ecc-01 (work in progress), October 2006.

[I-D.ietf-msec-mikey-rsa-r] Ignjatic, D., "An additional mode of key distribution in MIKEY: MIKEY-RSA-R", draft-ietf-msec-mikey-rsa-r-07 (work in progress), August 2006.

[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-03 (work in progress), December 2006.

[I-D.ietf-msec-mikey-dhhmac] Euchner, M., "HMAC-authenticated Diffie-Hellman for MIKEY", draft-ietf-msec-mikey-dhhmac-11 (work in progress), April 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[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-media-security-requirements] Wing, D., "Media Security Requirements", draft-wing-media-security-requirements-00 (work in progress), October 2006.

Author's Address

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

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

Full Copyright Statement

Intellectual Property

Acknowledgment

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