MSEC Working Group
Internet Draft
Expires December 2005

A. Milne
M. Blaser
D. Brown
Certicom

L. Dondeti
Qualcomm

June 2005

ECC Algorithms For MIKEY

<draft-ietf-msec-mikey-ecc-00.txt>

Status of this Memo

IPR Statement

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 [MIKEY].

Comments

Comments on this draft should be addressed to msec@securemulticast.org.

1. Introduction

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

RFC 3830 [MIKEY] defines three methods of key exchange during establishment of a TGK. The pre-shared key (MIKEY-PSA) and public key (MIKEY-RSA) methods are mandatory, while support for Diffie-Hellman (MIKEY-DHSIGN) is optional. Elliptic curve Diffie-Hellman (ECDH) can be used in the MIKEY Diffie-Hellman method; we specify this mode in this document. In addition, the elliptic curve protocols MCMQV and ECIES can be used in MIKEY in exchanges similar to those of MIKEY-RSA; we specify these modes, and name them 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	ECC	DH/DSA/RSA
80	163	1024
128	283	3072
192	409	7680
256	571	15360

Table 1: Comparable key sizes

Thus, for example, when securing a 192-bit symmetric key, it is prudent to use either 409-bit ECC or 7680-bit DH/DSA/RSA. Of course it is possible to use shorter asymmetric keys, but it should be recognized in this case that the security of the system is likely dependent on the strength of the public-key algorithm and claims such as "this system is highly secure because it uses 192-bit encryption" are misleading.

Section 2 below describes the use of elliptic curve methods for public-key authentication and encryption. Section 3 describes the use of ECIES (The Elliptic Curve Integrated Encryption Scheme). Section 4 describes methods for Elliptic Curve Diffie-Hellman, including fifteen ECDH groups. Section 5 describes the MIKEY-MQV method. Section 6 includes modifications to specific sections of [MIKEY].

2. Use of EC methods with public-key encryption (MIKEY-RSA)

MIKEY-RSA specifies the use of RSA PKCS#1, v1.5 as mandatory and RSA PSS as recommended. This section describes how ECDSA signatures may be used for certificate signature and signature operations, enabling use of smaller signatures and certificates. This section also describes the ECIES encryption/decryption scheme, for use with elliptic curve key pairs.

2.1 ECDSA signature

Section 6.5 of [MIKEY] describes the signature payload for the PK and DH exchange messages. The ECDSA signature algorithm can be applied to allow shorter and more-efficient signatures.

ECDSA signatures are detailed in ANSI X9.62 [X9.62]. Curve selection and other parameters will be defined by, and dependent on the certificate used.

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

3. MIKEY-ECIES

MIKEY's public-key encryption method (MIKEY-RSA) uses public key methods securely to transmit keying material between communicating parties having previously acquired one another's public keys. The Elliptic Curve Integrated Encryption Scheme (ECIES) specifies how two communicating parties having previously acquired one another's public keys--assuming these are EC public keys--may use these keys to transmit encrypted and authenticated messages. This section therefore proposes how ECIES may be used in MIKEY. We call this scheme MIKEY-ECIES.

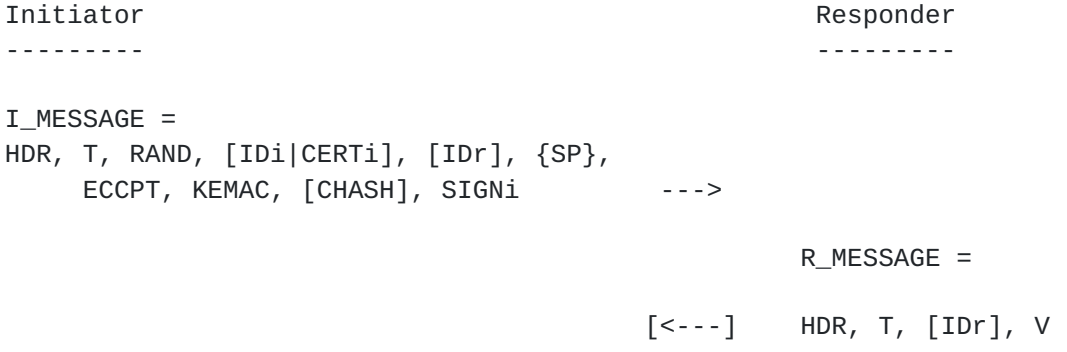
We propose that ECIES be used as follows:

1. The ephemeral public key transmitted by the initiator, is transmitted in an ECCPT payload (see section 5.1) preceding the KEMAC payload.
2. The ciphertext and message digest required under ECIES are transmitted in the KEMAC payload, as in other forms of the MIKEY protocol.
3. The encryption key and HMAC key in use in the KEMAC are those extracted from the shared key derived using ECIES.
4. The PKE payload is not used.

Note that this differs from the 'envelope key' method used in the MIKEY-RSA form of the protocol. ECIES, however, 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.

Note also that the derived ECIES key can be used as an input for the key generation algorithm described in section 4.1.4 of RFC 3830, in identical fashion as is the envelope key used in the MIKEY-RSA method.

A MIKEY-ECIES exchange



Note also that the derived key generated may also be cached for the lifetime of a CSB, at the direction of the Initiator, and used as a preshared key, as described for the envelope key in RFC 3830, section 3.2.

ECIES options

IEEE 1363A describes a number of options for ECIES. We recommend that in MIKEY-ECIES, the following options be understood as given:

-- the KDF in use for the generation of the ephemeral key pairs shall be ECSVDP-DHC, without compatibility for the corresponding -DH primitive

-- DHAES mode shall not be used.

4. Use of EC methods with Diffie-Hellman key exchange (MIKEY-DHSIGN)

The MIKEY-DHSIGN key exchange method is described in Section 3.3 of [MIKEY]. Section 4.2.7 of [MIKEY] specifies the use of OAKLEY group 5 as mandatory and groups 1 and 2 as optional. However, implementations have shown that users of elliptic curve groups can significantly improve performance and security by using groups other than the Oakley Groups 1, 2, or 5.

The DH data payload specified in Section 6.4 of [MIKEY] can be used without modification. The data in the KEMAC payload when using these groups is the octet string representation specified in ANSI X9.62 [X9.62], ANSI X9.63 [X9.63], FIPS 186-2 [FIPS 186-2], and IEEE P1363 of the point on the curve chosen by taking the randomly chosen secret Ka and computing Ka*P, where * is the repetition of the group addition and double operations.

An updated DH-Group table (as shown in Section 6.4 of [MIKEY]) will be specified upon assignment of IANA numbers for the groups described in section 8. See also the section "IANA considerations" in this document.

5. Using ECMQV in MIKEY (MIKEY-MQV)

MQV is a protocol primitive equivalent to simultaneous Diffie-Hellman key exchange and digital signature authentication, achievable in a single transmission. The S_i and S_r values function as implicit signatures proving possession of the private key corresponding to the communicating party's known public key.

ECMQV (Elliptic Curve Menezes-Qu-Vanstone) is a three-pass or 1-pass protocol that has been standardized in ANSI X9.63. Both modes of ECMQV provide mutual authentication between the communicating parties and key establishment for the secure transport of data; the 1-pass version is thus particularly attractive for MIKEY, as an alternative method of establishing a secure channel for the transport of the TGK. In this draft, we propose a fourth mode in MIKEY, called MIKEY-MQV, in which ECMQV is used in this fashion.

A MIKEY-MQV exchange proceeds in similar fashion to the MIKEY-RSA exchange; the PKE is absent, and an ECCPT payload (see section 5.1) MUST precede the KEMAC payload in the intiator's first message:

```
Initiator                                   Responder
---------                                   ---------

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

... the responder's acknowledgement, as in MIKEY-RSA, is optional, and the initiator indicates whether a response is required. If present, the acknowledgement is of the form:

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

The ECCPT payload carries Q_(e,I) as per the nomenclature of ANSI X9.63 -- the ephemeral public key contributed by the initiator.

The encr_key and auth_key used in forming the KEMAC are the encryption and authentication keys extracted from the derived key arrived at via MQV.

1-pass ECMQV algorithm steps

1-pass ECMQV is described in detail in ANSI X9.63-2001; for a detailed specification for implementation purposes, see that document. The following discussion is provided here for information purposes only, to clarify the mechanism, and to ease mapping it to the protocol components in this draft proposal.

Note that 1-pass MQV differs from 3-pass MQV in that only three key pairs are used as inputs: the initiator's public, private key pair, the respondent's public, private key pair, and the ephemeral public, private key pair contributed by the initiator. The respondent's public, private key pair is used twice--effectively replacing the ephemeral key pair contributed by the respondent in the 3-pass method.

Initiator transformation
---

1. Initiator selects an ephemeral private key k_A from the set {1..n-1}, where n is the order of the base point P for the curve in use. Initiator computes the corresponding ephemeral public key R_A = (k_A)P, and sends R_A to the respondent. The ephemeral public key R_A derived is the value Q_(e,I) in the protocol described above. The selection of k_A and the generation of R_A from it are covered in detail in X9.63-2001 section 5.2.1--Key Pair Generation Primitive.
2. Initiator calculates the shared secret value z as follows (see X9.63-2001 section 5.5)--
   - 2a. Initiator uses their own private and public key, and the ephemeral private and public key they generated in step 1 as inputs (d_1, Q_1) and (d_2, Q_2) respectively.
   - 2b. Initiator uses the respondent's public key for the values Q_3 and Q_4.
   - 2c. Using the associate value function avf (see X9.63-2001 section 5.6.1), and the values d_1, d_2 and Q_2 as above (their own private key, and the ephemeral key pair private and public values), the Initiator calculates the implicit signature S_i = d_2 + (avf(Q_2) x d_1) mod n.
   - 2d. Using the associate value function avf, the signature just calculated, the system cofactor h, and the respondent's public key, the Initiator finds the EC point P = h x S_i x (Q_4 + (avf(Q_4) x Q_3)).
   - 2e. Initiator verifies that P != 0.
   - 2f. Initiator sets z = x_P, where x_P is the x-coordinate of P.
3. Initiator converts z to bit string Z, using the convention specified in X9.63-2001 section 4.3.3.
4. Initiator uses the key derivation function described in X9.63-2001 section 5.6.3 with the hash function given in table MQV_PARAMS to derive the keying data; the length of the key to be generated is also given in MQV_PARAMS.

Respondent transformation

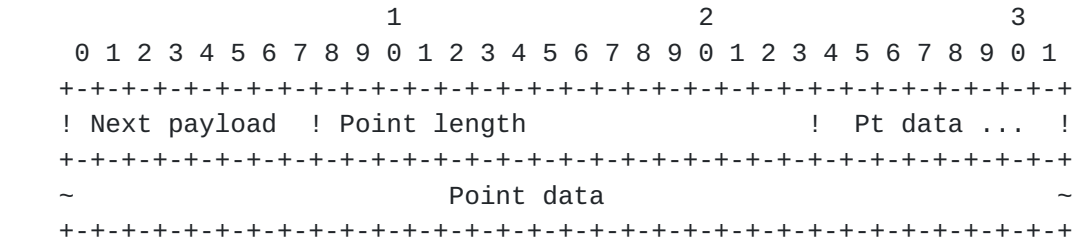
The respondent transformation is parallel to the initiator transformation, except that the respondent does not generate an ephemeral key pair, and the inputs d_1, Q_1, d_2, Q_2, Q_3 and Q_4 come from different sources. Again, see X9.63-2001 section 5.5 for a detailed description appropriate for implementation purposes.

1. Respondent verifies that Q_(e, U) is a valid key for the domain parameters (see X9.63-2001 section 5.2.2).
2. Respondent calculates the shared secret value z as follows (see X9.63-2001 section 5.5)--
   2a. Respondent uses their own private and public key in step 1 as inputs for both (d_1, Q_1) and (d_2, Q_2).
   2b. Respondent uses the initiator's public key for the value Q_3, and the initiator's ephemeral public key for the value Q_4.
   2c. Using the associate value function avf (see X9.63-2001 section 5.6.1), and the values d_1, d_2 and Q_2 as above (their own private key is both d_1 and d_2, while Q_2 is their public key), the Respondent calculates the implicit signature S_r = d_2 + (avf(Q_2) x d_1) mod n.
   2d. Using the associate value function avf, the signature just calculated, the system cofactor h, the respondent's public key, and the respondent's ephemeral public key, the Respondent finds the EC point P = h x S_r x (Q_4 + (avf(Q_4) x Q_3)).
   2e. Respondent verifies that P != 0.
   2f. Respondent sets z = x_P, where x_P is the x-coordinate of P.
3. Respondent converts z to bit string Z, using the convention specified in X9.63-2001 section 4.3.3.
4. Respondent uses the key derivation function described in X9.63-2001 section 5.6.3 with the hash function given in table MQV_PARAMS to derive the keying data; the length of the key to be generated is also given in MQV_PARAMS.

6. Additional MIKEY payloads

6.1 ECCPT payload format

The ECCPT payload provides for the transport of an EC point in the MIKEY-MQV and in the MIKEY-RSA exchange when ECIES is in use. It is of the form:



Point length is the length of the point data in *bits*.

The point_data field is padded to end on a 32-bit boundary, and is encoded as per ANSI X9.63-2001 4.3.6. Uncompressed format MUST be supported. Hybrid and compressed formats MAY be supported.

7. Multicast applications of MQV and ECIES

Both MQV and ECDSA/ECIES may be used in multicast environments to establish a group TEK, in the same fashion as MIKEY-RSA.

8. Recommended and optional domain parameter sets

8.1 Available domain parameter sets

Elliptic curve domain parameter sets available for use in this protocol for all methods employing EC primitives--and given here--are the fifteen groups that NIST recommends in FIPS 186-2 [FIPS-186-2]. Detailed descriptions of the ECC groups recommended here for MIKEY are not given in this document but can be found elsewhere: All fifteen groups are detailed in each of FIPS 186-2 [FIPS-186-2] and SEC 2 [SEC-2]. The elliptic curve domain parameters are uniquely identified in this document using the ASN.1 object identifiers provided in ANSI X9.63 [X9.63], which are also given in SEC2 [SEC-2].

The fifteen groups proposed in this document use elliptic curves over GF[2^N] with N prime or over GF[P] with P prime. Six of the groups proposed here have been assigned identifiers by IANA [IANA] and the remaining nine curves recommended by NIST might later be assigned identifiers by IANA. See also the 'IANA considerations' section.

IANA	Group Descriptions		X9.63 (and SEC 2) OID
NA	ECPRGF192Random	group P-192	secp192r1
NA	EC2NGF163Random	group B-163	sect163r2
7	EC2NGF163Koblitz	group K-163	sect163k1
NA	ECPRGF224Random	group P-224	secp224r1
NA	EC2NGF233Random	group B-233	sect233r1
NA	EC2NGF233Koblitz	group K-233	sect233k1
NA	ECPRGF256Random	group P-256	secp256r1
8	EC2NGF283Random	group B-283	sect283r1
9	EC2NGF283Koblitz	group K-283	sect283k1
NA	ECPRGF384Random	group P-384	secp384r1
10	EC2NGF409Random	group B-409	sect409r1
11	EC2NGF409Koblitz	group K-409	sect409k1
NA	ECPRGF521Random	group P-521	secp521r1
12	EC2NGF571Random	group B-571	sect571r1
13	EC2NGF571Koblitz	group K-571	sect571k1

Three curves are defined at each strength - two curves chosen verifiably at random (as defined in ANSI [X9.62]), one over a binary field and another over a prime field, and a Koblitz curve over a binary field that, which enables especially efficient implementations due to the special structure of the curve [Kob, NSA].

Note that the large number of proposed curves is for two reasons: Flexibility in implementation in using groups over prime fields (GF[p]) or binary fields (GF[2^N]), which have different characteristics; and to provide higher security strength capabilities for military-grade or future uses. In ECDH, The 163-bit and 192-bit curves provide equivalent security strength to Oakley group 2; all other proposed curves offer significantly higher security strength equivalents than the three Diffie-Hellman groups included in [MIKEY].

8.1 Recommended domain parameter sets

It is RECOMMENDED that, for minimum interoperability, all implementations except those in highly constrained environments support use of the P-256 curve.

It is RECOMMENDED that implementations in constrained environments support the K163 curve.

9. Security Considerations

Since this document proposes new methods for use within MIKEY, many of the security considerations contained within RFC 3830 apply here as well.

Some of the methods proposed in this document offer higher cryptographic strength than those proposed in RFC 3830. 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 [DSS], by the SECG in SEC2 [SEC2], and by ANSI in ANSI X9.62 [X9.62] and X9.63 [X9.63].

Proper validation of elliptic curve public keys can help prevent the attacks described in [BMM].

10. IANA Considerations

This specification requires additional parameter sets be defined for use in MIKEY when elliptic curve cryptographic methods are used. These are listed in section 8.1.

It is requested that these be added to the namespace for the DH-Group field in table 6.4 of RFC 3830, which that document requests shall be managed by the IANA.

12. References

[IANA] Internet Assigned Numbers Authority. Attribute Assigned Numbers. (http://www.isi.edu/in-notes/iana/assignments/ipsec-registry)

[IEEE 1363A-2004] Institute of Electrical and Electronics Engineers, IEEE P1363a, Draft Standard Specifications for Public Key Cryptography - Amendment 1: Additional Techniques.

[KOB] N. Koblitz, CM curves with good cryptographic properties. Proceedings of Crypto '91. Pages 279-287. Springer-Verlag, 1992.

[FIPS-186-2] U.S. Department of Commerce/National Institute of Standards and Technology. Digital Signature Standard (DSS), FIPS PUB 186-2, January 2000. (http://csrc.nist.gov/fips/fips186-2.pdf)

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

[LEN] A. Lenstra and E. Verhuel, Selecting cryptographic key sizes. Available at: www.cryptosavvy.com.

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

[NSA] J. Solinas, An improved algorithm for arithmetic on a family of elliptic curves, Proceedings of Crypto '97, Pages 357-371, Springer-Verlag, 1997.

[RFC-3278] S. Blake-Wilson, D. Brown and P. Lambert, The Use of Elliptic Curve Cryptography (ECC) Algorithms in the Cryptographic Message Syntax (CMS), RFC 3279, April 2002.

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

[SEC2] Standards for Efficient Cryptography Group. SEC 2 - Recommended Elliptic Curve Domain Parameters. Working Draft Ver. 1.0., 2000. (http://www.secg.org)

[X9.62] American National Standards Institute, ANSI X9.62-1998: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm. January 1999.

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

[HANKERSON] Hankerson, Darrel et al. "Guide to Elliptic Curve Cryptography". Springer-Verlag, 2004.

Authors' Addresses

Andrew Milne
Certicom Corp.
amilne@certicom.com

Mitch Blaser
Certicom Corp.
mblaser@certicom.com

Daniel R. L. Brown
Certicom Corp.
dbrown@certicom.com

Lakshminath Dondeti
Qualcomm, Inc.
ldondeti@qualcomm.com

Expiry reminder

This draft expires December 1, 2005.