          On the Applicability of Various Multimedia Internet KEYing (MIKEY)
                           Modes and Extensions

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Abstract

   Multimedia Internet Keying (MIKEY) is a key management protocol that
   can be used for real-time applications. In particular, it has been
   defined focusing on the support of the Secure Real-time Transport
   Protocol (SRTP). MIKEY itself is standardized within RFC 3830 and
   defines four key distribution methods. Moreover, it is defined to
   allow extensions of the protocol. As MIKEY becomes more and more
   accepted, extensions to the base protocol arise, especially in terms
   of additional key distribution methods but also in terms of payload
   enhancements.

   This document provides an overview about the MIKEY base document in
   general as well as the existing extensions for MIKEY, which have been
   defined or are in the process of definition. It is intended as an
   additional source of information for developers or architects to
   provide more insight in use case scenarios and motivations as well as
   advantages and disadvantages for the different key distribution
   schemes. The use cases discussed in this document are strongly
   related to dedicated SIP call scenarios providing challenges for key
   management in general, among them media before Session Description
   Protocol (SDP) answer, forking, and shared key conferencing.

Table of Contents

1. Introduction

   Key distribution describes the process of delivering cryptographic
   keys to the required parties. MIKEY [RFC3830], the Multimedia
   Internet Keying, has been defined focusing on support for the
   establishment of security context for the Secure Real-time Transport
   Protocol [RFC3711]. Note that RFC 3830 is not restricted to be used
   for SRTP only, as it features a generic approach and allows for
   extensions to the key distribution schemes. Thus, it may also be
   used for security parameter negotiation for other protocols.

   For MIKEY, meanwhile, seven key distribution methods are described:

   o Symmetric key distribution as defined in [RFC3830] (MIKEY-PSK)

   o Asymmetric key distribution as defined in [RFC3830] (MIKEY-RSA)

   o Diffie-Hellman key agreement protected by digital signatures as
      defined in [RFC3830] (MIKEY-DHSIGN)

   o Unprotected key distribution (MIKEY-NULL)

   o Diffie-Hellman key agreement protected by symmetric pre-shared
      keys as defined in [RFC4650] (MIKEY-DHHMAC)

   o Security Assertion Markup Language (SAML) assisted Diffie-Hellman
      key agreement as defined (not available as a separate document,
      but discussions are reflected within this document (MIKEY-DHSAML))

   o Asymmetric key distribution (based on asymmetric encryption) with
      in-band certificate provision as defined in [RFC4738]
      (MIKEY-RSA-R)

   Note that the latter three modes are extensions to MIKEY as there
   have been scenarios where none of the first four modes defined in
   [RFC3830] fits perfectly. There are further extensions to MIKEY
   comprising algorithm enhancements and a new payload definition
   supporting protocols other than SRTP.

   Algorithm extensions are defined in the following document:

   o Elliptic Curve Cryptography (ECC) algorithms for MIKEY as defined
      in [MSEC-MIKEY]

Payload extensions are defined in the following documents:

o Bootstrapping TESLA, defining a new payload for the Timed
   Efficient Stream Loss-tolerant Authentication (TESLA) protocol
   [RFC4082] as defined in [RFC4442]

o The Key ID information type for the general extension payload as
   defined in [RFC4563]

o Open Mobile Alliance (OMA) Broadcast (BCAST) MIKEY General
   Extension Payload Specification as defined in [RFC4909]

o Integrity Transform Carrying Roll-over Counter for SRTP as defined
   in [RFC4771]. Note that this is rather an extension to SRTP and
   requires MIKEY to carry a new parameter, but is stated here for
   completeness.

This document provides an overview about RFC 3830 and the relations
to the different extensions to provide a framework when using MIKEY.
It is intended as an additional source of information for developers
or architects to provide more insight in use case scenarios and
motivations as well as advantages and disadvantages for the different
key distribution schemes. The use cases discussed in this document
are inspired by specific protocol workings of SIP that have proved to
be problematic for a general key distribution mechanisms in general.
These protocol workings are described in detail in Wing, et al.
[SIP-MEDIA] and include the following:

o Early Media (i.e., media that arrives before the SDP answer)

o Forking

o Call Transfer/Redirect/Retarget

o Shared Key Conferencing

2. Terminology and Definitions

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].

The following definitions have been taken from [RFC3830]:

(Data) Security Protocol: the security protocol used to protect the
                           actual data traffic. Examples of security
                           protocols are IPsec and SRTP.

   Data SA   Data Security Association information for the security
                   protocol, including a TEK and a set of parameters/
                   policies.

   CS   Crypto Session, uni- or bidirectional data stream(s),
                   protected by a single instance of a security protocol.

   CSB   Crypto Session Bundle, collection of one or more
                   Crypto Sessions, which can have common TGKs (see
                   below) and security parameters.

   CS ID   Crypto Session ID, unique identifier for the CS within
                   a CSB.

   CSB ID   Crypto Session Bundle ID, unique identifier for the
                   CSB.

   TGK   TEK Generation Key, a bit-string agreed upon by two or
                   more parties, associated with CSB. From the TGK,
                   Traffic-Encrypting Keys can then be generated without
                   needing further communication.

   TEK   Traffic-Encrypting Key, the key used by the security
                   protocol to protect the CS (this key may be used
                   directly by the security protocol or may be used to
                   derive further keys depending on the security
                   protocol). The TEKs are derived from the CSB's TGK.

   TGK re-keying   the process of re-negotiating/updating the TGK (and
                   consequently future TEK(s)).

   Initiator   the initiator of the key management protocol, not
                   necessarily the initiator of the communication.

   Responder   the responder in the key management protocol.

   Salting key   a random or pseudo-random (see [RFC4086]) string used
                   to protect against some off-line pre-computation
                   attacks on the underlying security protocol.

   HDR   the protocol header

   PRF(k,x)   a keyed pseudo-random function

   E(k,m)   encryption of m with the key k

   RAND   random value

```
   T              timestamp

   CERTx          the certificate of x

   SIGNx          the signature from x using the private key of x

   PKx            the public key of x

   IDx            the identity of x

   []             an optional piece of information

   {}             zero or more occurrences

   ||             concatenation

   |              OR (selection operator)

   ^              exponentiation

   XOR            exclusive or
```

   The following definitions have been added to the ones from [RFC3830]:

```
   SSRC           Synchronization Source Identifier

   KEMAC          MIKEY Key Data Transport Payload, containing a set of
                  encrypted sub-payloads and a Message Authentication
                  Code (MAC).

   V              MIKEY Verification Message

   SP             Security Parameter

   Forking        The ability of a SIP proxy to replicate an incoming
                  request to multiple outgoing requests in order to
                  efficiently find the called party for rendezvous.  SIP
                  forking can be done in serial (depth-first search) or
                  in parallel (breadth-first search).

   Redirect       The ability of a SIP proxy to send a final response
                  that redirects the caller to send a request to an
                  alternate location.

   Retarget       The ability of a SIP proxy to re-write the Request-URI
                  thereby altering the destination of the request
                  without explicitly notifying the user agent client.
```

3. MIKEY Overview

   This section will provide an overview about MIKEY. MIKEY focuses on
   the setup of cryptographic context to secure multimedia sessions in a
   heterogeneous environment. MIKEY is mainly intended to be used for
   peer-to-peer, simple one-to-many, and small-size (interactive)
   groups. One objective of MIKEY is to produce a data security
   association (SA) for the security protocol, including a Traffic-
   Encrypting Key (TEK), which is derived from a TEK Generation Key
   (TGK), and used as input for the security protocol.

   MIKEY supports the possibility of establishing keys and parameters
   for more than one security protocol (or for several instances of the
   same security protocol) at the same time. The concept of Crypto
   Session Bundle (CSB) is used to denote a collection of one or more
   Crypto Sessions that can have common TGK and security parameters, but
   that obtain distinct TEKs from MIKEY.

   MIKEY as defined in RFC 3830 may proceed with one roundtrip at most,
   using a so-called Initiator message for the forward direction and a
   Responder message for the backward direction. Note that there exist
   MIKEY schemes that may proceed within a half roundtrip (e.g., based
   on a pre-shared key), while other schemes require a full roundtrip
   (e.g., Diffie-Hellman-based schemes). The main objective of the
   Initiator's message (I_MESSAGE) is to transport one or more TGKs
   (carried in the KEMAC field) and a set of security parameters (SPs)
   to the Responder in a secure manner. As the verification message
   from the Responder is optional for some schemes, the Initiator
   indicates whether or not it requires a verification message from the
   Responder.

   The focus of the following subsections lies on the key distribution
   methods as well as the discussion about advantages and disadvantages
   of the different schemes. Note that the MIKEY key distribution
   schemes rely on loosely synchronized clocks. If clock
   synchronization is not available, the replay handling of MIKEY (cf.
   [RFC3830]) may not work. This is due to the fact that MIKEY does not
   use a challenge-response mechanism for replay handling; instead,
   timestamps are used together with message caching. Thus, the
   required synchronization depends on the number of messages that can
   be cached on either side. Therefore, MIKEY recommends adjusting the
   cache size depending on the clock skew in the deployment environment.
   Moreover, RFC 3830 recommends the ISO time synchronization protocol
   [ISO_sec_time]. If replay handling is not available, an attacker may
   be able to replay an older message that he eavesdropped earlier,
   leading to different TGKs on both sides. As these are fed to the
   application utilizing MIKEY (e.g., SRTP or TESLA), both sides may
   rely on different keys and thus may be unable to communicate with

each other. The format applied to the timestamps submitted in MIKEY
have to match the NTP format described in [RFC1305]. In other cases,
such as of a SIP endpoint, clock synchronization by deriving time
from a trusted outbound proxy may be appropriate .

The different MIKEY-related schemes are compared regarding the
following criteria:

o Mandatory for implementation: provides information, if RFC 3830
   requires the implementation of this scheme.

o Scalability: describes the technical feasibility to easily deploy
   a solution based on the considered scheme.

o Dependency on PKI: states if the support of a PKI is required to
   support this scheme. Note that PKI here relates to PKI services
   like key generation, distribution, and revocation.

o Provision of Perfect Forward Secrecy (PFS): describes the support
   of PFS, which is, according to RFC 4949 [RFC4949], the property
   that compromising the long-term keying material does not
   compromise session keys that were previously derived from the
   long-term material.

o Key generation involvement: describes if both or just one of the
   participants is actively involved in key generation. The option
   to involve both parties in the key generation is considered here
   as it addresses several points:

   * If both sides contribute public entropy, it is ensured that
      each side can guarantee that keys are fresh to avoid replay
      attacks.

   * Involvement of both sides avoids that one side generates
      (intentionally or unintentionally) weak (predictable) nonces,
      which in turn may result in weak keys.

o Support of group keying: feasibility of the MIKEY option to be
   used also for group keying, e.g., in conferencing scenarios.

If MIKEY is used for SRTP [RFC3711] bootstrapping, it also uses the
SSRC to associate security policies with actual sessions. The SSRC
identifies the synchronization source. The value is chosen randomly,
with the intent that no two synchronization sources within the same
SRTP session will have the same SSRC. Although the probability of
multiple sources choosing the same identifier is low, all (S)RTP
implementations must be prepared to detect and resolve collisions.
Nevertheless, in multimedia communication scenarios supporting

forking (see Section 5.2) or retargeting (see Section 5.3) collisions
may occur leading to so-called two-time pads; i.e., the same key is
used for media streams to different destinations. This occurs if two
branches have the same TEK (based on the MIKEY key establishment) and
choose the same 32-bit SSRC for the SRTP streams. The SRTP key
derivation will then produce the same session keys (as the input
values are the same) and also derive the same initialization vector
per packet, as the SSRCs are the same. Note that two time pads may
also occur for media streams to the same destination. This is
outlined in [RFC3711].

3.1. Pre-Shared Key (PSK) Protected Distribution

   This option of the key management uses a pre-shared secret key to
   derive key material for integrity protection and encryption to
   protect the actual exchange of key material. Note that the pre-
   shared secret is agreed upon before the session, e.g., by out-of-band
   means. The responder message is optional and may be used for mutual
   authentication (proof of possession of the pre-shared secret) or
   error signaling.

   Initiator   Responder

   I_MESSAGE =
   HDR, T, RAND, [IDi],[IDr],
       {SP}, KEMAC   --->
                                               R_MESSAGE =
                                   [<---]   HDR, T, [IDr], V

   The advantages of this approach lay in the fact that there is no
   dependency on a PKI (Public Key Infrastructure), the solution
   consumes low bandwidth and enables high performance, and is all in
   all a simple straightforward master key provisioning. The
   disadvantages are that perfect forward secrecy is not provided and
   key generation is just performed by the Initiator. Furthermore, the
   approach is not scalable to larger configurations but is acceptable
   in small-sized groups. Note that according to [RFC3830], this option
   is mandatory to implement.

3.2. Public Key Encrypted Key Distribution

   Using the asymmetric option of the key management, the Initiator
   generates the key material (TGKs) to be transmitted and sends it
   encrypted with a so-called envelope key, which in turn is encrypted
   with the receiver's public key. The envelope key, env-key, which is
   a random number, is used to derive the auth-key and the enc-key.
   Moreover, the envelope key may be used as a pre-shared key to

establish further crypto sessions. The responder message is optional
and may be used for mutual authentication or error signaling.

```
   Initiator                                 Responder

   I_MESSAGE =
   HDR, T, RAND, [IDi|CERTi],
      [IDr], {SP}, KEMAC, [CHASH],
      PKE, SIGNi                    --->
                                             R_MESSAGE =
                                  [<---]      HDR, T, [IDr], V
```

An advantage of this approach is that it allows the usage of self-
signed certificates, which in turn can avoid a full-blown PKI. Note
that using self-signed certificates may result in limited scalability
and also require additional means for authentication such as exchange
of fingerprints of the certificates or similar techniques. The
disadvantages comprise the necessity of a PKI for full scalability,
the performance of the key generation just by the Initiator, and no
provision of perfect forward secrecy. Additionally, the Responder
certificate needs to be available in advance at the sender's side.
Furthermore, the verification of certificates may not be done in real
time. This could be the case in scenarios where the revocation
status of certificates is checked through a further component.
Depending on the Initiator role, this scheme can also be applied in
group-based communication, where a central server distributes the
group key protected with the public keys of the associated clients.
Note that according to [RFC3830], this option is mandatory to
implement.

3.3. Diffie-Hellman Key Agreement Protected with Digital Signatures

The Diffie-Hellman option of the key management enables a shared
secret establishment between the Initiator and Responder in a way
where both parties contribute to the shared secret. The Diffie-
Hellman key agreement is authenticated (and integrity protected)
using digital signatures.

```
   Initiator                                 Responder

   I_MESSAGE =
   HDR, T, RAND, [IDi|CERTi],
        [IDr], {SP}, DHi, SIGNi    --->
                                             R_MESSAGE =
                                  <---         HDR, T, [IDr|CERTr],
                                               IDi, DHr, DHi, SIGNr
```

[RFC3830] does mandate the support of RSA as a specific asymmetric
algorithm for the signature generation. Additionally, the algorithm
used for signature or public key encryption is defined by, and
dependent on, the certificate used. Besides the use of X.509v3
certificates, it is mandatory to support the Diffie-Hellman group
"OAKLEY5" [RFC2412]. It is also possible to use other Diffie-Hellman
groups within MIKEY. This can be done by defining a new mapping sub-
payload and the associated policy payload according to [RFC3830].
The advantages of this approach are a fair, mutual key agreement
(both parties provide to the key), perfect forward secrecy, and the
absence of the need to fetch a certificate in advance as needed for
the MIKEY-RSA method depicted above. Moreover, it also provides the
option to use self-signed certificates to avoid a PKI deployment.
Note that, depending on the security policy, self-signed certificates
may not be suitable for every use case.

Negatively to remark is that this approach scales mainly to point-to-
point and depends on PKI for full scalability. Multiparty
conferencing is not supported using just MIKEY-DHSIGN. Nevertheless,
the established Diffie-Hellman-Secret may serve as a pre-shared key
to bootstrap group-related security parameter. Furthermore, as for
the MIKEY-RSA mode described above, the verification of certificates
may not necessarily be done in real time. This could be the case in
scenarios where the revocation status of certificates is checked
through a further component. Note that, according to [RFC3830], it
is optional to implement this scheme.

3.4. Unprotected Key Distribution

RFC 3830 also supports a mode to provide a key in an unprotected
manner (MIKEY-NULL). This is based on the symmetric key encryption
option depicted in Section 3.1 but is used with the NULL encryption
and the NULL authentication algorithms. It may be compared with the
plain approach in SDP security descriptions [RFC4568]. MIKEY-NULL
completely relies on the security of the underlying layer, e.g.,
provided by TLS. This option should be used with caution as it does
not protect the key management.

Based on the missing cryptographic protection of this method, it is
obvious that perfect forward secrecy is not provided. As it is based
on the pre-shared secret mode, only the Initiator contributes to the
key management. The method itself is highly scalable, but again,
without proper protection through an underlying security layer, it is
not advisable for use.

3.5. Diffie-Hellman Key Agreement Protected with Pre-Shared Secrets

   This is an additional option, which has been defined in [RFC4650].
   In contrast to the method described in Section 3.3, here the Diffie-
   Hellman key agreement is authenticated (and integrity protected)
   using a pre-shared secret and keyed hash function.

   Initiator   Responder

   I_MESSAGE =
       HDR, T, RAND, [IDi],
       IDr, {SP}, DHi, KEMAC   --->
                                                 R_MESSAGE =
                                     <---   HDR, T,[IDr], IDi,
                                                     DHr, DHi, KEMAC

   TGK = g^(xi * yi)   TGK = g^(xi * yi)

   For the integrity protection of the Diffie-Hellman key agreement,
   [RFC4650] mandates the use of HMAC SHA-1. Regarding Diffie-Hellman
   groups, [RFC3830] is referenced. Thus, it is mandatory to support
   the Diffie-Hellman group "OAKLEY5" [RFC2412]. It is also possible to
   use other Diffie-Hellman groups within MIKEY. This can be done by
   defining a new mapping sub-payload and the associated policy payload
   according to RFC 3830. This option has also several advantages, as
   there are the fair mutual key agreement, the perfect forward secrecy,
   and no dependency on a PKI and PKI standards. Moreover, this scheme
   has a sound performance and reduced bandwidth requirements compared
   to MIKEY-DH-SIGN and provides a simple and straightforward master key
   provisioning. The establishment of shared secrets and the lack of
   support for group keying is a disadvantage.

   This mode of operation provides an efficient scheme in deployments
   where there is a central trusted server that is provisioned with
   shared secrets for many clients. Such setups could, for example, be
   enterprise Private Branch Exchanges (PBXs), service provider proxies,
   etc. In contrast to the plain pre-shared key encryption-based mode,
   described in Section 3.1, this mode offers perfect forward secrecy as
   well as active involvement in the key generation of both parties
   involved.

3.6. SAML-Assisted DH key Agreement

   There has been a longer discussion during IETF meetings and also on
   the IETF MSEC mailing list about a SAML-assisted DH approach. This
   idea has not been submitted as a separate document. Nevertheless,
   the discussion is reflected here as it is targeted to fulfill general

requirements on key management approaches. Those requirements can be
summarized as:

1. Mutual authentication of involved parties

2. Both parties involved contribute to the session key generation

3. Provide perfect forward secrecy

4. Support distribution of group session keys

5. Provide liveliness tests when involved parties do not have a
    reliable clock

6. Support of limited parties involved

To fulfill all of the requirements, it was proposed to use a classic
Diffie-Hellman key agreement protocol for key establishment in
conjunction with a User Agent's (UA's) SIP server signed element,
authenticating the Diffie-Hellman key and the ID using the SAML
(Security Assertion Markup Language [SAML_overview]) approach. Here
the client's public Diffie-Hellman credentials are signed by the
server to form a SAML assertion (referred to as CRED below), which
may be used for later sessions with other clients. This assertion
needs at least to convey the ID, public DH key, expiry, and the
signature from the server. It provides the involved clients with
mutual authentication and message integrity of the key management
messages exchanged.

```
Initiator                                  Responder

I_MESSAGE =
HDR, T, RAND1, [CREDi],
IDr, {SP}                      --->
                                           R_MESSAGE =
                                <---   HDR, T, [CREDr], IDi, DHr,
                                           RAND2, (SP)
          TGK = HMACx(RAND1|RAND2), where x = g^(xi * xr).
```

Additionally, the scheme proposes a second roundtrip to avoid the
dependence on synchronized clocks and provide liveliness checks.
This is achieved by exchanging nonces, protected with the session
key. The second roundtrip can also be used for distribution of group
keys or to leverage a weak DH key for a stronger session key. The
trigger for the second roundtrip would be handled via SP, the
security policy communicated via MIKEY.

```
   Initiator                              Responder

   I_MESSAGE =
   HDR, SIGN(ENC(RAND3))            --->
                                           R_MESSAGE =
                                   <---    SIGN(ENC(RAND4))
```

Note that if group keys are to be provided, RAND would be substituted by that group key.

With the second roundtrip, this approach also provides an option for all of the other key distribution methods, when liveliness checks are needed. The drawback of the second roundtrip is that these messages need to be integrated into the call flow of the signaling protocol. In a straight-forward call, one roundtrip may be enough to set up a session. Thus, this second roundtrip would require additional messages to be exchanged.

Regarding the different criteria discussed in the introduction of this section, the advantages of this approach are a fair, mutual key agreement (both parties provide to the key), and perfect forward secrecy. Through the second roundtrip, the dependency on synchronized clocks can be avoided. Moreover, this second roundtrip enables the distribution of a group key and thus enhances the scalability from mainly point-to-point to also multiparty conferencing. The usage of SAML-assisted DH may decrease the hidden latency cost through the credential validation necessary to be done for the signed DH scheme described in Section 3.3. If the UA received its SAML assertion from its domain's SIP server, it is trusting the server implicitly, thus, it may extend that trust to relying on it to validate the other party's SAML assertion. This eliminates not only the hidden validation latency but also its computational cost to the UA.

Negatively to remark is that this proposal does have one significant security risk. The UA's SIP server can cheat and create an extra authentication object for the UA where it has the Diffie-Hellman private key. With this, the (SIP) server issuing the SAML assertion can successfully launch a Man-in-the-Middle (MITM) attack against two of its UAs. Also, two SIP servers can collude so that either can successfully launch a MITM attack against their UAs. A UA can block this attack if its Diffie-Hellman key is authenticated by a trustworthy third party and this whole object is signed by the SIP server. Moreover, this approach uses two roundtrips, increasing the necessary bandwidth and also the setup time, which may be crucial for many scenarios. For the credential generation, usually a separate component (server) is necessary, so serverless call setup is not supported.

3.7. Asymmetric Key Distribution with In-Band Certificate Exchange

   This is an additional option, which has been defined in [RFC4738].
   It describes the asymmetric key distribution with optional in-band
   certificate exchange.

   Initiator   Responder

   I_MESSAGE =
   HDR, T, [IDi|CERTi], [IDr],
        {SP}, [RAND], SIGNi   --->
                                          R_MESSAGE =
                              <---   HDR, [GenExt(CSB-ID)], T,
                                          RAND, [IDr|CERTr], [SP],
                                          KEMAC, SIGNr

   This option has some advantages compared to the asymmetric key
   distribution stated in Section 3.2. Here, the sender and receiver do
   not need to know the certificate of the other peer in advance as it
   may be sent in the MIKEY Initiator message (if the receiver knows the
   certificate in advance, RFC 3830's MIKEY-RSA mode may be used
   instead). Thus, the receiver of this message can utilize the
   received key material to encrypt the session parameter and send them
   back as part of the MIKEY responder message. The certificate check
   may be done depending on the signing authority. If the certificate
   is signed by a publicly accepted authority, the certificate
   validation can be done in a straightforward manner, by using the
   commonly known certificate authority's public key. In the other
   case, additional steps may be necessary. The disadvantage is that no
   perfect forward secrecy is provided.

   This mode is meant to provide an easy option for certificate
   provisioning when PKI is present and/or required. Specifically in
   SIP, session invitations can be retargeted or forked. MIKEY modes
   that require the Initiator to target a single well-known Responder
   may be impractical here as they may require multiple roundtrips to do
   key negotiation. By allowing the Responder to generate secret
   material used for key derivation, this mode allows for an efficient
   key delivery scheme. Note that the Initiator can contribute to the
   key material since the key is derived from CSB-ID and RAND payloads
   in unicast use cases. This mode is also useful in multicast
   scenarios where multiple clients are contacting a known server and
   are downloading the key. Responder workload is significantly reduced
   in these scenarios compared to MIKEY in public key mode. This is due
   to the fact that the RSA asymmetric encryption requires less effort
   compared to the decryption using the private key (the public key is
   usually shorter than the private key, hence less performance for
   encryption compared to decryption). Examples of deployments where

   this mode can be used are enterprises with PKI, service provider
   setups where the service provider decides to provision certificates
   to its users, etc.

4. Further MIKEY Extensions

   This section will provide an overview about further MIKEY [RFC3830]
   extensions for crypto algorithms and generic payload enhancements, as
   well as enhancements to support the negotiation of security
   parameters for security protocols other than SRTP. These extensions
   have been defined in several additional documents.

4.1. ECC Algorithms Support

   [MSEC-MIKEY] proposes extensions to the authentication, encryption,
   and digital signature methods described for use in MIKEY, employing
   elliptic curve cryptography (ECC). These extensions are defined to
   align MIKEY with other ECC implementations and standards.

   The motivation for supporting ECC within MIKEY stems from the
   following advantages:

   o ECC modes are more and more added to security protocols.

   o ECC support requires considerably smaller keys by keeping the same
      security level compared to other asymmetric techniques (like RSA).
      Elliptic curve algorithms are capable of providing security
      consistent with Advanced Encryption Standard (AES) keys of 128,
      192, and 256 bits without extensive growth in asymmetric key
      sizes.

   o As stated in [MSEC-MIKEY], implementations have shown that
      elliptic curve algorithms can significantly improve performance
      and security-per-bit over other recommended algorithms.

   These advantages make the usage of ECC especially interesting for
   embedded devices, which may have only limited performance and storage
   capabilities.

   [MSEC-MIKEY] proposes several ECC-based mechanisms to enhance the
   MIKEY key distribution schemes:

   o Use of ECC methods extending the Diffie-Hellman key exchange:
      MIKEY-DHSIGN with ECDSA or ECGDSA

   o Use of ECC methods extending the Diffie-Hellman key exchange:
      MIKEY-DHSIGN with ECDH

   o Use of Elliptic Curve Integrated Encryption Scheme (MIKEY-ECIES)

   o Use of Elliptic Curve Menezes-Qu-Vanstone Scheme(MIKEY-ECMQV)

   The following subsections will provide more detailed information
   about the message exchanges for MIKEY-ECIES and MIKEY-ECMQV.

4.1.1. Elliptic Curve Integrated Encryption Scheme application in MIKEY

   The following figure shows the message exchange for the MIKEY-ECIES
   scheme:

   Initiator   Responder

   I_MESSAGE =
   HDR, T, RAND, [IDi|CERTi],
       [IDr], {SP}, KEMAC,
       [CHASH], PKE, SIGNi   --->
                                                  R_MESSAGE =
                                     [<---]   HDR, T, [IDr], V

4.1.2. Elliptic Curve Menezes-Qu-Vanstone Scheme Application in MIKEY

   The following figure shows the message exchange for the MIKEY-ECMQV
   scheme:

   Initiator   Responder

   I_MESSAGE =
   HDR, T, RAND, [IDi|CERTi],
      [IDr], {SP},
      ECCPTi, SIGNi   --->
                                                  R_MESSAGE =
                                     [<---]   HDR, T, [IDr], V

4.2. New MIKEY Payload for Bootstrapping TESLA

   TESLA [RFC4082] is a protocol for providing source authentication in
   multicast scenarios. TESLA is an efficient protocol with low
   communication and computation overhead, which scales to large numbers
   of receivers, and also tolerates packet loss. TESLA is based on
   loose time synchronization between the sender and the receivers.
   Source authentication is realized in TESLA by using Message
   Authentication Code (MAC) chaining. The use of TESLA within the
   Secure Real-time Transport Protocol (SRTP) has been published in
   [RFC4383] targeting multicast authentication in scenarios, where SRTP
   is applied to protect the multimedia data. This solution assumes
   that TESLA parameters are made available by out-of-band mechanisms.

[RFC4442] specifies payloads for MIKEY to bootstrap TESLA for source
authentication of secure group communications using SRTP. TESLA may
be bootstrapped using one of the MIKEY key management approaches
described above by sending the MIKEY message via unicast, multicast,
or broadcast. This approach provides the necessary parameter payload
extensions for the usage of TESLA in SRTP. Nevertheless, if the
parameter set is also sufficient for other TESLA use cases, it can be
applied as well.

4.3. MBMS Extensions to the Key ID Information Type

This extension specifies a new Type (the Key ID Information Type) for
the General Extension Payload. This is used in, e.g., the Multimedia
Broadcast/Multicast Service (MBMS) specified in the 3rd Generation
Partnership Project (3GPP). MBMS requires the use of MIKEY to convey
the keys and related security parameters needed to secure the
multimedia that is multicast or broadcast.

One of the requirements that MBMS puts on security is the ability to
perform frequent updates of the keys. The rationale behind this is
that it will be costly for subscribers to re-distribute the
decryption keys to non-subscribers. The cost for re-distributing the
keys using the unicast channel should be higher than the cost of
purchasing the keys for this scheme to have an effect. To achieve
this, MBMS uses a three-level key management, to distribute group
keys to the clients, and be able to re-key by pushing down a new
group key. MBMS has the need to identify which types of keys are
involved in the MIKEY message and their identity.

[RFC4563] specifies a new Type for the General Extension Payload in
MIKEY, to identify the type and identity of involved keys. Moreover,
as MBMS uses MIKEY both as a registration protocol and a re-key
protocol, this RFC specifies the necessary additions that allow MIKEY
to function both as a unicast and multicast re-key protocol in the
MBMS setting.

4.4. OMA BCAST MIKEY General Extension Payload Specification

The document [RFC4909] specifies a new general extension payload type
for use in the Open Mobile Alliance (OMA) Browser and Content
Broadcast (BCAST) group. OMA BCAST's service and content protection
specification uses short-term key message and long-term key message
payloads that in certain broadcast distribution systems are carried
in MIKEY. The document defines a general extension payload to allow
possible extensions to MIKEY without defining a new payload. The
general extension payload can be used in any MIKEY message and is
part of the authenticated or signed data part. Note that only a
parameter description is included, but no key information.

4.5. Supporting Integrity Transform Carrying the Rollover Counter

   The document [RFC4771] defines a new integrity transform for SRTP
   [RFC3711] providing the option to also transmit the Roll Over Counter
   (ROC) as part of dedicated SRTP packets. This extension has been
   defined for use in the 3GPP multicast/broadcast service. While the
   communicating parties did agree on a starting ROC, in some cases the
   receiver may not be able to synchronize his ROC with the one used by
   the sender even if it is signaled to him out of band. Here the new
   extension provides the possibility for the receiver to re-synchronize
   to the sender's ROC. To signal the use of the new integrity
   transform, new definitions for certain MIKEY payloads need to be
   done. These new definitions comprise the integrity transform itself
   as well as a new integrity transform parameters. Moreover, the
   document specifies additional parameter, to enable the usage of
   different integrity transforms for SRTP and SRTCP.

5. Selection and Interworking of MIKEY Modes

   While MIKEY and its extensions provide a variety of choices in terms
   of modes of operation, an implementation may choose to simplify its
   behavior. This can be achieved by operating in a single mode of
   operation when in the Initiator's role. Where PKI is available
   and/or required, an implementation may choose, for example, to start
   all sessions in RSA-R mode, and it would be trivial for it to act as
   a Responder in public key mode. If envelope keys are cached, it can
   then also choose to do re-keying in shared key mode. It is outside
   the scope of MIKEY or MIKEY extensions if the caching of envelope
   keys is allowed. This is a matter of the configuration of the
   involved components. This local configuration is also outside the
   scope of MIKEY. In general, modes of operation where the Initiator
   generates keying material are useful when two peers are aware of each
   other before the MIKEY communication takes place. If a peer chooses
   not to operate in the public key mode, it may reject the certificate
   of the Initiator. The same applies to peers that choose to operate
   in one of the DH modes exclusively.

   Forward MIKEY modes, where the Initiator provides the key material,
   like public key or shared key mode when used in SIP/SDP may lead to
   complications in some call scenarios, for example, forking scenarios
   where key derivation material gets distributed to multiple parties.
   As mentioned earlier, this may be impractical as some of the
   destinations may not have the resources to validate the message and
   may cause the Initiator to drop the session invitation. Even in the
   case in which all parties involved have all the prerequisites for
   interpreting the MIKEY message received, there is a possible problem
   with multiple Responders starting media sessions using the same key.
   While the SSRCs will be different in most of the cases, they are only

32 bits long and there is a high probability of a two-time pad
problem. This is due to the support of scenarios like forking (see
also Section 5.2) or retargeting (see also Section 5.3), where a two-
time pad occurs if two branches have the same TEK (based on the MIKEY
key establishment) and choose the same 32-bit SSRC for the SRTP
streams and transmit SRTP packets. As suggested earlier, forward
modes are most useful when the two peers are aware of each other
before the communication takes place (as is the case in key renewal
scenarios when costly public key operations can be avoided by using
the envelope key).

The following list gives an idea how the different MIKEY modes may be
used or combined, depending on available key material at the
Initiator side.

1. If the Initiator has a PSK with the Responder, it uses the PSK
    mode.

2. If the Initiator has a PSK with the Responder, but needs PFS or
    knows that the Responder has a policy that both parties should
    provide entropy to the key, then it uses the DH-HMAC mode.

3. If the Initiator has the RSA key of the Responder, it uses the
    RSA mode to establish the TGK. Note that the TGK may be used as
    PSK together with Option 1 for further key management operations.

4. If the Initiator does not expect the responder to have his
    certificate, he may use RSA-R. Using RSA-R, he can provide the
    Initiator's certificate information in-band to the receiver.
    Moreover, the Initiator may also provide a random number that can
    be used by the receiver for key generation. Thus, both parties
    can be involved in the key management. But as the inclusion of
    the random number cannot be forced by the Initiator, true PFS
    cannot be provided. Note that in this mode, after establishing
    the TGK, it may be used as PSK with other MIKEY modes.

5. The Initiator uses DH-SIGN when PFS is required by his policy and
    he knows that the Responder has a policy that both parties should
    provide entropy. Note that also in this mode, after establishing
    the TGK, it may be used as PSK with other MIKEY modes.

6. If no PSK or certificate is available at the Initiator's side
    (and likewise at the responder's side) but lower-level security
    (like TLS or IPsec) is in place the user may use the unprotected
    mode of MIKEY. It has to considered that using the unprotected
    mode enables intermediate nodes like proxies to actually get the
    exchanged master key in plain. This may not be intended,
    especially in cases where the intermediate node is not trusted.

Besides the available key material, choosing between the different
modes of MIKEY depends strongly on the use case. This section will
depict dedicated scenarios to discuss the feasibility of the
different modes in these scenarios. A comparison of the different
modes of operation regarding the influences and requirements to the
deploying infrastructure as well as the cryptographic strength can be
found in [SIP-MEDIA]. The following list provides the most prominent
call scenarios and are matter of further discussion:

o Early Media

o Forking

o Call Transfer/Redirect/Retarget

o Shared Key Conferencing

5.1. MIKEY and Early Media

The term early media describes two different scenarios. The first
one relates to the case where media data are received before the
actual SDP signaling answer has been received. This may arise
through the different latency on the signaling and media path. This
case is often referred to as media before signaling answer. The
second scenario describes the case were media data are send from the
callee before sending the final SIP 200 OK message. This situation
appears usually in call center scenarios, when queuing a waiting loop
or when providing personal ring tones.

In early media scenarios, SRTP data may be received before the answer
over the SIP signaling arrives. The two MIKEY modes, which only
require one message to be transported (Section 3.1 and Section 3.2),
work nicely in early media situations, as both sender and receiver
have all the necessary parameters in place before actually sending/
receiving encrypted data. The other modes, featuring either Diffie-
Hellman key agreement (Section 3.3, Section 3.5, and Section 3.6) or
the enhanced asymmetric variant (Section 3.7), suffer from the
requirements that the Initiator has to wait for the response before
being able to decrypt the incoming SRTP media. In fact, even if
early media is not used, in other words if media is not sent before
the SDP answer, a similar problem may arise from the fact that SIP/
SDP signaling has to traverse multiple proxies on its way back and
media may arrive before the SDP answer. It is expected that this
delay would be significantly shorter than in the case of early media
though.

It is worth mentioning here that security descriptions [RFC4568] have
basically the same problem as the initiating end needs the SDP answer
before it can start decrypting SRTP media.

To cope with the early media problem, there are further approaches to
describe security preconditions [RFC5027]; i.e., certain
preconditions need to be met to enable voice data encryption. One
example, for instance, is that a scenario where a provisional
response, containing the required MIKEY parameter, is sent before
encrypted media is processed.

5.2. MIKEY and Forking

In SIP forking scenarios, a SIP proxy server sends an INVITE request
to more than one location. This means also that the MIKEY payload,
which is part of the SDP, is sent to several (different) locations.
MIKEY modes supporting signatures may be used in forking scenarios
(Section 3.3 and Section 3.7) as here the receiver can validate the
signature. There are limitations with the symmetric key encryption
as well as the asymmetric key encryption modes (Section 3.1 and
Section 3.2). This is due to the fact that in symmetric encryption
the recipient needs to possess the symmetric key before handling the
MIKEY data. For asymmetric MIKEY modes, if the sender is aware of
the forking he may not know in advance to which location the INVITE
is forked and thus may not use the right receiver certificate to
encrypt the MIKEY envelope key. Note that the sender may include
several MIKEY containers into the same INVITE message to cope with
forking, but this requires the knowledge of all forking targets in
advance and also requires the possession of the target certificates.
It is out of the scope of MIKEY to specify behavior in such a case.
MIKEY Diffie Hellman modes or MIKEY-RSA_R Section 3.7 do not have
this problem. In scenarios where the sender is not aware of forking,
only the intended receiver is able to decrypt the MIKEY container.

If forking is combined with early media, the situation gets
aggravated. If MIKEY modes requiring a full roundtrip are used, like
the signed Diffie-Hellman, multiple responses may overload the end
device. An example is forking to 30 destinations (group pickup),
while MIKEY is used with the signed Diffie-Hellman mode together with
security preconditions. Here, every target would answer with a
provisional response, leading to 30 signature validations and Diffie-
Hellman calculations at the sender's site. This may lead to a
prolonged media setup delay.

Moreover, depending on the MIKEY mode chosen, a two-time pad may
occur in dependence of the negotiated key material and the SSRC. For
the non Diffie-Hellman modes other than RSA-R, a two-time pad may
occur when multiple receivers pick the same SSRC.

5.3. MIKEY and Call Transfer/Redirect/Retarget

   In a SIP environment, MIKEY exchange is tied to SDP offer/answer and
   irrespective of the implementation model used for call transfer the
   same properties and limitations of MIKEY modes apply as in a normal
   call setup scenario.

   In certain SIP scenarios, the functionality of redirect is supported.
   In redirect scenarios, the call initiator gets a response that the
   called party for instance has temporarily moved and may be reached at
   a different destination. The caller can now perform a call
   establishment with the new destination. Depending on the originally
   chosen MIKEY mode, the caller may not be able to perform this mode
   with the new destination. To be more precise, MIKEY-PSK and MIKEY-
   DHHMAC require a pre-shared secret in advance. MIKEY-RSA requires
   the knowledge about the target's certificate. Thus, these modes may
   influence the ability of the caller to initiate a session.

   Another functionality that may be supported in SIP is retargeting.
   In contrast to redirect, the call initiator does not get a response
   about the different target. The SIP proxy sends the request to a
   different target about receiving a redirect response from the
   originally called target. This most likely will lead to problems
   when using MIKEY modes requiring a pre-shared key (MIKEY-PSK, MIKEY-
   DHHMAC) or where the caller used asymmetric key encryption (MIKEY-
   RSA) because the key management was originally targeted to a
   different destination.

5.4. MIKEY and Shared Key Conferencing

   First of all, not all modes of MIKEY support shared key conferencing.
   Mainly the Diffie-Hellman modes cannot be used straight-forward for
   conferencing as this mechanism results in a pair wise shared secret
   key. All other modes can be applied in conferencing scenarios by
   obeying the Initiator and Responder roles; i.e., the half roundtrip
   modes need to be initiated by the conferencing unit to be able to
   distribute the conferencing key. The remaining full roundtrip mode,
   MIKEY RSA-R, will be initiated by the client, while the conferencing
   unit provides the conferencing key based on the received certificate.

   An example conferencing architecture is defined in the IETF's XCON
   WG. The scope of this working group relates to a mechanism for
   membership and authorization control, a mechanism to manipulate and
   describe media "mixing" or "topology" for multiple media types
   (audio, video, text), a mechanism for notification of conference-
   related events/changes (for example, a floor change), and a basic
   floor control protocol. A document describing possible use case
   scenarios is available in [RFC4597].

5.5. MIKEY Mode Summary

   The following two tables summarize the discussion from the previous
   subsections. The first table matches the scenarios discussed in this
   section to the different MIKEY modes.

MIKEY mode	Early Media	Secure Forking	Retarget	Redirect	Shared Key Conf
PSK (3.1)	Yes				Yes*
RSA (3.2)	Yes				Yes*
DH-SIGN (3.3)		Yes*	Yes	Yes	
Unprotected (3.4)	Yes				
DH-HMAC (3.5)					
RSA-R (3.7)		Yes	Yes	Yes	Yes

   * In centralized conferencing, the media mixer needs to send the
     MIKEY Initiator message.

   The following table maps the MIKEY modes to key management-related
   properties.

MIKEY mode	Manual Keys	Needs PKI	PFS	Key Generation Involvement
PSK (3.1)	Yes	No	No	Initiator
RSA (3.2)	No	Yes	No	Initiator
DH-SIGN (3.3)	No	Yes	Yes	Both
Unprotected (3.4)	No	No	No	Initiator
DH-HMAC (3.5)	Yes	No	Yes	Both
RSA-R (3.7)	No	Yes	No	Both*

   * Assumed the Initiator provides the (optional) RAND value

6. Transport of MIKEY Messages

   MIKEY defines message formats to transport key information and
   security policies between communicating entities. It does not define
   the embedding of these messages into the used signaling protocol.
   This definition is provided in separate documents, depending on the
   used signaling protocol. Nevertheless, MIKEY can also be transported
   over plain UDP or TCP to port 2269.

   Several IETF-defined protocols utilize the Session Description
   Protocol (SDP, [RFC4566]) to transport the session parameters.
   Examples are the Session Initiation Protocol (SIP, [RFC3261] or the
   Gateway Control Protocol (GCP, [RFC5125]). The transport of MIKEY
   messages as part of SDP is described in [RFC4567]. Here, the

complete MIKEY message is base64 encoded and transmitted as part of
the SDP part of the signaling protocol message. Note that as several
key distribution messages may be transported within one SDP
container, [RFC4567] also comprises an integrity protection regarding
all supplied key distribution attempts. Thus, bidding-down attacks
will be recognized. Regarding RTSP, [RFC4567] defines header
extensions allowing the transport of MIKEY messages. Here, the
initial messages uses SDP, while the remaining part of the key
management is performed using the header extensions.

MIKEY is also applied in ITU-T protocols like H.323, which is used to
establish communication sessions similar to SIP. For H.323, a
security framework exists, which is defined in H.235. Within this
framework, H.235.7 [H.235.7] describes the usage of MIKEY and SRTP in
the context of H.323. In contrast to SIP, H.323 uses ASN.1 (Abstract
Syntax Notation). Thus, there is no need to encode the MIKEY
container as base64. Within H.323, the MIKEY container is binary
encoded.

7. MIKEY Alternatives for SRTP Security Parameter Negotiation

   Besides MIKEY, there exist several approaches to handle the security
   parameter establishment. This is due to the fact that some
   limitations in certain scenarios have been seen. Examples are early
   media and forking situations as described in Section 5. The
   following list provides a short summary about possible alternatives:

   o sdescription - [RFC4568] describes a key management scheme, which
      uses SDP for transport and completely relies on underlying
      protocol security. For transport, the document defines an SDP
      attribute transmitting all necessary SRTP parameter in clear. For
      security, it references TLS and S/MIME. In contrast to MIKEY, the
      SRTP parameter in the Initiator-to-Responder direction is actually
      sent in the message from the Initiator to the Responder rather
      than vice versa. This may lead to problems in early media
      scenarios.

   o sdescription with early media support - [WING-MMUSIC] enhances the
      above scheme with the possibility to also be usable in early media
      scenarios, when security preconditions are not used.

   o Encrypted Key Transport for Secure RTP - [MCGREW-SRTP] is an
      extension to SRTP that provides for the secure transport of SRTP
      master keys, Rollover Counters, and other information, within
      SRTCP. This facility enables SRTP to work for decentralized
      conferences with minimal control, and to handle situations caused
      by SIP forking and early media. It may also be used in
      conjunction with MIKEY.

o Diffie-Hellman support in SDP - [BAUGHER] defines a new SDP
   attribute for exchanging Diffie-Hellman public keys. The
   attribute is an SDP session-level attribute for describing DH
   keys, and there is a new media-level parameter for describing
   public keying material for SRTP key generation.

o DTLS-SRTP describing SRTP extensions for DTLS - [AVT-DTLS]
   describes a method of using DTLS key management for SRTP by using
   a new extension that indicates that SRTP is to be used for data
   protection and that establishes SRTP keys.

o ZRTP - [ZIMMERMANN] defines ZRTP as RTP header extensions for a
   Diffie-Hellman exchange to agree on a session key and parameters
   for establishing SRTP sessions. The ZRTP protocol is completely
   self-contained in RTP and does not require support in the
   signaling protocol or assume a PKI.

There has been a long discussion regarding a preferred key management
approach in the IETF coping with the different scenarios and
requirements continuously sorting out key management approaches.
During IETF 68, three options were considered: MIKEY in an updated
version (referred to as MIKEYv2), ZRTP, and DTLS-SRTP. The potential
key management protocol for the standards track for media security
was voted in favor of DTLS-SRTP. Thus, the reader is pointed to the
appropriate resources for further information on DTLS-SRTP
[AVT-DTLS]. Note that MIKEY has already been deployed for setting up
SRTP security context and is also targeted for use in MBMS
applications.

8. Summary of MIKEY-Related IANA Registrations

For MIKEY and the extensions to MIKEY, IANA registrations have been
made. Here only a link to the appropriate IANA registration is
provided to avoid inconsistencies. The IANA registrations for MIKEY
payloads can be found under
http://www.iana.org/assignments/mikey-payloads. These registrations
comprise the MIKEY base registrations as well as registrations made
by MIKEY extensions regarding the payload.

The IANA registrations for MIKEY port numbers can be found under
http://www.iana.org/assignments/port-numbers (search for MIKEY).

9. Security Considerations

This document does not define extensions to existing protocols. It
rather provides an overview about the set of MIKEY modes and
available extensions and provides information about the applicability
of the different modes in different scenarios to support the decision

making for network architects regarding the appropriate MIKEY scheme
or extension to be used in a dedicated target scenario. Choosing
between the different schemes described in this document strongly
influences the security of the target system as the different schemes
provide different levels of security and also require different
infrastructure support.

As this document is based on the MIKEY base specification as well as
the different specifications of the extensions, the reader is
referred to the original documents for the specific security
considerations.

10. Acknowledgments

The authors would like to thank Lakshminath Dondeti for his document
reviews and for his guidance.

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

    [AVT-DTLS]   McGrew, D. and E. Rescorla, "Datagram Transport
                       Layer Security (DTLS) Extension to Establish Keys
                       for Secure Real-time Transport Protocol (SRTP)",
                       Work in Progress, February 2008.

    [BAUGHER]   Baugher, M. and D. McGrew, "Diffie-Hellman Exchanges
                       for Multimedia Sessions", Work in Progress,
                       February 2006.

    [H.235.7]   ""ITU-T Recommendation H.235.7: Usage of the MIKEY
                       Key Management Protocol for the Secure Real Time
                       Transport Protocol (SRTP) within H.235"", 2005.

    [ISO_sec_time]   ""ISO/IEC 18014 Information technology - Security
                       techniques - Time-stamping services, Part 1-
                       3.http://www.oasis-open.org/committees/
                       documents.php?wg_abbrev=security"", 2002.

   [MCGREW-SRTP]   McGrew, D., "Encrypted Key Transport for Secure
                      RTP", Work in Progress, March 2007.

   [MSEC-MIKEY]   Milne, A., "ECC Algorithms for MIKEY", Work in
                      Progress, June 2007.

   [RFC1305]   Mills, D., "Network Time Protocol (Version 3)
                      Specification, Implementation", RFC 1305,
                      March 1992.

   [RFC2412]   Orman, H., "The OAKLEY Key Determination Protocol",
                      RFC 2412, November 1998.

   [RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G.,
                      Johnston, A., Peterson, J., Sparks, R., Handley, M.,
                      and E. Schooler, "SIP: Session Initiation Protocol",
                      RFC 3261, June 2002.

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

   [RFC4082]   Perrig, A., Song, D., Canetti, R., Tygar, J., and B.
                      Briscoe, "Timed Efficient Stream Loss-Tolerant
                      Authentication (TESLA): Multicast Source
                      Authentication Transform Introduction", RFC 4082,
                      June 2005.

   [RFC4086]   Eastlake, D., Schiller, J., and S. Crocker,
                      "Randomness Requirements for Security", BCP 106,
                      RFC 4086, June 2005.

   [RFC4383]   Baugher, M. and E. Carrara, "The Use of Timed
                      Efficient Stream Loss-Tolerant Authentication
                      (TESLA) in the Secure Real-time Transport Protocol
                      (SRTP)", RFC 4383, February 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.

   [RFC4566]   Handley, M., Jacobson, V., and C. Perkins, "SDP:
                      Session Description Protocol", RFC 4566, July 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.

   [RFC4568]   Andreasen, F., Baugher, M., and D. Wing, "Session
                        Description Protocol (SDP) Security Descriptions for
                        Media Streams", RFC 4568, July 2006.

   [RFC4597]   Even, R. and N. Ismail, "Conferencing Scenarios",
                        RFC 4597, August 2006.

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

   [RFC4738]   Ignjatic, D., Dondeti, L., Audet, F., and P. Lin,
                        "MIKEY-RSA-R: An Additional Mode of Key Distribution
                        in Multimedia Internet KEYing (MIKEY)", RFC 4738,
                        November 2006.

   [RFC4771]   Lehtovirta, V., Naslund, M., and K. Norrman,
                        "Integrity Transform Carrying Roll-Over Counter for
                        the Secure Real-time Transport Protocol (SRTP)",
                        RFC 4771, January 2007.

   [RFC4909]   Dondeti, L., Castleford, D., and F. Hartung,
                        "Multimedia Internet KEYing (MIKEY) General
                        Extension Payload for Open Mobile Alliance BCAST
                        LTKM/STKM Transport", RFC 4909, June 2007.

   [RFC4949]   Shirey, R., "Internet Security Glossary, Version 2",
                        RFC 4949, August 2007.

   [RFC5027]   Andreasen, F. and D. Wing, "Security Preconditions
                        for Session Description Protocol (SDP) Media
                        Streams", RFC 5027, October 2007.

   [RFC5125]   Taylor, T., "Reclassification of RFC 3525 to
                        Historic", RFC 5125, February 2008.

   [SAML_overview]   Huges, J. and E. Maler, "Security Assertion Markup
                        Language (SAML) 2.0 Technical Overview, Working
                        Draft", 2005.

   [SIP-MEDIA]   Wing, D., Fries, S., Tschofenig, H., and F. Audet,
                        "Requirements and Analysis of Media Security
                        Management Protocols", Work in Progress, June 2008.

   [WING-MMUSIC]   Raymond, R. and D. Wing, "Security Descriptions
                      Extension for Early Media", Work in Progress,
                      October 2005.

   [ZIMMERMANN]   Zimmermann, P., Johnston, A., and J. Callas, "ZRTP:
                      Media Path Key Agreement for Secure RTP", Work in
                      Progress, June 2008.

Authors' Addresses

   Steffen Fries
   Siemens Corporate Technology
   Otto-Hahn-Ring 6
   Munich, Bavaria 81739
   Germany

   EMail: steffen.fries@siemens.com

   Dragan Ignjatic
   Polycom
   3605 Gilmore Way
   Burnaby, BC V5G 4X5
   Canada

   EMail: dignjatic@polycom.com