Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 23, 2007

A. Milne

M. Blaser
D. Brown
E. Chin
Certicom Corp.
L. Dondeti
QUALCOMM, Inc.
March 22, 2007

ECC Algorithms for MIKEY
draft-ietf-msec-mikey-ecc-02

Status of this Memo

Copyright Notice

Abstract

This document 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.

It should be noted that this document is not self-contained; it uses the notations and definitions of [RFC3830].

Table of Contents

1. Introduction

This document describes additional algorithms for use in MIKEY. The document assumes that the reader is familiar with the MIKEY protocol.

The MIKEY protocol [RFC3830] defines three methods for transporting or establishing keys: with the use of a pre-shared key, public-key encryption (MIKEY-RSA), and Diffie-Hellman (DH) key exchange (MIKEY-DHSIGN). This document extends MIKEY-DHSIGN to use Elliptic Curve Digital Signature Algorithm (ECDSA) or Elliptic Curve German Digital Signature Algorithm (ECGDSA) as the signature algorithm and further extends MIKEY-DHSIGN to use Elliptic Curve Diffie-Hellman (ECDH) groups. In addition, this document introduces two new methods based on the the Elliptic Curve Integrated Encryption Scheme (ECIES) and Elliptic Curve Menezes-Qu-Vanstone (ECMQV) in exchanges similar to those of MIKEY-RSA, and name these methods MIKEY-ECIES and MIKEY-ECMQV respectively.

Implementations have shown that elliptic curve algorithms can significantly improve performance and security-per-bit over other recommended algorithms. The purpose of this document is to expand the options available to implementers of MIKEY to take advantage of these benefits.

In addition, elliptic curve algorithms are capable of providing security consistent with AES keys of 128, 192, and 256 bits without extensive growth in asymmetric key sizes. The following table, taken from [HOF] and [LEN], gives approximate comparable key sizes for symmetric systems, ECC systems, and DH/DSA/RSA systems. The estimates are based on the running times of the best algorithms known today.

Symmetric	ECC2N	ECP	DH/DSA/RSA
80	163	192	1024
128	283	256	3072
192	409	384	7680
256	571	521	15360

Table 1: Comparable key sizes

Thus, for example, when securing a 192-bit symmetric key, it is prudent to use either 409-bit ECC2N, 384-bit ECP, or 7680-bit DH/DSA/RSA. With smaller key sizes the symmetric keys would be underprotected.

Section 2 describes the extension of MIKEY-DHSIGN to use the ECDSA or ECGDSA signature algorithm. Section 3 describes the extension of MIKEY-DHSIGN to use ECDH groups. Section 4 describes the MIKEY-ECIES

method. Section 5 describes the MIKEY-ECMQV method. Section 6 describes additional payloads required to support these new methods.

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 [RFC2119].

2. MIKEY-DHSIGN with ECDSA or ECGDSA

MIKEY-DHSIGN is described in Section 3.3 of [RFC3830]. The Initiator's message includes SIGNi, a signature covering the Initiator's message. As well, the Responder's message includes SIGNr, a signature covering the Responder's message. According to Section 4.2.6 of [RFC3830], the signature algorithm applied is defined by, and dependent on the certificate used. It is MANDATORY to support RSA PKCS#1, v1.5, and it is RECOMMENDED to support RSA PSS. Instead of these signature algorithms, ECDSA or ECGDSA may be used to allow shorter and more efficient signatures.

ECDSA signatures are detailed in [ANSI-X9.62] and ECGDSA signatures are detailed in [ISO-IEC-15946-2]. Curve selection and other parameters will be defined by, and dependent on the certificate used. When generating signatures, the hash function that MUST be used depends on the key size, as follows:

ECC2N	ECP	Hash To Use
163	192	SHA-1
233	224	SHA-224
283	256	SHA-256
409	384	SHA-384
571	521	SHA-512

Table 2: Hash to use with ECDSA and ECGDSA

The signature payload (SIGN) specified in Section 6.5 of [RFC3830] can be used without modification. Two additional S types for ECDSA and ECGDSA is defined as follows:

S type	Value	Comments
ECDSA	2	ECDSA signature [ANSI_X9.62]
ECGDSA	3	ECGDSA signature [ISO/IEC_15946-2]

[RFC3279] describes algorithms and identifiers for Internet X.509 certificates and CRLs. It includes ECC algorithms and identifiers.

To use the ECDSA or ECGDSA signature algorithm with Elliptic Curve Diffie-Hellman, this extension to MIKEY-DHSIGN may be combined with the extension described in Section 3.

3. MIKEY-DHSIGN with ECDH

MIKEY-DHSIGN is described in Section 3.3 of [RFC3830]. According to Section 4.2.7 of [RFC3830], the support for OAKLEY 5 is MANDATORY and support for OAKLEY 1 and OAKLEY 2 is OPTIONAL. Instead of these Diffie-Hellman (DH) groups, elliptic curve Diffie-Hellman (ECDH) groups may significantly improve performance and security.

The ECDH groups to be used by MIKEY are the groups recommended by NIST in FIPS 186-2 [FIPS-186-2]. Detailed descriptions of the ECDH groups can be found in each of FIPS 186-2 [FIPS-186-2] and SEC 2 [SEC2]. The ECDH groups use elliptic curves over GF[2^N] with N prime or over GF[P] with P prime. Eleven of the groups proposed here have been assigned identifiers by IANA [IANA] and the remaining five might later be assigned identifiers by IANA. The group with IANA number 6 is described in [ANSI-X9.62] and [SEC2], with object identifier sect163r1, but it is not one of the fifteen curves that NIST recommends [FIPS-186-2]. The remaining NIST recommended groups are suggested and anticipated to be assigned IANA numbers as specified in Table 3.

id	Group Type	Group Description	NIST Name	SEC 2 OID
22	2 ECP	ECPRGF192Random	P-192	secp192r1
23	3 EC2N	EC2NGF163Random	B-163	sect163r2
7	3 EC2N	EC2NGF163Koblitz	K-163	sect163k1
6	3 EC2N	EC2NGF163Random2	none	sect163r1
24	2 ECP	ECPRGF224Random	P-224	secp224r1
25	3 EC2N	EC2NGF233Random	B-233	sect233r1
26	3 EC2N	EC2NGF233Koblitz	K-233	sect233k1
19	2 ECP	ECPRGF256Random	P-256	secp256r1
8	3 EC2N	EC2NGF283Random	B-283	sect283r1
9	3 EC2N	EC2NGF283Koblitz	K-283	sect283k1
20	2 ECP	ECPRGF384Random	P-384	secp384r1
10	3 EC2N	EC2NGF409Random	B-409	sect409r1
11	3 EC2N	EC2NGF409Koblitz	K-409	sect409k1
21	2 ECP	ECPRGF521Random	P-521	secp521r1
12	3 EC2N	EC2NGF571Random	B-571	sect571r1
13	3 EC2N	EC2NGF571Koblitz	K-571	sect571k1

Table 3: Recommended Groups and Names

The ECDH groups in Table 3 are arranged into 5 classes, corresponding

to approximately equivalent security strengths. To encourage interoperability, implementations that support one of these classes, SHOULD support the one group in that class that is defined over a prime field (which will be one of P-192, P-224, P-256, P-384, or P-521). Implementations SHOULD support one of P-256 or P-384. Implementations MAY support any set of groups.

The DH data payload (DH) specified in Section 6.4 of [RFC3830] can be used without modification. Additional DH-Group identifiers are required as follows:

DH-Group	Value
ECPRGF192Random / P-192 / secp192r1	3
EC2NGF163Random / B-163 / sect163r2	4
EC2NGF163Koblitz / K-163 / sect163k1	5
EC2NGF163Random2 / none / sect163r1	6
ECPRGF224Random / P-224 / secp224r1	7
EC2NGF233Random / B-233 / sect233r1	8
EC2NGF233Koblitz / K-233 / sect233k1	9
ECPRGF256Random / P-256 / secp256r1	10
EC2NGF283Random / B-283 / sect283r1	11
EC2NGF283Koblitz / K-283 / sect283k1	12
ECPRGF384Random / P-384 / secp384r1	13
EC2NGF409Random / B-409 / sect409r1	14
EC2NGF409Koblitz / K-409 / sect409k1	15
ECPRGF521Random / P-521 / secp521r1	16
EC2NGF571Random / B-571 / sect571r1	17
EC2NGF571Koblitz / K-571 / sect571k1	18

When using the ECDH groups, the DH-value in the DH data payload (DH) is the octet string representation specified in ANSI X9.62 [ANSI-X9.62] and [SEC1].

If the initiator chooses secret i and the responder chooses secret r, then the raw shared secret is the x-coordinate(only) of (ir)*G.

To use ECDH and ECDSA signature algorithm or to use ECDH and ECGDSA signature algorithm, this extension to MIKEY-DHSIGN may be combined with the extension described in Section 2.

4. MIKEY-ECIES

The Elliptic Curve Integrated Encryption Scheme (ECIES) is a public-key encryption scheme based on ECC. Section 3.2 of [RFC3830] already specifies a public-key encryption method (MIKEY-RSA). Here we describe the new MIKEY-ECIES method.

```
Initiator                                            Responder

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

                                                     R_MESSAGE =
                                         [<---]   HDR, T, [IDr], V
```

As with the MIKEY-RSA case, the main objective of the Initiator's message is to transport one or more TGKs and a set of security parameters to the Responder in a secure manner.

With MIKEY-RSA, the TGKs are encrypted with an "envelope key". However, ECIES uses a symmetric encapsulation algorithm, so encrypting an envelope key (to be used with another symmetric method to decrypt the actual payload) would be redundant. As a result, the PKE payload is not used.

The ECCPT contains the elliptic curve point that represents the ephemeral public key required for ECIES.

As in MIKEY-RSA, the KEMAC contains a set of encrypted sub-payloads and a MAC:

KEMAC = E(encr_key, IDi || {TGK}) || MAC

The encr_key and auth_key are derived from the ECIES-derived key by using the algorithm described in Section 4.1.4 of [RFC3830], in identical fashion as the envelope key used in the MIKEY-RSA.

Both SIGNi and SIGNr will use either ECDSA or ECGDSA as a signature algorithm, as described in Section 2.

As in MIKEY-RSA, it is possible to cache the ECIES-derived key, so that it can be used as a pre-shared key.

ECIES is described in detail in [SEC1]. For ECIES, the key derivation function that MUST be used is ANSI-X9.63-KDF as described in [SEC1]. As well, the MAC scheme that MUST be used is HMAC-SHA-1-160. The 'standard' elliptic curve Diffie-Hellman primitive MUST be

used (as opposed to 'cofactor'). The symmetric encryption scheme that MUST be used depends on the key size, as follows:

ECC2N	ECP	Symmetric Cipher To Use
163	192	3DES-CBC
233	224	AES-128-CBC
283	256	AES-128-CBC
409	384	AES-256-CBC
571	521	AES-256-CBC

Table 4: Symmetric cipher to use with ECIES

5. MIKEY-ECMQV

ECMQV (Elliptic Curve Menezes-Qu-Vanstone) is a 3-pass or 1-pass protocol that has been standardized in ANSI X9.63 [ANSI-X9.63]. Both modes of ECMQV provide mutual authentication between the communicating parties and key establishment for the secure transport of data. Here we describe the new MIKEY-ECMQV method based on the 1-pass protocol.

```
Initiator                                        Responder

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

                                                 R_MESSAGE =
                                       [<---]   HDR, T, [IDr], V
```

The ECCPT contains the elliptic curve point that represents the ephemeral public key contributed by the Initiator.

As in MIKEY-RSA, the KEMAC contains a set of encrypted sub-payloads and a MAC:

KEMAC = E(encr_key, IDi || {TGK}) || MAC

The encr_key and auth_key are derived from the ECMQV-derived key by using the algorithm described in Section 4.1.4 of [RFC3830], in identical fashion as the envelope key used in the MIKEY-RSA.

1-pass ECMQV is described in detail in ANSI X9.63 [ANSI-X9.63].

6. Additional Payload Encoding

6.1. ECC Point payload (ECCPT)

The ECCPT payload carries a point on the elliptic curve used in MIKEY-ECIES and MIKEY-ECMQV. The payload identifier is 22.

```
                     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  ! Point length                  !  Pt data ...  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        Point data                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

* Next payload (8 bits): identifies the payload that is added after this payload. See Section 6.1 for values.

* Point length (16 bits): length of the Point data field (in *bits*).

* Point data (variable length): point data, padded to end on a 32-bit boundary, encoded in octet string representation specified in ANSI X9.62 [ANSI-X9.62] and [SEC1]. Uncompressed format MUST be supported. Hybrid and compressed formats MAY be supported.

7. Security Considerations

Since this document proposes new methods for use within MIKEY, many of the security considerations contained within [RFC3830] apply here as well. Some of the methods proposed in this document offer higher cryptographic strength than those proposed in [RFC3830]. In particular, there are elliptic curves corresponding to each of the symmetric key sizes 80 bits, 128 bits, 192 bits, and 256 bits. This allows the MIKEY key exchange to offer security comparable with higher-strength AES algorithms and SHA implementations. The methods proposed in this document are among those standardized by NIST in FIPS 186-2 [FIPS-186-2], by the SECG in SEC2 [SEC2], and by ANSI in ANSI X9.62 [ANSI-X9.62] and X9.63 [ANSI-X9.63].

8. IANA Considerations

This document adds entries to existing MIKEY namespaces in Section 2 (S types in signature payloads), Section 3 (DH Group identifier in DH payloads), and Section 6.1 (ECCPT payload identifier).

9. References

9.1. Normative References

[ANSI-X9.62]
American National Standards Institute, "ANSI X9.62: Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", 2005.

[ANSI-X9.63]
American National Standards Institute, "ANSI X9.63: Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography", 2001.

[FIPS-186-2]
National Institute of Standards and Technology, "FIPS 186-2 Digital Signature Standard", 2000.

[IANA] Internet Assigned Numbers Authority, "Attribute Assigned Numbers.", <http://www.isi.edu/in-notes/iana/assignments/ipsec-registry>.

[ISO-IEC-15946-2]
International Organization for Standardization and International Electrotechnical Commission, "ISO/IEC 15946-2: Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves -- Part 2: Digital signatures", 2002.

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

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

[SEC1] Standards for Efficient Crytopgraphy Group, "Elliptic Curve Cryptography", September 2000.

[SEC2] Standards for Efficient Crytopgraphy Group, "Recommended Elliptic Curve Domain Parameters", September 2000.

9.2. Informative References

[HOF] Hoffman, P. and H. Orman, "Determining strengths for public keys used for exchanging symmetric keys", August 2000.

[LEN] Lenstra, A. and E. Verhuel, "Selecting cryptographic key sizes", <http://www.cryptosavvy.com>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Andrew Milne

Mitch Blaser
Certicom Corp.
5520 Explorer Drive
Mississauga, Ontario L4W 5L1
CANADA

Phone: +1-905-507-4220
Fax: +1-905-507-4230
Email: mblaser@certicom.com
URI: http://www.certicom.com

Daniel R. L. Brown
Certicom Corp.
5520 Explorer Drive
Mississauga, Ontario L4W 5L1
CANADA

Phone: +1-905-507-4220
Fax: +1-905-507-4230
Email: dbrown@certicom.com
URI: http://www.certicom.com

Eugene Chin
Certicom Corp.
5520 Explorer Drive
Mississauga, Ontario L4W 5L1
CANADA

Phone: +1-905-507-4220
Fax: +1-905-507-4230
Email: echin@certicom.com
URI: http://www.certicom.com

Lakshminath Dondeti
QUALCOMM, Inc.
5775 Morehouse Drive
San Diego, CA
USA

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

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).