Internet-Draft R. Housley
Intended status: Standards Track Vigil Security
Expires: 15 October 2016 15 April 2016

Use of the Elliptic Curve Diffie-Hellamn Key Agreement Algorithm with Curve 25519 and Curve 448 in the Cryptographic Message Syntax (CMS)

<draft-housley-cms-ecdh-new-curves-00.txt>

Abstract

This document describes the conventions for using Elliptic Curve Diffie-Hellamn (ECDH) key agreement algorithm using Curve 25519 and Curve 448 in the Cryptographic Message Syntax (CMS).

Status of This Memo

Copyright Notice

1. Introduction

This document describes the conventions for using Elliptic Curve Diffie-Hellamn (ECDH) key agreement using Curve 25519 and Curve 448 [CURVE] in the Cryptographic Message Syntax (CMS) [CMS]. Key agreement is supported in three CMS content types: the enveloped-data content type [CMS], authenticated-data content type [CMS], and the authenticated-enveloped-data content type [AUTHENV].

The conventions for using some Elliptic Curve Cryptography (ECC) algorithms in CMS are described in [CMSECC]. These conventions cover the use of ECDH with some curves other than Curve 25519 and Curve 448 [CURVE]. Those other curves are not deprecated, but support for Curve 25519 and Curve 448 is encouraged.

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 [X680], which uses the Basic Encoding Rules (BER) and the Distinguished Encoding Rules (DER) [X690].

2. Key Agreement

In 1976, Diffie and Hellman describe a means for two parties to agree upon a shared secret value in manner that prevents eavesdroppers from learning the shared secret value [DH1976]. This secret may then be converted into pairwise symmetric keying material for use with other cryptographic algorithms. Over the years, many variants of this fundamental technique have been developed. This document describes the conventions for using Ephemeral-Static Elliptic Curve Diffie-Hellamn (ECDH) key agreement using Curve 25519 and Curve 448.

The originator uses an ephemeral public/private key pair that is generated on the same elliptic curve as the public key of the recipient. The ephemeral key pair is used for a single CMS protected content type, and then it is discarded. The originator obtains the recipient's static public key from the recipient's certificate

[PROFILE].

ECDH with Curve 25519 is described in Section 6.1 of [CURVE], and ECDH with Curve 448 is described in Section 6.2 of [CURVE]. Since Curve 25519 and Curve 448 have cofactors of 8 and 4, respectively, an

input point of small order will eliminate any contribution from the other party's private key. As described in Section 7 of [CURVE], implementations MAY detect this situation by checking for the all-zero output.

In [CURVE], the shared secret value that is produced by ECDH is called K. A key derivation function (KDF) is used to produce a pairwise key-encryption key from K, the length of the key-encryption key, and the DER-encoded ECC-CMS-SharedInfo structure [CMSECC].

The ECC-CMS-SharedInfo definition from [CMSECC] is repeated here for convenience.

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

The ECC-CMS-SharedInfo keyInfo field contains the object identifier of the key-encryption algorithm and associated parameters. This algorithm will be used to wrap the content-encryption key. In this specification, the AES Key Wrap algorithm identifier has absent parameters.

The ECC-CMS-SharedInfo entityUInfo field optionally contains additional keying material supplied by the sending agent. Note that [CMS] REQUIRES implementations to accept a KeyAgreeRecipientInfo SEQUENCE that includes the ukm field. If the ukm field is present, the ukm is placed in the entityUInfo field. The ukm value need not be longer than the key-encryption key that will be produced by the KDF. When present, the ukm ensures that a different key-encryption key is generated, even when the originator ephemeral private key is improperly used more than once.

The ECC-CMS-SharedInfo suppPubInfo field contains the length of the generated key-encryption key, in bits, represented as a 32-bit number. For example, the key length for AES-256 would be 0x00000100.

2.1. ANSI-X9.63-KDF

The ANSI-X9.63-KDF key derivation function is a simple construct based on a one-way hash function described in ANS X9.63 [X963]. This KDF is also described in Section 3.6.1 of [SEC1].

Three values are concatenated to produce the input string to the KDF:

1. The shared secret value generated by ECDH, K.
2. The length in octets of the keying data to be generated.
3. The DER-encoded ECC-CMS-SharedInfo structure.

To generate a key-encryption key, generates one or more KM blocks, with the counter starting at 0x00000001, and incrementing the counter for each subsequent KM block until enough material has been generated. The KM blocks are concatenated left to right:

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

KM(i) = Hash(K || INT32(counter=i) || DER(ECC-CMS-SharedInfo))
```

The output of the KDF is the pairwise key-encryption key.

2.2. HKDF

{{{ Should we specify a way to use HKDF from RFC 5869? }}}

```
if ukm is provided, then salt = ukm, else salt = zero
PRK = HKDF-Extract(salt, K)

KEK = HKDF-Expand(PRK, DER(ECC-CMS-SharedInfo), SizeOf(KEK))
```

3. Enveloped-data Conventions

The CMS enveloped-data content type [CMS] consists of an encrypted content and wrapped content-encryption keys for one or more recipients. The ECDH key agreement algorithm is used to generate a pairwise key-encryption key between the originator and a particular recipient. Then, the key-encryption key is used to wrap the content-encryption key for that recipient. When there more than one recipient, the same content-encryption key is wrapped for each of them.

A compliant implementation MUST meet the requirements for constructing an enveloped-data content type stated in Section 6 of [CMS].

A content-encryption key MUST be randomly generated for each instance of an enveloped-data content type. The content-encryption key is used to encipher the content.

3.1. EnvelopedData Fields

The enveloped-data content type is ASN.1 encoded using the EnvelopedData syntax. The fields of the EnvelopedData syntax MUST be populated as described in [CMS]; for the recipients that use ECDH with Curve 25519 or Curve 448 the RecipientInfo kari choice MUST be used.

3.2. KeyAgreeRecipientInfo Fields

The fields of the KeyAgreeRecipientInfo syntax MUST be populated as described in this section when ECDH with Curve 25519 or Curve 448 is employed for one or more recipients.

The KeyAgreeRecipientInfo version MUST be 3.

The KeyAgreeRecipientInfo originator provides three alternatives for identifying the originator's public key, and the originatorKey alternative MUST be used. The originatorKey MUST contain an ephemeral key for the originator. The originatorKey algorithm field MUST contain the id-ecPublicKey object identifier along with ECParameters as specified in [PKIXECC]. The originator's ephemeral public key MUST be encoded using the type ECPoint as specified in [CMSECC]. As a courtesy, the definitions are repeated here:

```
id-ecPublicKey OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) ansi-X9-62(10045) keyType(2) 1 }

ECPoint ::= OCTET STRING
```

```
ECParameters ::= CHOICE {
  namedCurve         OBJECT IDENTIFIER
  -- implicitCurve   NULL
  -- specifiedCurve  SpecifiedECDomain -- }
```

The object identifiers for Curve 25519 and Curve 448 have been assigned in [ID.josefsson-pkix-newcurves]. They are repeated below for convenience.

When using Curve 25519, the ECPoint contains exactly 32 octets, and the ECParameters namedCurve MUST contain the following object identifier:

```
id-Curve25519 OBJECT IDENTIFIER ::= { 1 3 6 1 4 1 11591 15 1 }
```

When using Curve 448, the ECPoint contains exactly 56 octets, and the ECParameters namedCurve MUST contain the following object identifier:

```
id-Curve448 OBJECT IDENTIFIER ::= { 1 3 6 1 4 1 11591 15 2 }
```

KeyAgreeRecipientInfo ukm is optional. Note that [CMS] REQUIRES implementations to accept a KeyAgreeRecipientInfo SEQUENCE that includes the ukm field. If present, the ukm is placed in the entityUInfo field of the ECC-CMS-SharedInfo as input to the KDF. The ukm value need not be longer than the key-encryption key produced by the KDF.

KeyAgreeRecipientInfo keyEncryptionAlgorithm MUST contain the object identifier of the key-encryption algorithm that will be used to wrap the content-encryption key. The conventions for using AES-128, AES-192, and AES-256 in the key wrap mode are specified in [CMSAES].

KeyAgreeRecipientInfo recipientEncryptedKeys includes a recipient identifier and encrypted key for one or more recipients. The RecipientEncryptedKey KeyAgreeRecipientIdentifier MUST contain either the issuerAndSerialNumber identifying the recipient's certificate or the RecipientKeyIdentifier containing the subject key identifier from the recipient's certificate. In both cases, the recipient's certificate contains the recipient's static ECDH public key with Curve 25519 or Curve 448 public key. RecipientEncryptedKey EncryptedKey MUST contain the content-encryption key encrypted with the pairwise key-encryption key using the algorithm specified by the

KeyWrapAlgorithm.

4. Authenticated-data Conventions

The CMS authenticated-data content type [CMS] consists an authenticated content, a message authentication code (MAC), and encrypted authentication keys for one or more recipients. The ECDH key agreement algorithm is used to generate a pairwise key-encryption key between the originator and a particular recipient. Then, the key-encryption key is used to wrap the authentication key for that recipient. When there more than one recipient, the same authentication key is wrapped for each of them.

A compliant implementation MUST meet the requirements for constructing an authenticated-data content type stated in Section 9 of [CMS].

A authentication key MUST be randomly generated for each instance of an authenticated-data content type. The authentication key is used to compute the MAC over the content.

4.1. AuthenticatedData Fields

The authenticated-data content type is ASN.1 encoded using the AuthenticatedData syntax. The fields of the AuthenticatedData syntax MUST be populated as described in [CMS]; for the recipients that use ECDH with Curve 25519 or Curve 448 the RecipientInfo kari choice MUST be used.

4.2. KeyAgreeRecipientInfo Fields

The fields of the KeyAgreeRecipientInfo syntax MUST be populated as described in Section 3.2 of this document.

5. Authenticated-Enveloped-data Conventions

The CMS authenticated-enveloped-data content type content type [AUTHENV] consists of an authenticated and encrypted content and encrypted content-authenticated-encryption keys for one or more recipients. The ECDH key agreement algorithm is used to generate a pairwise key-encryption key between the originator and a particular recipient. Then, the key-encryption key is used to wrap the content-authenticated-encryption key for that recipient. When there more than one recipient, the same content-authenticated-encryption key is wrapped for each of them.

A compliant implementation MUST meet the requirements for constructing an authenticated-data content type stated in Section 2 of [AUTHENV].

A content-authenticated-encryption key MUST be randomly generated for each instance of an authenticated-enveloped-data content type. The content-authenticated-encryption key key is used to authenticate and encrypt the content.

5.1. AuthEnvelopedData Fields

The authenticated-enveloped-data content type is ASN.1 encoded using the AuthEnvelopedData syntax. The fields of the AuthEnvelopedData syntax MUST be populated as described in [AUTHENV]; for the recipients that use ECDH with Curve 25519 or Curve 448 the RecipientInfo kari choice MUST be used.

5.2. KeyAgreeRecipientInfo Fields

The fields of the KeyAgreeRecipientInfo syntax MUST be populated as described in Section 3.2 of this document.

6. Certificate Conventions

RFC 5280 [PROFILE] specifies the profile for using X.509 Certificates in Internet applications. A recipient static public key is needed for ECDH with Curve 25519 or Curve 448, and the originator obtains that public key from the recipient's certificate. The conventions in this section augment RFC 5280.

The id-ecPublicKey object identifier continues to identify the static ECDH public key for the recipient. The associated EcpkParameters parameters structure is specified in [PKIXALG], and the namedCurve alternative MUST be used. The object identifiers from Section 3.2 of this document are used for Curve 25519 and Curve 448. The EcpkParameters parameters structure is repeated here for convenience:


```
EcpkParameters ::= CHOICE {
```

```
      ecParameters  ECParameters,
      namedCurve    OBJECT IDENTIFIER,
      implicitlyCA  NULL }
```

The certificate issuer MAY use indicate the intended usage for the certified public key by including the key usage certificate extension as specified in Section 4.2.1.3 of [PROFILE]. If the keyUsage extension is present in a certificate that conveys an ECDH static public key, then the key usage extension MUST set the keyAgreement bit.

7. SMIMECapabilities Attribute Conventions

A sending agent MAY announce to other agents that it supports ECDH key agreement using the SMIMECapabilities signed attribute in a signed message [SMIME] or a certificate [CERTCAP]. Following the pattern established in [CMSECC], the SMIMECapabilities associated with ECDH carries a DER-encoded object identifier that identifies support for ECDH in conjunction with a particular KDF, and it includes a parameter that names the key wrap algorithm.

The following SMIMECapabilities values (in hexidecimal) from [CMSECC] might be of interest to implementations that support Curve 25519 and Curve 448:

```
ECDH with SHA-256 as the KDF; uses AES-128 key wrap:
   30 15 06 06 2B 81 04 01 0B 01 30 0B 06 09 60 86 48 01 65 03 04
   01 05

ECDH with SHA-384 as the KDF; uses AES-128 key wrap:
   30 15 06 06 2B 81 04 01 0B 02 30 0B 06 09 60 86 48 01 65 03 04
   01 05

ECDH with SHA-512 as the KDF; uses AES-128 key wrap:
   30 15 06 06 2B 81 04 01 0B 03 30 0B 06 09 60 86 48 01 65 03 04
   01 05

ECDH with SHA-256 as the KDF; uses AES-256 key wrap:
   30 15 06 06 2B 81 04 01 0B 01 30 0B 06 09 60 86 48 01 65 03 04
   01 2D

ECDH with SHA-384 as the KDF; uses AES-256 key wrap:
   30 15 06 06 2B 81 04 01 0B 02 30 0B 06 09 60 86 48 01 65 03 04
   01 2D
```

      ECDH with SHA-512 as the KDF; uses AES-256 key wrap:
         30 15 06 06 2B 81 04 01 0B 03 30 0B 06 09 60 86 48 01 65 03 04
         01 2D

{{{ Should we specify a way to use HKDF from RFC 5869 with ECDH? }}}

8. Security Considerations

Please consult the security considerations of [CMS] and [AUTHENV] for security considerations related to the enveloped-data content type and the authenticated-enveloped-data content type, respectively.

Please consult the security considerations of [CURVES] for security considerations related to the use of ECDH with Curve 25519 and Curve 448.

The originator uses an ephemeral public/private key pair that is generated on the same elliptic curve as the public key of the recipient. The ephemeral key pair is used for a single CMS protected content type, and then it is discarded. If the originator wants to be able to decrypt the content (for enveloped-data and authenticated-enveloped-data) or check the authentication (for authenticated-data), then the originator needs to treat themselves as a recipient.

As specified in [CMS], implementations MUST support processing of the KeyAgreeRecipientInfo ukm field, so interoperability is not a concern if the ukm is present or absent. The ukm is placed in the entityUInfo field of the ECC-CMS-SharedInfo structure. When present, the ukm ensures that a different key-encryption key is generated, even when the originator ephemeral private key is improperly used more than once.

9. IANA Considerations

No IANA registrations are requested in this document.

10. Normative References

[AUTHENV] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, November 2007.

[CERTCAP] Santesson, S., "X.509 Certificate Extension for Secure/Multipurpose Internet Mail Extensions (S/MIME) Capabilities", RFC 4262, December 2005.

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

[CURVES] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, January 2016.

[ID.josefsson-pkix-newcurves] Josefsson, S., "Using Curve25519 and Curve448 in PKIX", 12 October 2015, Work-in-progress.

[PKIXALG] 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.

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

[PROFILE] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[SEC1] Standards for Efficient Cryptography Group, "SEC 1: Elliptic Curve Cryptography", version 2.0, May 2009, <http://www.secg.org/sec1-v2.pdf>.

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

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

[X680] ITU-T, "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, 2002.

[X690] ITU-T, "Information technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, 2002.

11. Informative References

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

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

[DH1976] Diffie, W., and M. E. Hellman, "New Directions in Cryptography", IEEE Trans. on Info. Theory, Vol. IT-22, Nov. 1976, pp. 644-654.

[X963] "Public-Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography", American National Standard X9.63-2001, 2001.

12. Acknowledgements

Thanks to Jim Schaad and Stefan Santesson for their insightful suggestions.

Author Address

Russ Housley
918 Spring Knoll Drive
Herndon, VA 20170
USA
housley@vigilsec.com