Internet Engineering Task Force (IETF)
Request for Comments: 6509
Category: Informational
ISSN: 2070-1721

M. Groves
CESG

February 2012

MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)

Abstract

This document describes the Multimedia Internet KEYing-Sakai-Kasahara Key Encryption (MIKEY-SAKKE), a method of key exchange that uses Identity-based Public Key Cryptography (IDPKC) to establish a shared secret value and certificateless signatures to provide source authentication. MIKEY-SAKKE has a number of desirable features, including simplex transmission, scalability, low-latency call setup, and support for secure deferred delivery.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6509.

Copyright Notice

Table of Contents

1. Introduction 3
   1.1. Requirements Terminology 3
2. A New MIKEY Mode: MIKEY-SAKKE 4
   2.1. Outline 4
        2.1.1. Parameters 5
        2.1.2. Key Types 5
   2.2. Preparing and Processing MIKEY-SAKKE Messages 6
        2.2.1. Components of the I_MESSAGE 6
        2.2.2. Processing the I_MESSAGE 7
   2.3. Forking and Retargeting 8
   2.4. Group Communications 9
   2.5. Deferred Delivery 9
3. Key Management 9
   3.1. Generating Keys from the Shared Secret Value 9
   3.2. Identifiers 10
   3.3. Key Longevity and Update 11
   3.4. Key Delivery 12
4. Payload Encoding 12
   4.1. Common Header Payload (HDR) 12
   4.2. SAKKE Payload 13
   4.3. SIGN Payload 14
   4.4. IDR Payload 14
5. Applicability of MIKEY-SAKKE Mode 14
6. Security Considerations 14
   6.1. Forking 15
   6.2. Retargeting 16
   6.3. Group Calls 16
   6.4. Deferred Delivery 16
7. IANA Considerations 16
8. References 17
   8.1. Normative References 17
   8.2. Informative References 18
Appendix A. Parameters for Use in MIKEY-SAKKE......................20

1. Introduction

Multimedia Internet KEYing (MIKEY) [RFC3830] defines a protocol framework for key distribution and specifies key distribution methods using pre-shared keys, RSA, and, optionally, a Diffie-Hellman Key Exchange. Since the original specification, several alternative key distribution methods for MIKEY have been proposed such as [RFC4650], [RFC4738], [RFC6043], and [RFC6267].

This document describes MIKEY-SAKKE, a method for key exchange and source authentication designed for use in IP Multimedia Subsystem (IMS) [3GPP.33.328] Media Plane Security, but with potential for wider applicability. This scheme makes use of a Key Management Service (KMS) as a root of trust and distributor of key material. The KMS provides users with assurance of the authenticity of the peers with which they communicate. Unlike traditional key distribution systems, MIKEY-SAKKE does not require the KMS to offer high availability. Rather, it need only distribute new keys to its users periodically.

MIKEY-SAKKE consists of an Identity-based Public Key Cryptography (IDPKC) scheme based on that of Sakai and Kasahara [S-K], and a source authentication algorithm that is tailored to use Identifiers instead of certificates. The algorithms behind this protocol are described in [RFC6507] and [RFC6508].

The primary motivation for the MIKEY protocol design is the low-latency requirement of real-time communication; hence, many of the defined exchanges finish in one-half to one roundtrip. However, some exchanges, such as those described in [RFC6043] and [RFC6267], have been proposed that extend the latency of the protocol with the intent of providing additional security. MIKEY-SAKKE affords similarly enhanced security, but requires only a single simplex transmission (one-half roundtrip).

MIKEY-SAKKE additionally offers support for scenarios such as forking, retargeting, deferred delivery, and pre-encoded content.

1.1. Requirements Terminology

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

2. A New MIKEY Mode: MIKEY-SAKKE

2.1. Outline

The proposed MIKEY mode requires a single simplex transmission. The Initiator sends a MIKEY I_MESSAGE containing SAKKE Encapsulated Data and a signature to the intended recipient. The Responder MUST validate the signature. Following signature validation, the Responder processes the Encapsulated Data according to the operations defined in [RFC6508] to derive a Shared Secret Value (SSV). This SSV is used as the TGK (the TEK Generation Key defined in [RFC3830]).

A verification message from the Responder (as in pre-shared key mode, for example) is not needed, as the parties are mutually authenticated following processing of the single I_MESSAGE. The notation used for MIKEY messages and their payloads in Figure 1, and in the rest of this document, is defined in [RFC3830].

```
         Initiator                                    Responder

         I_MESSAGE =
         HDR, T, RAND, [IDRi], [IDRr], [IDRkmsi], [IDRkmsr],
         [CERT], {SP}, SAKKE, SIGN        --->
```

Figure 1: MIKEY-SAKKE Unicast Mode

The Initiator wants to establish a secure media session with the Responder. The Initiator and the Responder trust a third party, the KMS, which provisions them with key material by a secure mechanism. In addition to the public and secret keys corresponding to their Identifier, the KMS MUST provision devices with its KMS Public Key and, where [RFC6507] is used, its KMS Public Authentication Key. A description of all key material used in MIKEY-SAKKE can be found in Section 2.1.2. The Initiator and the Responder do not share any credentials; instead, the Initiator is able to derive the Responder’s public Identifier.

Implementations MAY provide support for multiple KMSs. In this case, rather than a single KMS, several different KMSs could be involved, e.g., one for the Initiator and one for the Responder. To allow this, each interoperating KMS MUST provide its users with the KMS public keys for every KMS subscriber domain with which its users communicate. It is not anticipated that large mutually communicating groups of KMSs will be needed, as each KMS only needs to provide its domain of devices with key material once per key period (see Section 3.3) rather than to be active in each call.

As MIKEY-SAKKE is based on [RFC3830], the same terminology, processing, and considerations still apply unless otherwise stated. Following [RFC3830], messages are integrity protected and encryption is not applied to entire messages.

2.1.1. Parameters

[RFC6508] requires each application to define the set of public parameters to be used by implementations. The parameters in Appendix A SHOULD be used in MIKEY-SAKKE; alternative parameters MAY be subsequently defined; see Section 4.2.

[RFC6507] requires each application to define the hash function and various other parameters to be used (see Section 4.1 of [RFC6507]). For MIKEY-SAKKE, the P-256 elliptic curve and base point [FIPS186-3] and SHA-256 [FIPS180-3] MUST be used.

2.1.2. Key Types

Users require keys for [RFC6508] and to sign messages. These keys MUST be provided by the users' KMS. It is RECOMMENDED that implementations support the scheme for signatures described in [RFC6507]. Alternatively, RSA signing as defined in [RFC3830] MAY be used.

SAKKE keys

SAKKE requires each user to have a Receiver Secret Key, created by the KMS, and the KMS Public Key. For systems that support multiple KMSs, each user also requires the KMS Public Key of every KMS subscriber domain with which communication is authorized.

ECCSI keys

If the Elliptic Curve-based Certificateless Signatures for Identity-based Encryption (ECCSI) signatures are used, each user requires a Secret Signing Key and Public Validation Token, created by the KMS, and the KMS Public Authentication Key. For systems that support multiple KMSs, each user also requires the KMS Public Authentication Key of every KMS subscriber domain with which communication is authorized.

If instead RSA signatures are to be used, certificates and corresponding private keys MUST be supplied.

2.2. Preparing and Processing MIKEY-SAKKE Messages

Preparation and parsing of MIKEY messages are as described in Sections 5.2 and 5.3 of [RFC3830]. Error handling is described in Section 5.1.2, and replay protection guidelines are in Section 5.4 of [RFC3830]. In the following, we describe the components of MIKEY-SAKKE messages and specify message processing and parsing rules in addition to those in [RFC3830].

2.2.1. Components of the I_MESSAGE

MIKEY-SAKKE requires a single simplex transmission (a half roundtrip) to establish a shared TGK. The I_MESSAGE MUST contain the MIKEY Common Header Payload HDR defined in [RFC6043] together with the timestamp payload in order to provide replay protection. The HDR field contains a CSB_ID (Crypto Session Bundle ID) randomly selected by the Initiator. The V bit in the HDR payload MUST be set to ’0’ and ignored by the Responder, as a response is not expected in this mode. The timestamp payload MUST use TS type NTP-UTC (TS type 0) or NTP (TS type 1) as defined in Section 6.6 of [RFC3830] so that the Responder can determine the Identifiers used by the Initiator (see Section 3.2). It is RECOMMENDED that the time always be specified in UTC.

The I_MESSAGE MUST be signed by the Initiator following either the procedure to sign MIKEY messages specified in [RFC3830], or using [RFC6507] as specified in this document. The SIGN payload contains this signature. Thus, the I_MESSAGE is integrity and replay protected. The ECCSI signature scheme [RFC6507] SHOULD be used. If this signature scheme is used, then the Initiator MUST NOT include a CERT payload. To form this signature type, the Initiator requires a Secret Signing Key that is provided by the KMS.

Other signature types defined for use with MIKEY MAY be used. If signature types 0 or 1 (RSA) are used, then the Initiator SHOULD include a CERT payload; in this case, the CERT payload MAY be left out if it is expected that the Responder is able to obtain the certificate in some other manner. If a CERT payload is included, it MUST correspond to the private key used to sign the I_MESSAGE.

The Initiator MUST include a RAND payload in the I_MESSAGE, as this is used to derive session keys.

The identities of the Initiator, Responder, the Initiator’s KMS (root of trust for authentication of the Initiator), and the Responder’s KMS (root of trust for authentication of the Responder) MAY be contained in the IDRi, IDRr, IDRkmsi, and IDRkmsr I_MESSAGEs, respectively. The ID Payload with Role Indicator (IDR) is defined in

[RFC6043] and modified in Section 4.4. When used, this payload provides the Identifier for any of the Initiator, the Responder, and their respective KMSs.

The ID Role MUST be the Initiator (value 1) for the IDRi payload and Responder (value 2) for the IDRr payload. The Initiator's ID is used to validate signatures [RFC6507]. If included, the IDRi payload MUST contain the URI of the Initiator incorporated in the Identifier used to sign the I_MESSAGE (see Section 3.2). If included, the IDRr payload MUST contain the URI of the Responder incorporated in the Identifier that the Initiator used in SAKKE (see Section 3.2). If included, the ID Role MUST be the Initiator's KMS (value 6) for the IDRkmsi payload and Responder's KMS (value 7) for the IDRkmsr payload and MUST correspond to the KMS used as root of trust for the signature (for the IDRkmsi payload) and the KMS used as the root of trust for the SAKKE key exchange (for the IDRkmsr payload).

It is OPTIONAL to include any IDR payloads, as in some user groups Identifiers could be inferred by other means, e.g., through the signaling used to establish a call. Furthermore, a closed user group could rely on only one KMS, whose identity will be understood and need not be included in the signaling.

The I_MESSAGE MUST contain a SAKKE payload constructed as defined in Section 4.2.

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).

2.2.2. Processing the I_MESSAGE

The Responder MUST process the I_MESSAGE according to the rules specified in Section 5.3 of [RFC3830]. The following additional processing MUST also be applied.

* If the Responder does not support the MIKEY-SAKKE mode of operation, or otherwise cannot correctly parse the received MIKEY message, then it SHOULD send an error message "Unsupported message type" (Error no. 13). Error no. 13 is not defined in [RFC3830], and so implementations compliant with [RFC3830] MAY return an "Unspecified error" (Error no. 12).

- The Responder MAY compare the IDi payload against his local policy to determine whether he wishes to establish secure communications from the Initiator. If the Responder’s policy does not allow this communication, then the Responder MAY respond with an "Auth failure" error (Error no. 0).

- If the Responder supports MIKEY-SAKKE and has determined that it wishes to establish secure communications with the Initiator, then it MUST verify the signature according to the method described in Section 5.2.2 of [RFC6507] if it is of type 2, or according to the certificate used if a signature of type 0 or 1 is used. If the verification of the signature fails, then an "Auth failure" error (Error no. 0) MAY be sent to the Initiator.

- If the authentication is successful, then the Responder SHALL process the SAKKE payload and derive the SSV according to the method described in [RFC6508].

2.3. Forking and Retargeting

Where forking is to be supported, Receiver Secret Keys can be held by multiple devices. To facilitate this, the Responder needs to load his Receiver Secret Key into each of his devices that he wishes to receive MIKEY-SAKKE communications. If forking occurs, each of these devices can then process the SAKKE payload, and each can verify the Identifier of the Initiator as they hold the KMS Public Authentication Key. Therefore, the traffic keys could be derived by any of these devices. However, this is the case for any scheme employing simplex transmission, and it is considered that the advantages of this type of scheme are significant for many users. Furthermore, it is for the owner of the Identifier to determine on which devices to allow his Receiver Secret Key to be loaded. Thus, it is anticipated that he would have control over all devices that hold his Receiver Secret Key. This argument also applies to applications such as call centers, in which the security relationship is typically between the call center and the individual calling the center, rather than the particular operative who receives the call.

Devices holding the same Receiver Secret Key ought to each hold a different Secret Signing Key corresponding to the same Identifier. This is possible because the Elliptic Curve-based Certificateless Signatures for Identity-based Encryption (ECCSI) scheme allows multiple keys to be generated by KMS for the same Identifier.

Secure retargeted calls can only be established in the situation where the Initiator is aware of the Identifier of the device to whom the call is being retargeted; in this case, the Initiator ought to initiate a new MIKEY-SAKKE session with the device to whom it has been retargeted (if willing to do so). Retargeting an Initiator’s call to another device (with a different Identifier) is to be viewed as insecure when the Initiator is unaware that this has occurred, as this prevents authentication of the Responder.

2.4. Group Communications

SAKKE supports key establishment for group communications. The Initiator needs to form an I_MESSAGE for each member in the group, each using the same SSV. Alternatively, a bridge can be used. In this case, the bridge forms an I_MESSAGE for each member of the group. Any member of the group can invite new members directly by forming an I_MESSAGE using the group SSV.

2.5. Deferred Delivery

Deferred delivery / secure voicemail is fully supported by MIKEY-SAKKE. A deferred delivery server that supports MIKEY-SAKKE needs to store the MIKEY-SAKKE I_MESSAGE along with the encrypted data. When the recipient of the voicemail requests his data, the server needs to initiate MIKEY-SAKKE using the stored I_MESSAGE. Thus, the data can be received and decrypted only by a legitimate recipient, who can also verify the Identifier of the sender. This requires no additional support from the KMS, and the deferred delivery server need not be trusted, as it is unable to read or tamper with the messages it receives. Note that the deferred delivery server does not need to fully implement MIKEY-SAKKE merely to store and forward the I_MESSAGE.

The deferred delivery message needs to be collected by its recipient before the key period in which it was sent expires (see Section 3.3 for a discussion of key periods). Alternatively, if greater longevity of deferred delivery payloads is to be supported, the Initiator needs to include an I_MESSAGE for each key period during the lifetime of the deferred delivery message, each using the same SSV. In this case, the deferred delivery server needs to forward the I_MESSAGE corresponding to the current key period to the recipient.

3. Key Management

3.1. Generating Keys from the Shared Secret Value

Once a MIKEY-SAKKE I_MESSAGE has been successfully processed by the Responder, he will share an authenticated SSV with the Initiator. This SSV is used as the TGK. The keys used to protect application traffic are derived as specified in [RFC3830].

3.2. Identifiers

One of the primary features and advantages of Identity-Based Encryption (IBE) is that the public keys of users are their Identifiers, which can be constructed by their peers. This removes the need for Public Key or Certificate servers, so that all data transmission per session can take place directly between the peers, and high-availability security infrastructure is not needed. In order for the Identifiers to be constructable, they need to be unambiguously defined. This section defines the format of Identifiers for use in MIKEY-SAKKE.

If keys are updated regularly, a KMS is able to revoke devices. To this end, every Identifier for use in MIKEY-SAKKE MUST contain a timestamp value indicating the key period for which the Identifier is valid (see Section 3.3). This document uses a year and month format to enforce monthly changes of key material. Further Identifier schemes MAY be defined for communities that require different key longevity.

An Identifier for use in MIKEY-SAKKE MUST take the form of a timestamp formatted as a US-ASCII string [ASCII] and terminated by a null byte, followed by identifying data which relates to the identity of the device or user, also represented by a US-ASCII string and terminated by a null byte.

For the purposes of this document, the timestamp MUST take the form of a year and month value, formatted according to [ISO8601], with the format "YYYY-MM", indicating a four-digit year, followed by a hyphen "-", followed by a two-digit month.

For the Identifier scheme defined in this document, the identifying data MUST take the form of a constrained "tel" URI. If an alternative URI scheme is to be used to form SAKKE Identifiers, a subsequent RFC MUST define constraints to ensure that the URI can be formed unambiguously. The normalization procedures described in Section 6 of [RFC3986] MUST be used as part of the constraining rules for the URI format. It would also be possible to define Identifier types that used identifying data other than a URI.

The restrictions for the "tel" URI scheme [RFC3966] for use in MIKEY-SAKKE Identifiers are as follows:

* the "tel" URI for use in MIKEY-SAKKE MUST be formed in global notation,

* visual separators MUST NOT be included,

- the "tel" URI MUST NOT include additional parameters, and

- the "tel" URI MUST NOT include phone-context parameters.

These constraints on format are necessary so that all parties can unambiguously form the "tel" URI.

For example, suppose a user's telephone number is +447700900123 and the month is 2011-02, then the user's Identifier is defined as the ASCII string:

```
2011-02\0tel:+447700900123\0,
```

where '\0' denotes the null 8-bit ASCII character 0x00.

If included in I_MESSAGE, the IDRi and IDRr payloads MUST contain the URI used to form the Identifier. The value of the month used to form the Identifiers MUST be equal to the month as specified by the data in the timestamp payload.

3.3. Key Longevity and Update

Identifiers for use in MIKEY-SAKKE change regularly in order to force users to regularly update their key material; we term the interval for which a key is valid a "key period". This means that if a device is compromised (and this is reported procedurally), it can continue to communicate with other users for at most one key period. Key

periods SHOULD be indicated by the granularity of the format of the timestamp used in the Identifier. In particular, the Identifier scheme in this document uses monthly key periods. Implementations MUST allow devices to hold two periods' keys simultaneously to allow for differences in system time between the Initiator and Responder.

Where a monthly key period applies, it is RECOMMENDED that implementations receive the new key material before the second-to-last day of the old month, commence allowing receipt of calls with the new key material on the second-to-last day of the old month, and continue to allow receipt calls with the old key material on the first and second days of the new month. Devices SHOULD cease to receive calls with key material corresponding to the previous month on the third day of the month; this is to allow compromised devices to be keyed out of the communicating user group.

KMSs MAY update their KMS Master Secret Keys and KMS Master Secret Authentication Keys. If such an update is not deemed necessary, then the corresponding KMS Public Keys and KMS Public Authentication Keys will be fixed. If KMS keys are to be updated, then this update MUST occur at the change of a key period, and new KMS Public Key(s) and KMS Public Authentication Key(s) MUST be provided to all users with their user key material.

It is NOT RECOMMENDED for KMSs to distribute multiple key periods' keys simultaneously, as this prevents the periodic change of keys from excluding compromised devices.

3.4. Key Delivery

This document does not seek to restrict the mechanisms by which the necessary key material might be obtained from the KMS. The mechanisms of [RFC5408] are not suitable for this application, as the MIKEY-SAKKE protocol does not require public parameters to be obtained from a server: these are fixed for all users in order to facilitate interoperability and simplify implementation.

The delivery mechanism used MUST provide confidentiality to all secret keys, integrity protection to all keys, and mutual authentication of the device and the KMS.

4. Payload Encoding

This section describes the new SAKKE payload and also the payloads for which changes have been made compared to [RFC3830]. A detailed description of MIKEY payloads is provided in [RFC3830].

4.1. Common Header Payload (HDR)

An additional value is added to the data type and next payload fields.

* Data type (8 bits): describes the type of message.

Data type	Value	Comment
SAKKE msg	26	Initiator's SAKKE message

Table 1: Data type (additions)

* Next payload (8 bits): identifies the payload that is added after this payload.

Next payload	Value	Section
SAKKE	26	4.2

Table 2: Next payload (additions)

- V (1 bit): flag to indicate whether a response message is expected ('1') or not ('0'). It MUST be set to '0' and ignored by the Responder in a SAKKE message.

4.2. SAKKE Payload

The SAKKE payload contains the SAKKE Encapsulated Data as defined in [RFC6508].

```
1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next payload  ! SAKKE params  !   ID scheme   !  SAKKE data   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~ length (cont) !                  SAKKE data                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Table 3: SAKKE payload

- Next payload (8 bits): identifies the payload that is added after this payload.

- SAKKE params (8 bits): indicates the SAKKE parameter set to be used.

SAKKE params	Value
Parameter Set 1 (See Appendix A)	1

Table 4: SAKKE params

- ID scheme (8 bits): indicates the SAKKE identifier scheme to be used.

ID scheme	Value
tel URI with monthly keys (See Section 3.2)	1

Table 5: ID scheme

- SAKKE data length (16 bits): length of SAKKE data (in bytes).

- SAKKE data (variable): the SAKKE Encapsulated Data formatted as defined in Section 4 of [RFC6508].

4.3. SIGN Payload

To enable use of the ECCSI signature algorithm, which has efficiency benefits for use with Identity-based encryption, we define an additional signature type.

- S type (4 bits): indicates the signature algorithm applied by the Signer.

S type	Value	Comments
ECCSI	2	ECCSI signature

Table 6: S type (additions)

4.4. IDR Payload

The IDR payload was defined in [RFC6043], but its definition only provided the facility to identify one KMS per exchange. Since it is possible that different KMSs could be used by the Initiator and Responder, this payload is extended to define an ID Role for the KMS of the Initiator and the KMS of the Responder.

- ID Role (8 bits): specifies the sort of identity.

ID Role	Value
Initiator’s KMS (IDRkmsi)	6
Responder’s KMS (IDRkmsr)	7

Table 7: ID Role (additions)

5. Applicability of MIKEY-SAKKE Mode

MIKEY-SAKKE is suitable for use in a range of applications in which secure communications under a clear trust model are needed. In particular, the KMS need not provide high availability, as it is only necessary to provide a periodic refresh of key material. Devices are provided with a high level of authentication, as the KMS acts as a root of trust for both key exchange and signatures.

6. Security Considerations

Unless explicitly stated, the security properties of the MIKEY protocol as described in [RFC3830] apply to MIKEY-SAKKE as well. In addition, MIKEY-SAKKE inherits some properties of Identity-based cryptography. For instance, by concatenating the "date" with the URI to form the Identifier, the need for any key revocation mechanisms is virtually eliminated. It is NOT RECOMMENDED for KMSs to distribute multiple months' keys simultaneously in an IBE system, as this prevents the monthly change of keys from excluding compromised devices.

The solution proposed provides protection suitable for high-security user groups, but is scalable enough that it could be used for large numbers of users. Traffic keys cannot be derived by any infrastructure component other than the KMS.

The effective security of the public parameters defined in this document is 112 bits, as this is the security offered by the prime p of size 1024 bits used in SAKKE (see Section 7 of [RFC6508]). For similar parameter sizes, MIKEY-SAKKE provides equivalent levels of effective security to other schemes of this type (such as [RFC6267]). For reasons of efficiency and security, it is RECOMMENDED to use a mode of AES-128 [AES] in the traffic application to which MIKEY-SAKKE supplies key material, but users SHOULD be aware that 112 bits of security are offered by the defined public parameters. Following [SP800-57], this choice of security strength is appropriate for use to protect data until 2030.

User identities cannot be spoofed, since the Public Authentication Token is tied to the Identifier of the sender by the KMS. In particular, the Initiator is provided with assurance that nobody other than a holder of the legitimate Receiver Secret Key can process the SAKKE Encapsulated Data, and the signature binds the holder of the Initiator's Secret Signing Key to the I_MESSAGE. Since these keys are provided via a secure channel by the KMS, mutual authentication is provided. This mechanism protects against both passive and active attacks.

If there were a requirement that a caller remain anonymous from any called parties, then it would be possible to remove the signature from the protocol. A called user could then decide, according to local policy, whether to accept such a secure session.

6.1. Forking

Where forking is used, the view is taken that it is not necessary for each device to have a separate Receiver Secret Key. Rather, where a user wishes his calls to be forked between his devices, he loads the same Receiver Secret Key onto each of them. This does not compromise his security as he controls each of the devices, and is consistent with the Initiator's expectation that he is authenticated to the owner of the Identifier he selected when initiating the call.

6.2. Retargeting

Since the Initiator is made aware by the forwarding server of the change to the Identifier of the Responder, he creates an I_MESSAGE that can only be processed by this legitimate Responder. The Initiator MAY also choose to discontinue the session after checking his local policy.

6.3. Group Calls

Any device that possesses an SSV can potentially provide it securely to any other device using SAKKE. Thus, group calls can either be established by an Initiator, or can be extended to further Responders by any party to whom the original Initiator has sent an I_MESSAGE.

The Initiator in this context MAY be a conference bridge. If a mode of operation in which a bridge has no knowledge of the SSV is needed, the role of the MIKEY-SAKKE Initiator MUST be carried out by one or more of the communicating parties, not by the bridge.

Where multi-way communications (rather than broadcast) are needed, the application using the supplied key material MUST ensure that a suitable Initialization Vector (IV) scheme is used in order to prevent cryptovariable re-use.

6.4. Deferred Delivery

Secure deferred delivery is supported in a manner such that no trust is placed on the deferred delivery server. This is a significant advantage, as it removes the need for secure infrastructure components beyond the KMS.

7. IANA Considerations

This document defines new values for the namespaces Data Type, Next Payload, and S type defined in [RFC3830], and for the ID Role namespace defined in [RFC6043]. The following IANA assignments have been added to the MIKEY Payload registry:

- 26 - Data type (see Table 1)
- 26 - Next payload (see Table 2)
- 2 - S type (see Table 6)
- ID Role (see Table 7)
    - 6 - Initiator’s KMS (IDRkmsi)
    - 7 - Responder’s KMS (IDRkmsr)

The SAKKE payload defined in Section 4.2 defines two fields for which IANA has created and now maintains namespaces in the MIKEY Payload registry. These two fields are the 8-bit SAKKE Params field, and the 8-bit ID Scheme field. IANA has recorded the pre-defined values defined in Section 4.2 for each of the two name spaces. Values in the range 1-239 SHOULD be approved by the process of Specification Required, values in the range 240-254 are for Private Use, and the values 0 and 255 are Reserved according to [RFC5226].

Initial values for the SAKKE Params registry are given below. Assignments consist of a SAKKE parameters name and its associated value.

Value	SAKKE params	Definition
0	Reserved	
1	Parameter Set 1	See Appendix A
2-239	Unassigned	
240-254	Private Use	
255	Reserved	

Initial values for the ID scheme registry are given below. Assignments consist of a name of an identifier scheme name and its associated value.

Value	ID Scheme	Definition
0	Reserved	
1	tel URI with monthly keys	See Section 3.2
2-239	Unassigned	
240-254	Private Use	
255	Reserved	

8. References

8.1. Normative References

[AES] NIST, "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001, http://www.itl.nist.gov/fipspubs/by-num.htm.

[ASCII] American National Standards Institute, "Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)", ANSI X3.4, 1986.

[FIPS180-3] Federal Information Processing Standards Publication (FIPS PUB) 180-3, "Secure Hash Standard (SHS)", October 2008.

[FIPS186-3] Federal Information Processing Standards Publication (FIPS PUB) 186-3, "Digital Signature Standard (DSS)", June 2009.

[ISO8601] "Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO 8601:2004(E), International Organization for Standardization, December 2004.

[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.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC6043] Mattsson, J. and T. Tian, "MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6043, March 2011.

[RFC6507] Groves, M., "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)", RFC 6507, February 2012.

[RFC6508] Groves, M., "Sakai-Kasahara Key Encryption (SAKKE)", RFC 6508, February 2012.

[SP800-57] Barker, E., Barker, W., Burr, W., Polk, W., and M. Smid, "Recommendation for Key Management - Part 1: General (Revised)", NIST Special Publication 800-57, March 2007.

8.2. Informative References

[3GPP.33.328] 3GPP, "IP Multimedia Subsystem (IMS) media plane security", 3GPP TS 33.328 10.0.0, April 2011.

[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.

[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.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5408] Appenzeller, G., Martin, L., and M. Schertler, "Identity-Based Encryption Architecture and Supporting Data Structures", RFC 5408, January 2009.

[RFC6267] Cakulev, V. and G. Sundaram, "MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6267, June 2011.

[S-K] Sakai, R., Ohgishi, K., and M. Kasahara, "ID based cryptosystem based on pairing on elliptic curves", Symposium on Cryptography and Information Security - SCIS, 2001.

Appendix A. Parameters for Use in MIKEY-SAKKE

[RFC6508] requires each application to define the set of public parameters to be used by implementations. Parameter Set 1 is defined in this appendix. Descriptions of the parameters are provided in Section 2.1 of [RFC6508].

```
      n       = 128

      p       = 997ABB1F 0A563FDA 65C61198 DAD0657A
                416C0CE1 9CB48261 BE9AE358 B3E01A2E
                F40AAB27 E2FC0F1B 228730D5 31A59CB0
                E791B39F F7C88A19 356D27F4 A666A6D0
                E26C6487 326B4CD4 512AC5CD 65681CE1
                B6AFF4A8 31852A82 A7CF3C52 1C3C09AA
                9F94D6AF 56971F1F FCE3E823 89857DB0
                80C5DF10 AC7ACE87 666D807A FEA85FEB

      q       = 265EAEC7 C2958FF6 99718466 36B4195E
                905B0338 672D2098 6FA6B8D6 2CF8068B
                BD02AAC9 F8BF03C6 C8A1CC35 4C69672C
                39E46CE7 FDF22286 4D5B49FD 2999A9B4
                389B1921 CC9AD335 144AB173 595A0738
                6DABFD2A 0C614AA0 A9F3CF14 870F026A
                A7E535AB D5A5C7C7 FF38FA08 E2615F6C
                203177C4 2B1EB3A1 D99B601E BFAA17FB

      Px      = 53FC09EE 332C29AD 0A799005 3ED9B52A
                2B1A2FD6 0AEC69C6 98B2F204 B6FF7CBF
                B5EDB6C0 F6CE2308 AB10DB90 30B09E10
                43D5F22C DB9DFA55 718BD9E7 406CE890
                9760AF76 5DD5BCCB 337C8654 8B72F2E1
                A702C339 7A60DE74 A7C1514D BA66910D
                D5CFB4CC 80728D87 EE9163A5 B63F73EC
                80EC46C4 967E0979 880DC8AB EAE63895

      Py      = 0A824906 3F6009F1 F9F1F053 3634A135
                D3E82016 02990696 3D778D82 1E141178
                F5EA69F4 654EC2B9 E7F7F5E5 F0DE55F6
                6B598CCF 9A140B2E 416CFF0C A9E032B9
                70DAE117 AD547C6C CAD696B5 B7652FE0
                AC6F1E80 164AA989 492D979F C5A4D5F2
                13515AD7 E9CB99A9 80BDAD5A D5BB4636
                ADB9B570 6A67DCDE 75573FD7 1BEF16D7
       g      = 66FC2A43 2B6EA392 148F1586 7D623068
                C6A87BD1 FB94C41E 27FABE65 8E015A87
                371E9474 4C96FEDA 449AE956 3F8BC446
                CBFDA85D 5D00EF57 7072DA8F 541721BE
                EE0FAED1 828EAB90 B99DFB01 38C78433
                55DF0460 B4A9FD74 B4F1A32B CAFA1FFA
                D682C033 A7942BCC E3720F20 B9B7B040
                3C8CAE87 B7A0042A CDE0FAB3 6461EA46

       Hash   = SHA-256 (defined in [FIPS180-3]).
```

Author’s Address

Michael Groves
CESG
Hubble Road
Cheltenham
GL51 8HJ
UK
EMail: Michael.Groves@cesg.gsi.gov.uk