

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 13, 2007

D. Brown
E. Chin
C. Tse
Certicom Corp.
June 11, 2007

ECC Algorithms for MIKEY
draft-ietf-msec-mikey-ecc-03

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 13, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

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	3
2.	MIKEY-DHSIGN with ECDSA or ECGDSA	5
3.	MIKEY-DHSIGN with ECDH	6
4.	MIKEY-ECIES	8
5.	MIKEY-ECMQV	9
6.	Additional Payload Encoding	11
6.1.	ECC Point payload (ECCPT)	11
7.	Security Considerations	13
8.	IANA Considerations	14
9.	References	15
9.1.	Normative References	15
9.2.	Informative References	16
	Authors' Addresses	17
	Intellectual Property and Copyright Statements	18

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). The ECIES method (MIKEY-ECIES) is similar to MIKEY-RSA method, and the ECMQV method (MIKEY-ECMQV) is similar to MIKEY-DHSIGN method.

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

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

Brown, et al.

Expires December 13, 2007

[Page 3]

Internet-Draft

ECC Algorithms for MIKEY

June 2007

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 SIGN_i, a signature covering the Initiator's message. As well, the Responder's message includes SIGN_r, 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 are 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 are 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](#).

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},	
KEMAC, [CHASH], PKE, 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. In general, the MIKEY-ECIES and MIKEY-RSA methods are exactly the same, except that the supported signature algorithm and the public key encryption algorithm are different.

The signature algorithm applied is defined by, and dependent on the certificate used. The MIKEY-ECIES method supports ECDSA as described in [\[ANSI-X9.62\]](#) and ECGDSA as described in [\[ISO-IEC-15946-2\]](#). The SIGNi will use either ECDSA or ECGDSA as a signature algorithm, as described in [Section 2](#).

The public key encryption algorithm applied is defined by, and dependent on the certificate used. The MIKEY-ECIES method supports ECIES as 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 defined in ANSI X9.63 [[ANSI-X9.63](#)]. ECMQV provides 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 2-pass protocol.

Initiator

Responder

I_MESSAGE =

HDR, T, RAND, [IDi|CERTi], [IDr],
 {SP}, ECCPTi, SIGNi

--->

R_MESSAGE =

[<---] HDR, T, [IDr|CERTr],
 IDi, ECCPTr, ECCPTi, V

The MIKEY-ECMQV method is similar to the MIKEY-DHSIGN method, except that with MIKEY-ECMQV, a variable-length shared secret is created using ECMQV instead of a fixed-length shared secret. Same as the MIKEY-DHSIGN method, this method cannot be used to create group keys; it can only be used to create single peer-to-peer keys.

The MIKEY-ECMQV method create a variable-length shared secret. From this shared secret, the TGK and the auth_key for the Responder's verification message are derived. The first portion of the shared secret is the TGK, and the second portion of the shared secret is the auth_key. The length of TGK is specified in ECCPT payload by the Initiator. The length of auth_key is derived from the authentication algorithm, which is also specified in ECCPT payload by the Initiator.

The main objective of the Initiator's message is to provide the Responder with its ephemeral public key represented by the elliptic curve point (ECCPTi), and a set of security protocol parameters. These parameters include the authentication algorithm for the Responder's verification message, the length of TGK, and the key validity information.

The SIGNi is a signature covering the Initiator's message using the Initiator's signature key from the Initiator's certificate. The signature algorithm applied is defined by, and dependent on the certificate used. The MIKEY-ECMQV method supports ECDSA as described in [[ANSI-X9.62](#)] and ECGDSA as described in [[ISO-IEC-15946-2](#)]. The SIGNi will use either ECDSA or ECGDSA as a signature algorithm, as described in [Section 2](#).

The main objective of the Responder's message is to provide the

Initiator with its ephemeral public key represented by the elliptic curve point (ECCPTr). The set of security protocol parameters are the same as the one in the Initiator's message.

If the Initiator's message is authenticated and accepted by the Responder, and the ECMQV shared secret is created successfully, then the verification message (V) is created using the auth_key. V is calculated in the same way as in the MIKEY-PSA method (see [Section 5.2 of \[RFC3830\]](#)).

If the Responder does not support the set of parameters suggested by the Initiator, the error message SHOULD include the supported parameters (see Section 5.1.1 of [\[ANSI-X9.63\]](#)).

The error message is formed as:

HDR, T, {ERR}, {SP}, [SIGNr]

In case of error, the ECMQV shared secret should not be computed. Without the shared secret, V cannot be generated. As a result, the error message should include a SIGNr instead of V, in the cases when the Responder is able to authenticate the Initiator's message.

The SIGNr is a signature covering the Responder's error message using the Responder's signature key from the Responder's certificate. The signature algorithm applied is defined by, and dependent on the certificate used. The MIKEY-ECMQV method support ECDSA as described in [\[ANSI-X9.62\]](#) and ECGDSA as described in [\[ISO-IEC-15946-2\]](#). The SIGNi will use either ECDSA or ECGDSA as a signature algorithm, as described in [Section 2](#).

2-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-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 ! ECC Curve   ! ECC Point                               ~
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
! Auth alg     ! TGK len     ! Reserv! KV   !
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
! KV data (optional)                                               ~
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

* Next payload (8 bits): identifies the payload that is added after this payload. See [Section 6.1 of \[RFC3830\]](#) for values.

* ECC curve (8 bits): identifies the ECC curve used.

ECC curve	Value
ECPRGF192Random / P-192 / secp192r1	0
EC2NGF163Random / B-163 / sect163r2	1
EC2NGF163Koblitz / K-163 / sect163k1	2
EC2NGF163Random2 / none / sect163r1	3
ECPRGF224Random / P-224 / secp224r1	4
EC2NGF233Random / B-233 / sect233r1	5
EC2NGF233Koblitz / K-233 / sect233k1	6
ECPRGF256Random / P-256 / secp256r1	7

EC2NGF283Random	/	B-283	/	sect283r1		8
EC2NGF283Koblitz	/	K-283	/	sect283k1		9
ECPRGF384Random	/	P-384	/	secp384r1		10
EC2NGF409Random	/	B-409	/	sect409r1		11
EC2NGF409Koblitz	/	K-409	/	sect409k1		12
ECPRGF521Random	/	P-521	/	secp521r1		13
EC2NGF571Random	/	B-571	/	sect571r1		14
EC2NGF571Koblitz	/	K-571	/	sect571k1		15

- * ECC point (variable length): ECC 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.
- * Auth alg (8 bits): specifies the MAC algorithm used for the verification message. See [Section 6.2 of \[RFC3830\]](#) for defined

values.

- * TGK len (16 bits): the length of the TGK (in bytes).
- * KV (4 bits): indicates the type of key validity period specified. This may be done by using an SPI (alternatively an MKI in SRTP) or by providing an interval in which the key is valid (e.g., in the latter case, for SRTP this will be the index range where the key is valid). See [Section 6.13 of \[RFC3830\]](#) for pre-defined values.
- * KV data (variable length): This includes either the SPI/MKI or an interval (see [Section 6.14 of \[RFC3830\]](#)). If KV is NULL, this field is not included.

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), [Section 6.1](#) (ECCPT payload identifier), and [Section 6.1](#) (ECC curve).

[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 Cryptography Group, "Elliptic Curve Cryptography", September 2000.
- [SEC2] Standards for Efficient Cryptography 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.

Internet-Draft

ECC Algorithms for MIKEY

June 2007

Authors' Addresses

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>

Chi Chiu Tse
Certicom Corp.
5520 Explorer Drive
Mississauga, Ontario L4W 5L1
CANADA

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

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at

<http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

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