Internet Engineering Task Force
Internet Draft
Expires: August 2002

M. Euchner
Siemens AG

February 2002

HMAC-authenticated Diffie-Hellman for MIKEY
(draft-euchner-mikey-dhhmac-00.txt)

Status of this memo

Comments should be sent to the MSEC WG mailing list at msec@securemulticast.org and to the author.

Abstract

This document describes a key management protocol variant for the multimedia Internet keying (MIKEY). In particular, the classic Diffie-Hellman key agreement protocol is used for key establishment in conjunction with a keyed hash (HMAC-SHA1) for achieving mutual authentication and message integrity of the key management messages exchanged. This MIKEY variant is called the HMAC-authenticated Diffie-Hellmann. It addresses the real-time aspects of multimedia key management in MIKEY.

Table of Contents

1. Introduction

As pointed out in MIKEY (see [1]), secure real-time multimedia applications demand a particular adequate key management scheme that cares for how to securely and efficiently establish dynamic session keys in a conversational multimedia scenario.

In general, MIKEY scenarios cover peer-to-peer, simple-one-to-many and small-sized groups. MIKEY in particular, describes three key management schemes for the peer-to-peer case that all finish their task within one round trip:

- a symmetric key distribution protocol based upon pre-shared master keys;

- a public-key encryption-based key distribution protocol assuming a public-key infrastructure with RSA-based private/public keys and digital certificates;

- and a Diffie-Hellman key agreement protocol deploying digital signatures and certificates.

All these three key management protocols are designed such that they complete their work within just one round trip. This requires depending on loosely synchronized clocks and deploying timestamps within the key management protocols.

However, it is known [4] that each of the three key management schemes has its subtle constraints and limitations:

- The symmetric key distribution protocol is simple to implement but does not nicely scale in any larger configuration of potential peer entities due to the need of mutually pre-assigned shared master secrets.

Moreover, the security provided does not achieve the property of perfect forward secrecy; i.e. compromise of the shared master secret would render past and even future session keys susceptible to compromise.
Further, the generation of the session key happens just at the initiator. Thus, the responder has to fully trust the initiator on choosing a good and secure session secret; the responder neither is able to participate in the key generation nor to influence that process. This is considered as a specific limitation in less trusted environments.

- The public-key encryption scheme depends upon a public-key infrastructure that certifies the private-public keys by issuing and maintaining digital certificates. While such a key management scheme provides full scalability in large networked configurations, public-key infrastructures are still not widely available and in general, implementations are significantly more complex.
Further additional round trips might be necessary for each side in order to ascertain verification of the digital certificates.
Finally, as in the symmetric case, the responder depends completely upon the initiator choosing good and secure session keys.

- The third MIKEY key management protocol deploys the Diffie-Hellman key agreement scheme and authenticates the exchange of the Diffie-Hellman half-keys in each direction by using a digital signature upon. As in the previous method, this introduces the dependency upon a public-key infrastructure with its strength on scalability but also the limitations on computational costs in performing the asymmetric long-integer operations and the potential need for additional communication for verification of the digital certificates.
However, the Diffie-Hellman key agreement protocol is known for its subtle security strengths in that it is able to provide full perfect secrecy and further have both parties actively involved in session key generation.

This document describes a fourth key management scheme for MIKEY that could somehow be seen as a synergetic optimization among the pre-shared key distribution scheme and the Diffie-Hellman key agreement. The idea of that protocol is to apply the Diffie-Hellman key agreement but instead of deploying a digital signature for authenticity of the exchanged keying material rather uses a keyed-hash upon using symmetrically pre-assigned shared secrets. This combination of security mechanisms is called the HMAC-authenticated Diffie-Hellman key agreement for MIKEY (DH-HMAC).

1.1. Notational Conventions

The key word "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.

1.2. Definitions

auth-key	pre-shared authentication key
IDa, IDb	Identity of sender and receiver
k_p	common Diffie-Hellman shared secret
T	timestamp
x, y	secret, random value

1.3. Abbreviations

DH	Diffie-Hellman
DH-HMAC	HMAC-authenticated Diffie-Hellman
HMAC	keyed Hash Message Authentication
HMAC-SHA1	HMAC using SHA1 as hash function
MIKEY	Multimedia Internet KEYing
PMK	Pre-Master Key
RSA	Rivest, Shamir and Adleman
TEK	Traffic Encryption Key

2. Scenario

The HMAC-authenticated Diffie-Hellman key agreement protocol (DH-HMAC) for MIKEY addresses the same scenarios and scope as the other three key management schemes in MIKEY address.

DH-HMAC is applicable in a small-sized, peer-to-peer group where no access to a public-key infrastructure can be assumed available. Rather, pre-shared master secrets are available among the entities in such a small-sized group.

In a small-sized group, it is assumed that each client will be setting up a session key for its outgoing links with its peer using the DH-MAC key agreement protocol.

As is the case for the other three MIKEY key management protocol, DH-HMAC assumes loosely synchronized clocks among the entities in the small group.

2.1. DH-HMAC Security Protocol

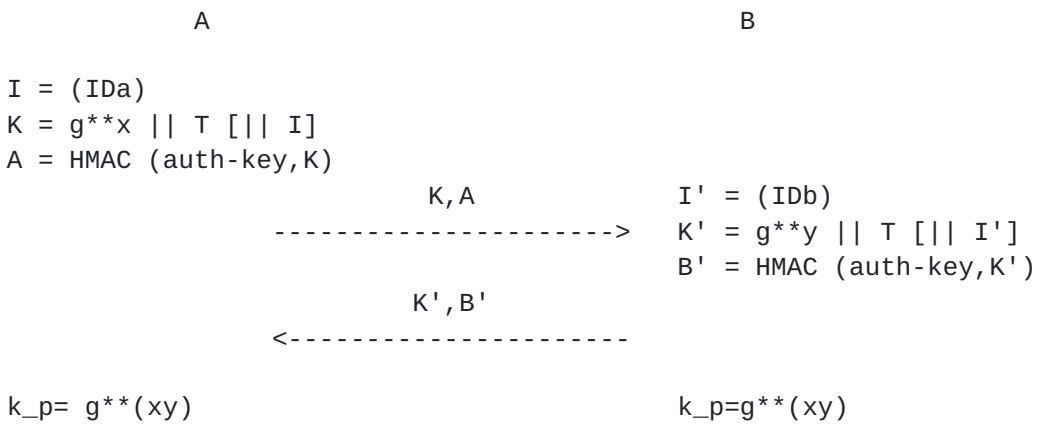


Figure 1: HMAC-authenticated Diffie-Hellman key based exchange, where x and y are randomly chosen respectively by A and B.

The key exchange is done according to Figure 1. The initiator chooses a random value x, and sends an HMACed message including g**x and a timestamp to the responder (optionally also including its identity).

The group parameters (e.g., the group G) are a set of parameters chosen by the initiator. The responder chooses a random positive integer y, and sends an HMACed message including g**y and the timestamp to the initiator (optionally also providing its identity).

Both parties then calculate the PMK, g**(xy).

The HMAC authentication is due to provide authentication of the DH half-keys, and is necessary to avoid man-in-the-middle attacks.

This approach is the less expensive than digitally signed Diffie-Hellman. It requires first of all, that both sides compute one exponentiation and one HMAC, then one HMAC verification and finally another Diffie-Hellman exponentiation.
With off-line pre-computation, the initial Diffie-Hellman MAY be computed before the key management transaction and thereby MAY further reduce the overall round trip delay as well as reduce the risk of denial-of-service attacks.

Processing of the TEK SHALL be accomplished as described in MIKEY chapter 4.

The computed HMAC result A or B' SHALL be conveyed in the DH-HMAC

payload field of the MIKEY payload as specified in chapter 5 of MIKEY.

3. DH-HMAC payload formats

This section specifies the payload formats and data type values for DH-HMAC.

3.1. Common header payload

For DH-HMAC the following data type SHALL be used:

Data type	Value	Comment
DHHMAC init	7	Initiator's DH-HMAC exchange message
DHMHAC resp	8	Responder's DH-HMAC exchange message

The Error payload data type SHALL be applied by the responder in case of a decoding error or of a failed HMAC authentication verification.

Hash func	Value	Comments
SHA-1	0	Mandatory, Default (see [SHA1])
SHA-1-96	5	Optional, SHA1 truncated to the 96 leftmost bits of SHA-1 result when represented in network byte order.

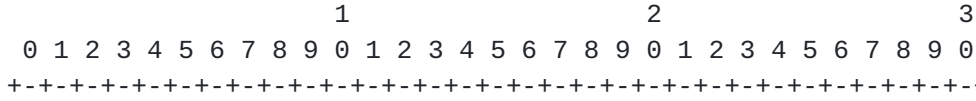
SHA-1 is the default hash function that MUST be implemented as part of the DH-HMAC. The length of the HMAC result is 160 bits. SHA-1-96 produces a slightly shorter HMAC result where the SHA-1 result SHALL be truncated to the 96 leftmost bits when represented in network byte order. This will save some bandwidth.

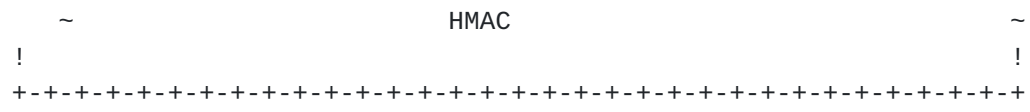
3.2. DH-HMAC data payload

There is no separate payload defined for DH-HMAC. Rather, the DH payload data as defined in MIKEY Appendix A.4 SHALL be used.

3.3. HMAC payload

The HMAC payload carries the computed HMAC value upon the DH-MAC message and corresponding payload data. The HMAC MUST always be the last payload in the DH-HMAC exchange messages.




HMAC

* HMAC: The computed HMAC upon the entire message. The hashing function within HMAC shall be used as indicated by the hash func value (0 or 5).

Appendix B. - Payload usage summary

This section describes the relationship of DH-HMAC messages and the MIKEY payload types.

Depending on the type of message, different payloads MUST and MAY be included. There are six distinct types of messages:

* Pre-shared key transport message

* Public key transport message

* Verification message (for either pre-shared key or public key)

* DH exchange message (bi-directional)

* DH-HMAC exchange message (bi-directional)

* Error message

	Message Type				
Payload type	PS	PK	DH	DH-HMAC	Ver	Error
PS data	M	-	-	-	-	-
PK data	-	M*	-	-	-	-
DH data	-	-	M	M	-	-
Ver msg	-	-	-	-	M	-
Error	-	-	-	-	-	M
Timestamp	M	M	M	O	-	O
ID	O	O	O	O	O	O
Signature	-	O	M	-	-	-
HMAC	-	-	-	M	-	-
Certificate	-	O	O	-	-	-
Cert hash	-	O	O	-	-	-
n_start	O	O	O	O	-	-
n_end	O	O	O	O	-	-
SPI	O	O	O	O	-	-
SP	O	O	O	O	-	O

When a payload is not included, the default values for the information carried by it SHALL be used (when applicable). The

following table summarizes what messages may be included in a specific message.

4. Security Considerations

This document addresses key management security issues throughout. For a comprehensive explanation of security considerations, please refer to MIKEY section 10. In addition to that, the following security considerations apply in particular to this document:

Other than the MIKEY pre-shared and public-key based key distribution protocols, the Diffie-Hellman key agreement protocol features a security property called perfect forward secrecy. That is, that even if the long-term master would be compromised at some point in time, this would not render past session keys compromised.

Further, Diffie-Hellman key management protocol is not a strict key distribution protocol per se. Actually, both parties involved in the protocol exchange are able to equally contribute to the common Diffie-Hellman master session key. This reduces the risk of either party cheating or unintentionally generating a weak session key. In order for Diffie-Hellman key agreement to be secure, each party shall generate its x or y values using a strong, unpredictable pseudo-random generator. Further these values x or y shall be kept private. It is recommended that these secret values be destroyed once the common Diffie-Hellman shared secret key has been established.

For the sake of improved performance and reduced round trip delay either party may off-line pre-compute its public Diffie-Hellman half-key.

As such, DH-HMAC but also digitally signed DH provides a far superior security level over the pre-shared or public-key based key distribution protocol in that respect.

This document describes the HMAC-authenticated Diffie-Hellman key agreement protocol that completely avoids digital signatures and the associated public-key infrastructure as would be necessary for the X.509 RSA public-key based key distribution protocol or the digitally signed Diffie-Hellman key agreement protocol as described in MIKEY. Public-key infrastructures may not always be available in certain environments nor may they be deemed adequate for real-time multimedia applications when taking additional steps for certificate validation and certificate revocation methods with additional round-trips into account.

The HMAC computation corroborates for authentication and message

integrity of the exchanged Diffie-Hellman half-keys and associated messages. The authentication is absolute necessary in order to avoid man-in-the-middle attacks on the exchanged messages in transit.

Using HMAC in conjunction with a strong one-way hash function such as SHA1 may be achieved more efficiently in software than expensive public-key operations. This yields a particular performance benefit of DH-HMAC over signed DH or the public-key encryption protocol.

DH-HMAC optionally features a variant where the HMAC-SHA-1 result is truncated to 96-bit instead of 160 bits. It is believed that although the truncated HMAC appears significantly shorter, the security provided would not suffer; it appears even reasonable that the shorter HMAC could provide increased security against known-plaintext crypt-analysis, see RFC 2104 for more details. In any way, truncated DH-HMAC is able to reduce the bandwidth during Diffie-Hellman key agreement and yield better round trip delay on low-bandwidth links. If very high security level is desired for long-term secrecy of the negotiated Diffie-Hellman shared secret, longer hash values may be deployed such as SHA256, SHA384 or SHA512 provide, possibly in conjunction with stronger Diffie-Hellman groups.

5. Acknowledgements

6. Intellectual Property Rights

This proposal is in full conformity with [RFC-2026].

Siemens may have patent rights on technology described in this document which employees of Siemens contribute for use in IETF standards discussions. In relation to any IETF standard incorporating any such technology, Siemens hereby agrees to license on fair, reasonable and non-discriminatory terms, based on reciprocity, any patent claims it owns covering such technology, to the extent such technology is essential to comply with such standard.

7. Normative References

[1] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman; "MIKEY:Multimedia Internet KEYing", Internet Draft, Work in Progress (MSEC WG)

[2] H. Krawczyk, M. Bellare, R. Canetti; "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[3] NIST, FIBS-PUB 180-1, "Secure Hash Standard", April 1995,

http://csrc.nist.gov/fips/fip180-1.ps

8. Informative References

[4] A.J. Menezes, P v. Oorschot, S. Vanstone: "Applied Cryptography", CRC Press, 1996

9. Author's Address

Please address all comments to:

Martin Euchner
Email: martin.euchner@icn.siemens.de
Phone: +49 89 722 55790
Fax: +49 89 722 47713

Siemens AG
ICN M SR 3
Hofmannstr. 51
81359 Munich, Germany

10. Expiration Date

This Internet Draft expires on 30 August 2002.