Internet-Draft
Obsoletes: 5008 (if approved)
Intended Status: Informational

Expires: October 27, 2011

R. Housley
Vigil Security
J. Solinas
National Security Agency

April 25, 2011

Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)
draft-housley-rfc5008bis-01

Abstract

This document specifies the conventions for using the United States National Security Agency's Suite B algorithms in Secure/Multipurpose Internet Mail Extensions (S/MIME) as specified in RFC 5751. This document obsoletes RFC 5008.

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on August 14, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Fact Sheet on National Security Agency (NSA) Suite B Cryptography [NSA] states:

> A Cryptographic Interoperability Strategy (CIS) was developed to find ways to increase assured rapid sharing of information both within the U.S. and between the U.S. and her partners through the use of a common suite of public standards, protocols, algorithms and modes referred to as the "Secure Sharing Suite" or S.3. The implementation of CIS will facilitate the development of a broader range of secure cryptographic products which will be available to a wide customer base. The use of selected public cryptographic standards and protocols and Suite B is the core of CIS.
>
> In 2005, NSA announced Suite B Cryptography which built upon the National Policy on the use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information. In addition to the AES algorithm, Suite B includes cryptographic algorithms for key exchanges, digital signatures and hashing. Suite B cryptography has been selected from cryptography that has been approved by NIST for use by the U.S. Government and specified in NIST standards or recommendations.

This document specifies the conventions for using the United States National Security Agency's Suite B algorithms [NSA] in Secure/Multipurpose Internet Mail Extensions (S/MIME) [MSG]. S/MIME makes use of the Cryptographic Message Syntax (CMS) [CMS]. In particular, the signed-data and the enveloped-data content types are used. This document only addresses Suite B compliance for S/MIME. Other applications of CMS are outside the scope of this document.

Since many of the Suite B algorithms enjoy uses in other environments as well, the majority of the conventions needed for the Suite B algorithms are already specified in other documents. This document references the source of these conventions, with some relevant

details repeated to aid developers that choose to support Suite B.

This specification obsoletes RFC 5008 [SUITEBSMIME]. The primary reason for the publication of this document is to allow greater flexibility in the use of the Suite B Security Levels as discussed in Section 1.3. It also removes some duplication between this document and referenced RFCs.

1.1. Terminology

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 RFC 2119 [STDWORDS].

1.2. ASN.1

CMS values are generated using ASN.1 [X.208-88], the Basic Encoding Rules (BER) [X.209-88], and the Distinguished Encoding Rules (DER) [X.509-88].

1.3. Suite B Security Levels

Suite B offers two suites of algorithms for key agreement, key derivation, key wrap and content encryption and two possible combinations of hash and signing algorithm. Suite B algorithms are defined to support two minimum levels of cryptographic security: 128 and 192 bits.

For S/MIME signed messages, Suite B follows the direction set by RFC 5753 [CMSECC] and RFC 5754 [SHA2]. Suite B uses these combinations of message digest (hash) and signature functions (Sig Sets):

	Sig Set 1	Sig Set 2
Message Digest:	SHA-256	SHA-384
Signature:	ECDSA with P-256	ECDSA with P-384

For S/MIME encrypted messages, Suite B follows the direction set by RFC 5753 [CMSECC] and follows the conventions set by RFC 3565 [CMSAES].

Suite B uses these key establishment (KE) algorithms (KE Sets):

	KE Set 1	KE Set 2
Key Agreement:	ECDH with P-256	ECDH with P-384
Key Derivation:	SHA-256	SHA-384
Key Wrap:	AES-128 Key Wrap	AES-256 Key Wrap
Content Encryption:	AES-128 CBC	AES-256 CBC

The two elliptic curves used in Suite B are specified in [DSS] and each appear in the literature under two different names. For sake of clarity, we list both names below:

Curve	NIST Name	SECG Name	OID [DSS]
nistp256	P-256	secp256r1	1.2.840.10045.3.1.7
nistp384	P-384	secp384r1	1.3.132.0.34

If configured at a minimum level of security of 128 bits, a Suite B compliant S/MIME system performing encryption MUST use either KE Set 1 or KE Set 2 with KE Set 1 the preferred suite. A digital signature, if applied, MUST use either Sig Set 1 or Sig Set 2, independent of the encryption choice.

A recipient in an S/MIME system configured at a minimum level of security of 128 bits MUST be able to verify digital signatures from Sig Set 1 and SHOULD be able to verify digital signatures from Sig Set 2.

Note that for S/MIME systems configured at a minimum level of security of 128 bits the algorithm set used for a signed-data content type is independent of the algorithm set used for an enveloped-data content type.

If configured at a minimum level of security of 192 bits, a Suite B

compliant S/MIME system performing encryption MUST use KE Set 2. A digital signature, if applied, MUST use Sig Set 2.

A recipient in an S/MIME system configured at a minimum level of security of 192 bits MUST be able to verify digital signatures from Sig Set 2.

2. SHA-256 and SHA-384 Message Digest Algorithms

SHA-256 and SHA-384 are the Suite B message digest algorithms. RFC 5754 [SHA2] specifies the conventions for using SHA-256 and SHA-384 with the Cryptographic Message Syntax (CMS). Suite B compliant S/MIME implementations MUST follow the conventions in RFC 5754. Relevant details are repeated below.

Within the CMS signed-data content type, message digest algorithm identifiers are located in the SignedData digestAlgorithms field and the SignerInfo digestAlgorithm field.

The SHA-256 and SHA-384 message digest algorithms are defined in FIPS Pub 180-3 [SHA2FIPS]. The algorithm identifiers for SHA-256 and SHA-384 are defined in [SHA2] and are repeated here:

```
id-sha256  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistalgorithm(4) hashalgs(2) 1 }
```



```
id-sha384  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistalgorithm(4) hashalgs(2) 2 }
```

For both SHA-256 and SHA-384, the AlgorithmIdentifier parameters field is OPTIONAL, and if present, the parameters field MUST contain a NULL. Implementations MUST accept SHA-256 and SHA-384 AlgorithmIdentifiers with absent parameters. Implementations MUST accept SHA-256 and SHA-384 AlgorithmIdentifiers with NULL parameters. As specified in RFC 5754 [SHA2], implementations MUST generate SHA-256 and SHA-384 AlgorithmIdentifiers with absent parameters.

3. ECDSA Signature Algorithm

In Suite B, public key certificates used to verify S/MIME signatures

MUST be compliant with the Suite B Certificate Profile specified in RFC 5759 [SUITEBCERT].

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the Suite B digital signature algorithm. RFC 5753 [CMSECC] specifies the conventions for using ECDSA with the Cryptographic Message Syntax (CMS). Suite B compliant S/MIME implementations MUST follow the conventions in RFC 5753. Relevant details are repeated below.

Within the CMS signed-data content type, signature algorithm identifiers are located in the SignerInfo signatureAlgorithm field of SignedData. In addition, signature algorithm identifiers are located in the SignerInfo signatureAlgorithm field of countersignature attributes.

RFC 5480 [PKI-ALG] defines the signature algorithm identifiers used in CMS for ECDSA with SHA-256 and ECDSA with SHA-384. The identifiers are repeated here:

```
ecdsa-with-SHA256  OBJECT IDENTIFIER  ::=  { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-sha2(3) 2 }

ecdsa-with-SHA384  OBJECT IDENTIFIER  ::=  { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-sha2(3) 3 }
```

When either the ecdsa-with-SHA256 or the ecdsa-with-SHA384 algorithm identifier is used, the AlgorithmIdentifier parameters field MUST be absent.

When signing, the ECDSA algorithm generates two values, commonly called r and s. To transfer these two values as one signature,

they MUST be encoded using the ECDSA-Sig-Value type specified in RFC 5480 [PKI-ALG]:

```
ECDSA-Sig-Value  ::=  SEQUENCE {
   r  INTEGER,
   s  INTEGER }
```

4. Key Management

CMS accommodates the following general key management techniques: key agreement, key transport, previously distributed symmetric key-encryption keys, and passwords. In Suite B for S/MIME, ephemeral-static key agreement MUST be used as described in Section 4.1.

When a key agreement algorithm is used, a key-encryption algorithm is also needed. In Suite B for S/MIME, the Advanced Encryption Standard (AES) Key Wrap, as specified in RFC 3394 [AESWRAP, SH], MUST be used as the key-encryption algorithm. AES Key Wrap is discussed further in Section 4.2. The key-encryption key used with the AES Key Wrap algorithm is obtained from a key derivation function (KDF). In Suite B for S/MIME, there are two KDFs, one based on SHA-256 and one based on SHA-384. These KDFs are discussed further in Section 4.3.

4.1. ECDH Key Agreement Algorithm

Elliptic Curve Diffie-Hellman (ECDH) is the Suite B key agreement algorithm.

S/MIME is used in store-and-forward communications, which means that ephemeral-static ECDH is always employed. This means that the message originator possesses an ephemeral ECDH key pair and that the message recipient possesses a static ECDH key pair whose public key is represented by an X.509 certificate. In Suite B, the certificate used to obtain the recipient's public key MUST be compliant with the Suite B Certificate Profile specified in RFC 5759 [SUITEBCERT].

Section 3.1 of RFC 5753 [CMSECC] specifies the conventions for using ECDH with the CMS. Suite B compliant S/MIME implementations MUST follow these conventions. Relevant details are repeated below.

Within the CMS enveloped-data content type, key agreement algorithm identifiers are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm field.

keyEncryptionAlgorithm MUST be one of the two algorithm identifiers listed below, and the algorithm identifier parameter field MUST be present and identify the key wrap algorithm. The key wrap algorithm

denotes the symmetric encryption algorithm used to encrypt the

content-encryption key with the pairwise key-encryption key generated using the ephemeral-static ECDH key agreement algorithm (see Section 4.3).

When implementing KE Set 1, the keyEncryptionAlgorithm MUST be dhSinglePass-stdDH-sha256kdf-scheme and the keyEncryptionAlgorithm parameter MUST be a KeyWrapAlgorithm containing id-aes128-wrap (see Section 4.2). When implementing KE Set 2, the keyEncryptionAlgorithm MUST be dhSinglePass-stdDH-sha384kdf-scheme and the keyEncryptionAlgorithm parameter MUST be a KeyWrapAlgorithm containing id-aes256-wrap.

The algorithm identifiers for dhSinglePass-stdDH-sha256kdf-scheme and dhSinglePass-stdDH-sha384kdf-scheme, repeated from [CMSECC] Section 7.1.4, are:

```
dhSinglePass-stdDH-sha256kdf-scheme  OBJECT IDENTIFIER  ::=
    { iso(1) identified-organization(3) certicom(132)
      schemes(1) 11 1 }

dhSinglePass-stdDH-sha384kdf-scheme  OBJECT IDENTIFIER  ::=
    { iso(1) identified-organization(3) certicom(132)
      schemes(1) 11 2 }
```

Both of these algorithm identifiers use KeyWrapAlgorithm as the type for their parameter:

```
KeyWrapAlgorithm  ::=  AlgorithmIdentifier
```

4.2. AES Key Wrap

The AES Key Wrap key-encryption algorithm, as specified in RFC 3394 [AESWRAP, SH], is used to encrypt the content-encryption key with a pairwise key-encryption key that is generated using ephemeral-static ECDH. Section 8 of RFC 5753 [CMSECC] specifies the conventions for using AES Key Wrap with the pairwise key generated with epheneral-static ECDH with the CMS. Suite B compliant S/MIME implementations MUST follow these conventions. Relevant details are repeated below.

When implementing KE Set 1, the KeyWrapAlgorithm MUST be id-aes128-wrap. When implementing KE Set 2, the KeyWrapAlgorithm MUST be id-aes256-wrap.

Within the CMS enveloped-data content type, key wrap algorithm identifiers are located in the KeyWrapAlgorithm parameters within the EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm field.

The algorithm identifiers for AES Key Wrap are specified in RFC 3394 [SH], and the ones needed for Suite B compliant S/MIME implementations are repeated here:

```
id-aes128-wrap  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   nistAlgorithm(4) aes(1) 5 }

id-aes256-wrap  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   nistAlgorithm(4) aes(1) 45 }
```

4.3. Key Derivation Functions

KDFs based on SHA-256 and SHA-384 are used to derive a pairwise key-encryption key from the shared secret produced by ephemeral-static ECDH. Sections 7.1.8 and 7.2 of RFC 5753 [CMSECC] specify the conventions for using the KDF with the shared secret generated with ephemeral-static ECDH with the CMS. Suite B compliant S/MIME implementations MUST follow these conventions. Relevant details are repeated below.

When implementing KE Set 1, the KDF based on SHA-256 MUST be used. When implementing KE Set 2, the KDF based on SHA-384 MUST be used.

As specified in Section 7.2 of RFC 5753 [CMSECC], using ECDH with the CMS enveloped-data content type, the derivation of key-encryption keys makes use of the ECC-CMS-SharedInfo type, which is repeated here:

```
ECC-CMS-SharedInfo  ::=  SEQUENCE {
   keyInfo      AlgorithmIdentifier,
   entityUInfo  [0] EXPLICIT OCTET STRING OPTIONAL,
   suppPubInfo  [2] EXPLICIT OCTET STRING }
```

In Suite B for S/MIME, the fields of ECC-CMS-SharedInfo are used as follows:

keyInfo contains the object identifier of the key-encryption algorithm used to wrap the content-encryption key. In Suite B for S/MIME, if the AES-128 Key Wrap is used, then the keyInfo will contain id-aes128-wrap and the parameters will be absent. In Suite B for S/MIME, if AES-256 Key Wrap is used, then the keyInfo will contain id-aes256-wrap and the parameters will be absent.

entityUInfo optionally contains a random value provided by the message originator. If the ukm is present, then the

entityUInfo MUST be present, and it MUST contain the ukm value. If the ukm is not present, then the entityUInfo MUST be absent.

suppPubInfo contains the length of the generated key-encryption key, in bits, represented as a 32-bit unsigned number, as described in RFC 2631 [CMSDH]. When a 128-bit AES key is used, the length MUST be 0x00000080. When a 256-bit AES key is used, the length MUST be 0x00000100.

ECC-CMS-SharedInfo is DER-encoded and used as input to the key derivation function, as specified in Section 3.6.1 of [SEC1]. Note that ECC-CMS-SharedInfo differs from the OtherInfo specified in [CMSDH]. Here, a counter value is not included in the keyInfo field because the KDF specified in [SEC1] ensures that sufficient keying data is provided.

The KDF specified in [SEC1] provides an algorithm for generating an essentially arbitrary amount of keying material from the shared secret produced by ephemeral-static ECDH, which is called Z for the remainder of this discussion. The KDF can be summarized as:

KM = Hash (Z || Counter || ECC-CMS-SharedInfo)

To generate a key-encryption key, one or more KM blocks are generated, incrementing Counter appropriately, until enough material has been generated. The KM blocks are concatenated left to right:

KEK = KM (counter=1) || KM (counter=2) ...

The elements of the KDF are used as follows:

Hash is the one-way hash function. If KE Set 1 is used, the SHA-256 hash MUST be used. If KE Set 2 is used, the SHA-384 hash MUST be used.

Z is the shared secret value generated by ephemeral-static ECDH. Leading zero bits MUST be preserved. In Suite B for S/MIME, if KE Set 1 is used, Z MUST be exactly 256 bits. In Suite B for S/MIME, if KE Set 2 is used, Z MUST be exactly 384 bits.

Counter is a 32-bit unsigned number, represented in network byte order. Its initial value MUST be 0x00000001 for any key derivation operation. In Suite B for S/MIME, with both KE Set 1 and KE Set 2, exactly one iteration is needed; the Counter is not incremented.

ECC-CMS-SharedInfo is composed as described above. It MUST be DER encoded.

To generate a key-encryption key, one KM block is generated, with a Counter value of 0x00000001:

KEK = KM (1) = Hash (Z || Counter=1 || ECC-CMS-SharedInfo)

In Suite B for S/MIME, when KE Set 1 is used, the key-encryption key MUST be the most significant 128 bits of the SHA-256 output value. In Suite B for S/MIME, when KE Set 2 is used, the key-encryption key MUST be the most significant 256 bits of the SHA-384 output value.

Note that the only source of secret entropy in this computation is Z. The effective key space of the key-encryption key is limited by the size of Z, in addition to any security level considerations imposed by the elliptic curve that is used. However, if entityUInfo is different for each message, a different key-encryption key will be generated for each message.

5. AES CBC Content Encryption

AES [AES] in Cipher Block Chaining (CBC) mode [MODES] is the Suite B for S/MIME content-encryption algorithm. RFC 3565 [CMSAES] specifies the conventions for using AES with the CMS. Suite B compliant S/MIME implementations MUST follow these conventions. Relevant details are repeated below.

In Suite B for S/MIME, if KE Set 1 is used, AES-128 in CBC mode MUST be used for content encryption. In Suite B for S/MIME, if KE Set 2 is used, AES-256 in CBC mode MUST be used.

Within the CMS enveloped-data content type, content encryption algorithm identifiers are located in the EnvelopedData EncryptedContentInfo contentEncryptionAlgorithm field. The content

encryption algorithm is used to encipher the content located in the EnvelopedData EncryptedContentInfo encryptedContent field.

The AES CBC content-encryption algorithm is described in [AES] and [MODES]. The algorithm identifier for AES-128 in CBC mode is:

```
id-aes128-CBC  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   nistAlgorithm(4) aes(1) 2 }
```

The algorithm identifier for AES-256 in CBC mode is:

```
id-aes256-CBC  OBJECT IDENTIFIER  ::=  { joint-iso-itu-t(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   nistAlgorithm(4) aes(1) 42 }
```


The AlgorithmIdentifier parameters field MUST be present, and the parameters field must contain AES-IV:

```
AES-IV  ::=  OCTET STRING (SIZE(16))
```

The 16-octet initialization vector is generated at random by the originator. See [RANDOM] for guidance on generation of random values.

6. IANA Considerations

This document has no IANA considerations.

{{{ RFC Editor: Please delete this section prior to publication as an RFC. }}}

7. Security Considerations

This document specifies the conventions for using the NSA's Suite B algorithms in S/MIME. All of the algorithms and algorithm identifiers have been specified in previous documents.

Two minimum levels of security may be achieved using this specification. Users must consider their risk environment to determine which level is appropriate for their own use.

See [RANDOM] for guidance on generation of random values.

The security considerations in RFC 5652 [CMS] discuss the CMS as a method for digitally signing data and encrypting data.

The security considerations in RFC 3370 [CMSALG] discuss cryptographic algorithm implementation concerns in the context of the CMS.

The security considerations in RFC 5753 [CMSECC] discuss the use of elliptic curve cryptography (ECC) in the CMS.

The security considerations in RFC 3565 [CMSAES] discuss the use of AES in the CMS.

8. References

8.1. Normative References

[AES] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001.

[AESWRAP] National Institute of Standards and Technology, "AES Key Wrap Specification", 17 November 2001. [http://csrc.nist.gov/encryption/kms/key-wrap.pdf].

[DSS] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS PUB 186-3, June 2009.

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

[CMSAES] Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3565, July 2003.

[CMSALG] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.

[CMSDH] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, June 1999.

[CMSECC] Turner, S., and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 5753, January 2010.

[MODES] National Institute of Standards and Technology, "DES Modes of Operation", FIPS Pub 81, 2 December 1980.

[MSG] Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[PKI-ALG] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.

[SEC1] Standards for Efficient Cryptography Group, "Elliptic Curve Cryptography", 2000. [http://www.secg.org/collateral/sec1.pdf].

[SH] Schaad, J., and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, September 2002.

[SHA2] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, January 2010.

[SHA2FIPS] National Institute of Standards and Technology, "Secure Hash Standard", FIPS 180-3, October 2008.

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

[SUITEBCERT] Solinas, J. and Zieglar, L., "Suite B Certificate and Certificate Revocation List Profile", RFC 5759, January 2010.

[SUITEBSMIME]
Housley, R. and Solinas, J., "Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)", RFC 5008, September 2007.

[X.208-88] CCITT. Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1). 1988.

[X.209-88] CCITT. Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). 1988.

[X.509-88] CCITT. Recommendation X.509: The Directory - Authentication Framework. 1988.

8.2. Informative References

[RANDOM] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[NSA] U.S. National Security Agency, "Fact Sheet NSA Suite B Cryptography", January 2009. [http://www.nsa.gov/ia/programs/suiteb_cryptography]

Authors' Addresses

Russell Housley
Vigil Security, LLC

918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: housley@vigilsec.com

Jerome A. Solinas
National Information Assurance Laboratory
National Security Agency
9800 Savage Road
Fort George G. Meade, MD 20755
USA

EMail: jasolin@orion.ncsc.mil