MSEC WG D. Ignjatic
Internet-Draft Polycom
Expires: August 4, 2006 L. Dondeti
QUALCOMM
F. Audet
P. Lin
Nortel
JAN 31, 2006

An additional mode of key distribution in MIKEY: MIKEY-RSA-R
draft-ietf-msec-mikey-rsa-r-02

Status of this Memo

Copyright Notice

Abstract

The Multimedia Internet Keying (MIKEY) specification describes several modes of key distribution to setup Secure Real-time Rransport Protocol (SRTP) sessions -- using pre-shared keys, public keys, and optionally a Diffie-Hellman key exchange. In the public-key mode,

the Initiator encrypts a random key with the Responder's public key and sends it to the Responder. In many communication scenarios, the Initiator may not know the Responder's public key, or in some cases the Responder's ID (e.g., call forwarding) in advance. We propose a new MIKEY mode that works well in such scenarios. This mode also enhances the group key management support in MIKEY; it supports member-initiated group key download (in contrast to group manager pushing the group keys to all members).

Table of Contents

1. Introduction

The MIKEY protocol [RFC3830] has three different methods for key transport or exchange: a pre-shared key mode (PSK), a public-key (RSA) mode, and an optional Diffie-Hellman exchange (DHE) mode. In addition, there is also an optional DH-HMAC mode [I-D.ietf-msec-mikey-dhhmac], bringing the total number of modes to four. The primary motivation for the MIKEY protocol design is low-latency requirements of real-time communication, and thus all the exchanges finish in one-half to 1 round-trip; note that this offers no room for security parameter negotiation of the key management protocol itself. In this document, we note that the MIKEY modes defined in RFC3830 [RFC3830] and [I-D.ietf-msec-mikey-dhhmac] are insufficient to address some deployment scenarious and common use cases, and propose a new mode called MIKEY-RSA in Reverse mode, or simply as MIKEY-RSA-R.

1.1. Terminology used in this document

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 BCP 14, RFC 2119 [RFC2119].

2. Motivation

As noted in the introduction, the MIKEY specification and other proposals define four different modes of efficient key management for real-time applications. Those modes differ from each other in either the authentication method of choice (public-key, or symmetric shared key-based), or the key establishment method of choice (key download, or key agreement using a Diffie-Hellman exchange). We summarize the modes, their advantages, and shortcomings in the following and also describe use cases where these modes are unusable or inefficient.

2.1. Description of the MIKEY modes

The PSK mode requires that the Initiator and the Responder have a common secret key established offline. In this mode, the Initiator selects a TEK Generation Key (TGK), encrypts it with a key derived from the PSK, and sends it to the Responder as part of the first message, namely, I_MESSAGE. The I_MESSAGE is replay protected with timestamps, and integrity protected with another key derived from the PSK. An optional Verification message from the Responder to the Initiator provides mutual authentication. This mode does not scale well as it requires pre-establishment of a shared key between communicating parties; for example, consider the use cases where any

user may want to communicate to any other user in an Enterprise or the Internet at large. The RSA mode might be more suitable for such applications.

In the RSA mode, the Initiator selects a TGK, encrypts and authenticates it with an envelope key, and sends it to the Responder as part of the I_MESSAGE. The Initiator includes the envelope key, encrypted with the Responder's public key, in I_MESSAGE. The I_MESSAGE is replay protected with timestamps, and signed with the Initiator's public key. The Initiator's ID, Certificate (CERT) and the Responder's ID that the Initiator intends to talk may be included in I_MESSAGE. If the Initiator knows several public-keys of the Responder, it can indicate the key used in the optional CHASH payload. An optional Verification message from the Responder to the Initiator provides mutual authentication. The RSA mode works well if the Initiator knows the Responder's ID and the corresponding CERT (or can obtain the CERT independent of the MIKEY protocol). RFC 3830 suggests that an Initiator, in the event that it does not have the Responder's CERT, may obtain the CERT from a directory agent using one or more round trips. However, in some cases, the Initiator may not even know the Responder's ID in advance, and because of that or for other reasons cannot obtain the Responder's CERT.

In addition to the case where the Responder may have several IDs, some applications may allow for the Responder's ID to change unilaterally, as is typical in telephony (e.g., forwarding). In those cases and in others, the Initiator might be willing to let the other party establish identity and prove it via an Initiator-trusted third party (e.g., a Certification Authority or CA).

The DH mode or the DH-HMAC mode of MIKEY might be useful in cases where the Initiator does not have access to the Responder's exact identity and/or CERT. In these modes, the two parties engage in an authenticated DH exchange to derive the TGK. On the downside, the DH modes have higher computational and communication overhead compared to the RSA and the PSK modes. More importantly, these modes are unsuitable for group key distribution.

In summary, in some communication scenarios -- where the Initiator might not have the correct ID and/or the CERT of the Responder -- none of the MIKEY modes described in [RFC3830] and [I-D.ietf-msec-mikey-dhhmac] are suitable/efficient for SRTP [RFC3711] key establishment.

2.2. Use case motivating the proposed mode

In addition to the issues listed above, there are some types of applications that motivate the new MIKEY mode design proposed in this

document.

Note that in the MIKEY-RSA mode (as in case of the PSK mode), the Initiator proposes the SRTP security policy, and chooses the TGK. However, it is also possible that the Initiator wants to allow the Responder to specify the security policy and send the TGK. Consider for example the case of a conferencing scenario where the convener sends an invitation to a group of people to attend a meeting. The procedure then might be for the invitees to request the convener for group key material by sending a MIKEY I_MESSAGE. Thus, in the MIKEY definition of initiators and responders, the Initiator is asking the Responder for keying material. Note that this mode of operation is inline with the MSEC group key management architecture [RFC4046].

3. A new MIKEY-RSA mode: MIKEY-RSA-R

3.1. Outline

The proposed MIKEY mode requires 1 full round trip. The Initiator sends a signed I_MESSAGE to the intended Responder requesting the Responder to send the SRTP keying material. The I_MESSAGE MAY

contain the Initiator's CERT or a link (URL) to the CERT, and similarly the Responder's reply, R_MESSAGE MAY contain the Responder's CERT or a link to it. The Responder can use the Initiator's public key from the CERT in the I_MESSAGE to send the encrypted TGK in the R_MESSAGE. Upon receiving the R_MESSAGE, the Initiator can use the CERT in the R_MESSAGE to verify whether the Responder is in fact the party that it wants to communicate to, and accept the TGK. We refer to this protocol as MIKEY-RSA in the reverse mode, or simply as MIKEY-RSA-R.

The MIKEY-RSA-R mode exchange is defined as follows:

```
Initiator                                                Responder
---------                                                ---------

I_MESSAGE = HDR, T, [RAND], [IDi|CERTi], [IDr], {SP}, SIGNi

R_MESSAGE = HDR, [GenExt(CSB-ID)], T, [RAND], [IDr|CERTr], [SP],
KEMAC, PKE, SIGNr
```

Figure 1: MIKEY-RSA-R mode

3.2. Components of the I_MESSAGE

MIKEY-RSA-R requires a full round trip to download the TGKs. The I_MESSAGE MUST have the MIKEY HDR and the timestamp payload for

replay protection. The HDR field contains a CSB_ID (Crypto Session Bundle ID) randomly selected by the Initiator. The V bit MUST be set to '1' and ignored by the Responder, as a response is MANDATORY in this mode. The Initiator MAY indicate the number of CSs supported, and SHOULD/MUST fill in the CS ID map type and CS ID info fields for the RTP/RTCP streams it originates (This is because the sender of the streams chooses the SSRC which is carried in the CS ID info field; see Section 6.1.1 of RFC 3830). The I_MESSAGE MUST be signed by the Initiator following the procedure to sign MIKEY messages specified in RFC 3830. The SIGNi payload contains this signature. Thus the I_MESSAGE is integrity and replay protected.

The RAND payload SHOULD be included in the I_MESSAGE when MIKEY-RSA-R mode is used for unicast communication. It MUST be omitted when this mode is used to establish group keys. The reason for the inclusion

of the RAND payload in unicast scenarios is to allow for the Initiator to contribute entropy to the key derivation process (in addition to the CSB_ID).

IDi and CERTi SHOULD be included, but they MAY be left out when it is expected that the peer already knows the Initiating party's ID (or can obtain the certificate in some other manner). For example, this could be the case if the ID is extracted from SIP. For certificate handling, authorization, and policies, see Section 4.3. of RFC 3830. If CERTi is included, it MUST correspond to the private key used to sign the I_MESSAGE.

If the Responder has multiple Identities, the Initiator MAY also include the specific ID, IDr, of the Responder that it wants to communicate with.

The Initiator MAY also send security policy (SP) payload(s) containing all the security policies that it supports. If the Responder does not support any of the policies included, it should reply with an Error message of type "Invalid SPpar" (Error no. 10).

SIGNi is a signature covering the Initiator's MIKEY message, I_MESSAGE, using the Initiator's signature key (see Section 5.2 of RFC 3830 for the exact definition). The I_MESSAGE is signed to protect against DoS attacks.

3.3. Components of the R_MESSAGE

Upon receiving an I_MESSAGE of the RSA-R format, the Responder MUST respond with one of the following messages:

- o The Responder SHOULD send an Error message "Message type not supported" (Error no. 13), if it cannot correctly parse the received MIKEY message. Error no. 13 is not defined in RFC 3830,

  and so RFC 3830 compliant implementations MAY return "an unspecified error occurred" (Error no. 12).
- o The Responder MUST send an Error message "Invalid SPpar" (Error no. 10), if it does not support any of the security policies included in the SP payload.
- o The Responder MUST send an R_MESSAGE, if SIGNi can be correctly verified and the timestamp is current; if an SP payload is present in the I_MESSAGE the Responder MUST return one of the proposed

security policies that matches the Responder's local policy.

The HDR payload in the R_MESSAGE is formed following the procedure described in RFC 3830. Specifically, the CSB ID in the HDR payload MUST be the same as the one in the HDR of the I_MESSAGE. The Responder MUST fill in the number of CSs and the CS ID map type and CS ID info fields of the HDR payload.

For group communication, all the members MUST use the same CSB ID and CS ID in computing the SRTP keying material. Therefore, for group key establishment, the Responder MUST include a General Extension Payload containing a new CSB ID in the R_MESSAGE. If a new CSB ID is present in the R_MESSAGE, the Initiator and the Responder MUST use that value in key material computation. Furthermore, the complete CS map SHOULD be populated by the Responder. The General Extension Payload carrying a CSB ID MUST NOT be present in case of one-to-one communication.

When used in group scenarios with bi-directional media, the Responder SHOULD include two TGKs or TEKs in the KEMAC payload. The first TGK or TEK SHOULD be used for receiving media on the Initiator's side and the second one to encrypt/authenticate media originating on the Initiator's side. This allows all the multicast traffic to be encrypted/authenticated by the same group key while keys used for unicast streams from all the group members can still be independent.

The T payload is exactly the same as that received in the I_MESSAGE.

If the I_MESSAGE did not include the RAND payload, it MUST be present in the R_MESSAGE. In case it has been included in the I_MESSAGE, it MUST NOT be present in the R_MESSAGE. In group communication, the RAND payload is always sent by the Responder and in unicast communication, either the Initiator or the Responder (but not both) generate and send the RAND payload.

IDr and CERTr SHOULD be included, but they MAY be left out when it can be expected that the peer already knows the other party's ID (or can obtain the certificate in some other manner). For example, this could be the case if the ID is extracted from SIP. For certificate handling, authorization, and policies, see Section 4.3. of RFC 3830.

If CERTr is included, it MUST correspond to the private key used to

sign the R_MESSAGE.

An SP payload MAY be included in the R_MESSAGE. If an SP payload was in the I_MESSAGE, then R_MESSAGE MUST contain an SP payload specifying the security policies of the secure RTP session being negotiated. More specifically, the Initiator may have provided multiple options, but the Responder must choose one option per Security Policy Parameter.

The KEMAC payload contains a set of encrypted sub-payloads and a MAC: KEMAC = E(encr_key, IDr || {TGK}) || MAC. The first payload (IDr) in KEMAC is the identity of the Responder (not a certificate, but generally the same ID as the one specified in the certificate). Each of the following payloads (TGK) includes a TGK randomly and independently chosen by the Responder (and possible other related parameters, e.g., the key lifetime). The encrypted part is then followed by a MAC, which is calculated over the KEMAC payload. The encr_key and the auth_key are derived from the envelope key, env_key, as specified in Section 4.1.4. of RFC 3830. The payload definitions are specified in Section 6.2 of RFC 3830.

The Responder encrypts and integrity protects the TGK with keys derived from a randomly/pseudo-randomly chosen envelope key, and encrypts the envelope key itself with the public key of the Initiator. The PKE payload contains the encrypted envelope key: PKE = E(PKi, env_key). It is encrypted using the Initiator's public key (PKi). Note that, as suggested in RFC 3830, the envelope key MAY be cached and used as the PSK for re-keying.

To compute the signature that goes in the SIGNr payload, the Responder MUST sign R_MESSAGE (excluding the SIGNr payload itself) || IDi || IDr || T. Note that the added identities and timestamp are identical to those transported in the ID and T payloads.

3.4. Additions to RFC 3830 message types and other values

3.4.1. Modified Table 6.1a from RFC 3830

Modified Table 6.1a from RFC 3830:

Data type	Value	Comment
Pre-shared	0	Initiator's pre-shared key message
PSK ver msg	1	Verification message of a Pre-shared key msg
Public key	2	Initiator's public-key transport message
PK ver msg	3	Verification message of a public-key message
D-H init	4	Initiator's DH exchange message
D-H resp	5	Responder's DH exchange message
Error	6	Error message
DHHMAC init	7	DH HMAC message 1
DHHMAC resp	8	DH HMAC message 2
RSA-R I_MSG	9	Initiator's public-key message in RSA-R mode
RSA-R R_MSG	10	Responder's public-key message in RSA-R mode

Figure 2: Table 6.1a from RFC 3830 (Revised)

3.4.2. Modified Table 6.12 from RFC 3830

Modified Table 6.12 from RFC 3830:

Error no	Value	Comment
Auth failure	0	Authentication failure
Invalid TS	1	Invalid timestamp
Invalid PRF	2	PRF function not supported
Invalid MAC	3	MAC algorithm not supported
Invalid EA	4	Encryption algorithm not supported
Invalid HA	5	Hash function not supported
Invalid DH	6	DH group not supported
Invalid ID	7	ID not supported
Invalid Cert	8	Certificate not supported
Invalid SP	9	SP type not supported
Invalid SPpar	10	SP parameters not supported
Invalid DT	11	not supported Data type
Unspecified error	12	an unspecified error occurred
Unsupported message type	13	unparseable MIKEY message

Figure 3: Table 6.12 from RFC 3830 (Revised)

3.4.3. Modified Table 6.15 from RFC 3830

Modified Table 6.15 from RFC 3830:

Type	Value	Comment
Vendor ID	0	Vendor specific byte string
SDP IDs	1	List of SDP key mgmt IDs (allocated for use in [KMASDP])
CSB-ID	2	Responder's modified CSB-ID (group mode)

Figure 4: Table 6.15 from RFC 3830 (Revised)

4. Applicability of the RSA-R and RSA modes

MIKEY-RSA-R mode and RSA mode are both very useful: deciding on which mode to use depends on the application.

The RSA-R mode is useful when you have reasons to believe that the Responder may be different from who you are sending the MIKEY message to. This is quite common in telephony and multimedia applications where the session/call can be retargeted/forwarded. When the security policy allows it, it may be appropriate for the Initiator to have some flexibility in who the Responder may turn out to be. In such cases, the main objective of the Initiator's RSA-R message is to present its public key/certificate to the Responder.

The second scenario is when the Initiator already has the Responder's certificate but wants to allow the Responder to come up with all the keying material. This is applicable in conferences where the Responder is the key distributor and the Initiators contact the Responder to initiate key download. Notice that this is quite similar to the group key download model as specified in GDOI [RFC3547], GSAKMP [I-D.ietf-msec-gsakmp-sec], and GKDP [I-D.ietf-msec-gkdp] protocols (also see [RFC4046]). The catch however is that the participating entities must know that they need to contact a well-known address as far as that conferencing group is concerned. Note that they only need the Responder's address, not necessarily its

CERT. If the group members have the Responder's CERT, there is no harm; they simply do not need the CERT to compose I_MESSAGE.

The RSA mode is useful when the Initiator knows the Responder's identity and CERT. This mode is also useful when the key exchange is happening in an established session with a Responder (for example, when switching from a non-secure mode to a secure mode), and when the policy is such that it is only appropriate to establish a MIKEY session with the Responder that is targeted by the Initiator.

4.1. Limitations

The RSA-R mode may not easily support 3-way calling, under the assumptions that motivated the design. An extra message may be required compared to the MIKEY-RSA mode specified in RFC 3830. Consider that A wants to talk to B and C, but does not have B's or C's CERT. A might contact B and request that B supply a key for a 3-way call. Now if B knows C's CERT, then B can simply use the MIKEY-RSA mode (as defined in RFC 3830) to send the TGK to C. If not, then the solution is not straightforward. For instance, A might ask C to contact B or itself to get the TGK, in effect initiating a 3-way exchange. It should be noted that 3-way calling is typically implemented using a bridge, in which case there are no issues (it looks like 3 point-to-point sessions, where one end of each session is a bridge mixing the traffic into a single stream).

5. Security Considerations

We offer a brief overview of the security properties of the exchange. There are two messages, viz., I_MESSAGE and R_MESSAGE. I_MESSAGE is a signed request by an Initiator requesting the Responder to select a TGK to be used to protect SRTP and SRTCP sessions.

The message is signed, which assures the Responder that the claimed Initiator has indeed generated the message. This automatically provides message integrity as well.

There is a timestamp in the I_MESSAGE, which when generated and interpreted in the context of the MIKEY specification, assures the Responder that the request is live and not a replay. Indirectly, this also provides protection against a DoS attack in that the

I_MESSAGE must itself be signed. The Responder however would have to verify the Initiator's signature and the timestamp, and thus would spend significant computing resources. It is possible to mitigate this by caching recently received and verified requests.

Note that the I_MESSAGE in this method basically equals DoS protection properties of the DH method and not the public key method as there are no payloads encrypted by the Responder's public key in I_MESSAGE. If IDr is not included in the I_MESSAGE, the Responder will accept the message and a response (and state) would be created for the malicious request.

The R_MESSAGE is quite similar to the I_MESSAGE in the MIKEY-RSA mode and has all the same security properties.

When using the RSA-R mode, the Responder may turn out to be different

from who the Initiator sent the MIKEY message to. It is the responsibility of the Initiator to verify that the identity of the Responder is acceptable (based on its local policy) if it changes from the intended Responder, and to take appropriate action based on the outcome. In some cases, it could be appropriate to accept Responder's identity if it can be strongly authenticated; in other cases, a blacklist or a whitelist may be appropriate.

When both unicast and multicast streams are negotiated it is suggested to use multiple instances of MIKEY rather than a single instance in group mode. Unicast and multicast keys have different security properties and should not be derived from the same pool. The authors believe that multiple TGK payloads can be used for this purpose but the exact method is not specified in MIKEY.

5.1. Impact of the Responder choosing the TGK

In the MIKEY-RSA or PSK modes, the Initiator chooses the TGK and the Responder has the option to accept the key or not. In the RSA-R mode for unicast communication, the Initiator and the Responder contribute random information in generating the TEK (RAND from the Initiator and the TGK from the Responder). For group communication, the sender will choose the TGK and the RAND; note that it is in the interest of the sender to provide sufficient entropy to TEK generation since the TEK protects data sent by the Responder.

Thus, in case of one-to-one communication, the RSA-R mode is slightly better than the RSA mode in that it allows the Initiator as well as the Responder to contribute entropy to the TEK generation process. This comes at the expense of the additional message. However, as noted earlier, the new mode needs the additional message to allow simpler provisioning.

RFC 3830 has additional notes on the security properties of the MIKEY protocol, key derivation functions and other components.

6. IANA Considerations

This specification requires the following IANA assignments to the MIKEY registry :

```
   Add to "Error Payload namespace:"
      Unsupported message type ------- 13
   Add to "Common Header payload name spaces:"
      RSA-R I_MSG ------------ 9
      RSA-R R_MSG ------------- 10
```



```
   General Extensions payload name spaces:
      CSB-ID ----------------- 2 (Note: another draft may use '2';
      please assign next available number)
```

7. Acknowledgments

8. References

8.1. 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.

8.2. Informative References

[RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

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

[I-D.ietf-msec-gsakmp-sec] Harney, H., "GSAKMP: Group Secure Association Group Management Protocol", draft-ietf-msec-gsakmp-sec-10 (work in progress), May 2005.

[I-D.ietf-msec-gkdp] Dondeti, L. and J. Xiang, "GKDP: Group Key Distribution Protocol", draft-ietf-msec-gkdp-00 (work in progress), September 2005.

Authors' Addresses

Dragan Ignjatic
Polycom
1000 W. 14th Street
North Vancouver, BC V7P 3P3
Canada

Phone: +1 604 982 3424
Email: dignjatic@polycom.com

Lakshminath Dondeti
QUALCOMM
5775 Morehouse drive
San Diego, CA 92121
US

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

Francois Audet
Nortel
4655 Great America Parkway
Santa Clara, CA 95054
US

Phone: +1 408 495 3756
Email: audet@nortel.com

Ping Lin
Nortel
250 Sidney St.
Belleville, Ontario K8P3Z3
Canada

Phone: +1 613 967 5343
Email: linping@nortel.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

Disclaimer of Validity

Copyright Statement

Acknowledgment

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