              MIKEY-RSA-R: An Additional Mode of Key Distribution
                   in Multimedia Internet KEYing (MIKEY)

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   The Multimedia Internet Keying (MIKEY) specification describes
   several modes of key distribution solution that address multimedia
   scenarios (e.g., SIP calls and Real Time Streaming Protocol (RTSP)
   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). This document updates RFC 3830 with the RSA-R mode.

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 [RFC4650], 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
   roundtrip; 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] and [RFC4650] are
   insufficient to address some deployment scenarios and common use
   cases, and we propose a new mode called MIKEY-RSA in Reverse mode, or
   simply MIKEY-RSA-R. This document updates RFC 3830 with the addition
   of this new mode to that specification.

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].

   Furthermore, this document reuses the terminology of the MIKEY
   specification [RFC3830].

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
   these modes below, including their advantages and shortcomings. We
   then discuss the 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 the 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 may be included in the 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
roundtrips. 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 (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. The DH-HMAC mode also
requires establishment of PSKs between all possible communicating
entities and thus has similar scaling issues as any PSK-based key
management protocol.

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] or [RFC4650] are
suitable and efficient for multimedia session 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 session 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 group key
   material from the convener 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 in line 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 roundtrip. The Initiator
   sends a signed I_MESSAGE to the intended Responder requesting the
   Responder to send the traffic 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 Unicast Mode

3.2. Group Communication Using the MIKEY RSA-R Mode

   For group conferencing using MIKEY RSA-R mode, the members receive an
   invitation to initiate MIKEY with the group key server to download
   the secure session information. In this case, the Responder is
   either the group sender or group key server. Group members request
   group policy and keying material as MIKEY RSA-R Initiators.
   Initiators MUST NOT send the SP payload. The Responder sends all the
   payloads necessary to distribute the secure group policy as well as
   payloads used in the group key derivation: specifically, the SP
   payload is used to convey the session policy, and the GenExt(CSB-ID),
   TGK, and the RAND payloads selected by the Responder and included in
   the R_Message are used to compute the Secure Realtime Transport
   Protocol (SRTP) session keys.

   MIKEY RSA-R for group communication:

   Initiator   Responder
   ---------   ---------

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

   R_MESSAGE = HDR, GenExt(CSB_ID), T, RAND, [IDr|CERTr], SP,
               KEMAC, PKE, SIGNr

                    Figure 2: MIKEY-RSA-R in Group Mode

   Note that the SP payload in the I_MESSAGE is not present. In the
   R_MESSAGE, the CSB_ID, RAND, and SP payloads are not optional.

3.3. Preparing RSA-R Messages

   Preparation and parsing of RSA-R messages are as described in
   Sections 5.2 and 5.3 of RFC 3830. Error handling is described in
   Section 5.1.2 and replay protection guidelines are in Section 5.4 of
   RFC 3830. In the following, we describe the components of RSA-R
   messages and specify message processing and parsing rules in addition
   to those in RFC 3830.

3.4. Components of the I_MESSAGE

   MIKEY-RSA-R requires a full roundtrip 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 SHOULD indicate the number of CSs
   supported, and SHOULD 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 that is carried in the CS ID
info field; see Section 6.1.1 of RFC 3830. The exception to
Initiators not specifying SSRC values is to allow the Responder to
pick them to avoid SSRC collisions. Initiators of MIKEY messages
that do not originate RTP streams MUST specify a '0' as the number of
CSs supported. This typically applies to group communication and to
the entities in the listen-only mode.

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. The reason for recommending
the inclusion of the RAND payload in the I_MESSAGE for unicast
communication is to allow the Initiator to contribute entropy to the
key derivation process (in addition to the CSB_ID). When the RAND
payload is not included, the Initiator will be relying on the
Responder to supply all the entropy for SRTP key generation, which is
in fact similar (but with the reversal of roles) to the MIKEY-RSA
mode, where the Responder supplies all the entropy.

The RAND payload MAY be included when MIKEY-RSA-R is used to
establish group keys. However, the RAND payload in the I_MESSAGE
MUST NOT be used for MIKEY key generation, in case of group
communication. The Responder MUST include a RAND payload in the
R_MESSAGE for TEK generation from a TGK when MIKEY-RSA-R is used for
group communication.

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 Sections 4.3 and 6.7 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 identity, IDr, of the Responder with whom
communication is desired. If the Initiator's policy does not allow
acceptance of an R_MESSAGE from any entity other than one that can
assert a specific identity, the Initiator MUST include that specific
identity in an IDr payload in the I_MESSAGE.

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).
The Responder has the option not to send the Error message in MIKEY
if a generic session establishment failure indication is deemed
appropriate and communicated via other means (see Section 4.1.2 of
[RFC4567] for additional guidance).

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 signature assures the
Responder that the claimed Initiator has indeed generated the
message. This automatically provides message integrity as well.

3.5. Processing the I_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 message format is as specified in
      Section 5.1.2 of RFC 3830. 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 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.

   o If a RAND payload is present in the I_MESSAGE, both sides use that
      RAND payload as the RAND value in the MIKEY key computation. In
      case of multicast, if a RAND payload is present in the I_MESSAGE,
      the Responder SHOULD ignore the payload. In any case, the
      R_MESSAGE for multicast communication MUST contain a RAND payload
      and that RAND payload is used for key computation.

   o The rest of the error message rules are as described in Section
      5.1.2 of RFC 3830, and message processing rules are as described
      in Section 5.3 of RFC 3830.

3.6. Components of the R_MESSAGE

   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 traffic 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 CS ID map
   type and CS ID map info MUST be populated by the Responder. The
   General Extension Payload carrying a CSB_ID MUST NOT be present in
   case of unicast communication.

   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
   Responder always sends the RAND payload 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 the 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 or 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,
env_key: PKE = E(PKi, env_key). PKi denotes the Initiator's public
key. 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.

3.7. Processing the R_MESSAGE

   In addition to the processing rules in RFC 3830, the following rules
   apply to processing of the R_MESSAGE of MIKEY RSA-R mode.

      If the I_MESSAGE contained a RAND payload, the Initiator MUST
      silently discard an R_MESSAGE that contains a RAND payload.
      Similarly, if the I_MESSAGE did not contain a RAND payload, the
      Initiator MUST silently discard an R_MESSAGE that does not contain
      a RAND payload.

      If the SP payload contains a policy not specified in the SP
      message, if present, in the I_MESSAGE, such an R_MESSAGE MUST be
      discarded silently.

3.8. Certificate Handling

   If a Certificate payload is present, the X.509v3 URL Cert type from
   Table 6.7.b [RFC3830] is the default method in RSA-R mode and MUST be
   implemented. The HTTP URL to fetch a certificate as specified in RFC
   2585 [RFC2585] MUST be supported. Devices are not required to
   support the FTP URLs. When retrieving data from the URL,
   application/pkix-cert MIME type with X.509 certificates DER-encoded
   MUST be supported.

The RECOMMENDED way of doing certificate validation is by using OCSP
as specified by RFC 2560 [RFC2560]. When OCSP is used and nextUpdate
time is present in the response, it defines how long the certificate
can be considered valid and cached. If OCSP is not supported or
nextUpdate time is not present in the response, the certificate cache
timeout is a matter of local policy.

The communicating peers (such as SIP User Agents for instance) MAY
choose to create a URL pointing to certificate files residing on
themselves or by appending their ID and a ".cer" extension to a
provisioned root path to the certificate. Other methods MAY also be
used, subject to local policy.

3.9. Additions to RFC 3830 Message Types and Other Values

   This document introduces two new message types (Table 6.1a of RFC
   3830), an Error no (Table 6.12 of RFC 3830), and a general extension
   payload (Table 6.15 of RFC 3830). This section specifies those
   additions.

3.9.1. 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 RSA-R public-key message (NEW)
RSA-R R_MSG	10	Responder's RSA-R public-key message (NEW)

                 Figure 3: Table 6.1a from RFC 3830 (Revised)

3.9.2. 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 (NEW)
```

              Figure 4: Table 6.12 from RFC 3830 (Revised)

3.9.3. Modified Table 6.15 from RFC 3830

   Modified Table 6.15 from RFC 3830:

```
      Type         | Value | Comments
      ---------------------------------------
      Vendor ID    |     0 | Vendor specific byte string
      SDP IDs      |     1 | List of SDP key mgmt IDs (allocated for use in
                   |       |   [RFC4567])
      TESLA I-Key  |     2 |   [RFC4442]
      Key ID       |     3 | information on type and identity of keys
                   |       |   [RFC4563])
      CSB_ID       |     4 | Responder's modified CSB_ID (group mode)
```

               Figure 5: 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 a different party than the one to which the MIKEY
    I_MESSAGE was sent. This is quite common in telephony and multimedia
    applications where the session or the call can be retargeted or
    forwarded. When the security policy allows it, leaving some
    flexibility for the Initiator to see who the Responder may turn out
    to be, before making the decision to continue or discontinue the
    session, may be appropriate. In such cases, the main objective of
    the Initiator's RSA-R message is to present its public key/
    certificate to the Responder, and wait for a Responder to present its
    identity.

    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 [RFC4535], and GKDP [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 the 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: the I_MESSAGE and the R_MESSAGE. The
   I_MESSAGE is a signed request by an Initiator requesting the
   Responder to select a TGK to be used to protect multimedia (e.g.,
   Secure RTP or SRTP [RFC3711]) 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 denial of service (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
   the 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 be a different party
   than the one to which the MIKEY I_MESSAGE was sent. 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 party to which the MIKEY I_MESSAGE was sent, and to take
   appropriate action based on the outcome. In some cases, it could be
   appropriate to accept a 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 need to be negotiated, it is
RECOMMENDED to use multiple instances of MIKEY-RSA-R rather than a
single instance in group mode. This is to avoid potential key reuse
with counter mode.

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 RECOMMENDED mode of operation is for
the Initiator and the Responder to contribute random information in
generating the TEK (RAND from the Initiator and the TGK from the
Responder). For group communication, the sender (MIKEY Responder)
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 unicast 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.

5.2. Updates to Security Considerations in RFC 3830

MIKEY requires clock synchronization, and a secure network clock
synchronization protocol SHOULD be used, e.g., [ISO3] or secure NTP
[NTPv4].

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

6. IANA Considerations

The following IANA assignments were added to the MIKEY registry:

   Added to "Error payload name spaces:"

      Unsupported message type ------- 13

   Added to "Common Header payload name spaces:"

      RSA-R I_MSG ------------- 9
      RSA-R R_MSG ------------- 10

      Added to "General Extensions payload name spaces:"

         CSB_ID ----------------- 4

7. Acknowledgments

   Many thanks to Mark Baugher, Steffen Fries, Russ Housley, Cullen
   Jennings, and Vesa Lehtovirta for their reviews of earlier version of
   this document.

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.

   [RFC2560]   Myers, M., Ankney, R., Malpani, A., Galperin, S., and C.
               Adams, "X.509 Internet Public Key Infrastructure Online
               Certificate Status Protocol - OCSP", RFC 2560, June 1999.

   [RFC2585]   Housley, R. and P. Hoffman, "Internet X.509 Public Key
               Infrastructure Operational Protocols: FTP and HTTP",
               RFC 2585, May 1999.

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

8.2. Informative References

   [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.

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

   [RFC4650]   Euchner, M., "HMAC-Authenticated Diffie-Hellman for
               Multimedia Internet KEYing (MIKEY)", RFC 4650, September
               2006.

   [RFC4535]   Harney, H., Meth, U., Colegrove, A., and G. Gross,
               "GSAKMP: Group Secure Association Key Management
               Protocol", RFC 4535, June 2006.

   [GKDP]   Dondeti, L., "GKDP: Group Key Distribution Protocol", Work
               in Progress, March 2006.

   [RFC4567]   Arkko, J., Lindholm, F., Naslund, M., Norrman, K., and E.
               Carrara, "Key Management Extensions for Session
               Description Protocol (SDP) and Real Time Streaming
               Protocol (RTSP)", RFC 4567, July 2006.

   [RFC4442]   Fries, S. and H. Tschofenig, "Bootstrapping Timed
               Efficient Stream Loss-Tolerant Authentication (TESLA)",
               RFC 4442, March 2006.

   [RFC4563]   Carrara, E., Lehtovirta, V., and K. Norrman, "The Key ID
               Information Type for the General Extension Payload in
               Multimedia Internet KEYing (MIKEY)", RFC 4563, June 2006.

   [NTPv4]   Burbank, J., "The Network Time Protocol Version 4 Protocol
               Specification", Work in Progress, May 2006.

   [ISO3]   ISO, "ISO/IEC 18014 Information technology - Security
               techniques - Time-stamping services, Part 1-3", 2002.

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

Full Copyright Statement

Intellectual Property

Acknowledgement