Internet Engineering Task Force (IETF)
Request for Comments: 6160
Category: Standards Track
ISSN: 2070-1721

S. Turner
IECA

April 2011

Algorithms for Cryptographic Message Syntax (CMS) Protection of Symmetric Key Package Content Types

Abstract

This document describes the conventions for using several cryptographic algorithms with the Cryptographic Message Syntax (CMS) to protect the symmetric key package content type. Specifically, it includes conventions necessary to implement SignedData, EnvelopedData, EncryptedData, and AuthEnvelopedData.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6160.

1. Introduction

This document describes the conventions for using several cryptographic algorithms with the Cryptographic Message Syntax (CMS) [RFC5652] to protect the symmetric key package content type defined in [RFC6031]. Specifically, it includes conventions necessary to implement the following CMS content types: SignedData [RFC5652], EnvelopedData [RFC5652], EncryptedData [RFC5652], and AuthEnvelopedData [RFC5083]. Familiarity with [RFC5083], [RFC5652], [RFC5753], and [RFC6031] is assumed.

This document does not define any new algorithms; instead, it refers to previously defined algorithms.

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 [RFC2119].

2. SignedData

If an implementation supports SignedData, then it MUST support the signature scheme RSA [RFC3370] and SHOULD support the signature schemes RSA Probabilistic Signature Scheme (RSASSA-PSS) [RFC4056] and Digital Signature Algorithm (DSA) [RFC3370]. Additionally, implementations MUST support the hash function SHA-256 [RFC5754] in concert with these signature schemes, and they SHOULD support the hash function SHA-1 [RFC3370]. If an implementation supports SignedData, then it MAY support Elliptic Curve Digital Signature Algorithm (ECDSA) [RFC6090][RFC5753].

3. EnvelopedData

If an implementation supports EnvelopedData, then it MUST implement key transport, and it MAY implement key agreement.

When key transport is used, RSA encryption [RFC3370] MUST be supported, and RSA Encryption Scheme - Optimal Asymmetric Encryption Padding (RSAES-OAEP) [RFC3560] SHOULD be supported.

When key agreement is used, Diffie-Hellman (DH) ephemeral-static [RFC3370] MUST be supported. When key agreement is used, Elliptic Curve Diffie-Hellman (ECDH) [RFC6090][RFC5753] MAY be supported.

Regardless of the key management technique choice, implementations MUST support AES-128 Key Wrap with Padding [RFC5649] as the content-encryption algorithm. Implementations SHOULD support AES-256 Key Wrap with Padding [RFC5649] as the content-encryption algorithm.

When key agreement is used, the same key-wrap algorithm MUST be used for both key and content encryption. If the content-encryption algorithm is AES-128 Key Wrap with Padding, then the key-wrap algorithm MUST be AES-128 Key Wrap with Padding [RFC5649]. If the content-encryption algorithm is AES-256 Key Wrap with Padding, then the key-wrap algorithm MUST be AES-256 Key Wrap with Padding [RFC5649].

4. EncryptedData

If an implementation supports EncryptedData, then it MUST implement AES-128 Key Wrap with Padding [RFC5649] and SHOULD implement AES-256 Key Wrap with Padding [RFC5649].

NOTE: EncryptedData requires that keys be managed by other means; therefore, the only algorithm specified is the content-encryption algorithm.

5. AuthEnvelopedData

If an implementation supports AuthEnvelopedData, then it MUST implement the EnvelopedData recommendations except for the content-encryption algorithm, which, in this case, MUST be AES-GCM [RFC5084]; the 128-bit version MUST be implemented, and the 256-bit version SHOULD be implemented. Implementations MAY also support AES-CCM [RFC5084].

6. Public Key Sizes

The easiest way to implement SignedData, EnvelopedData, and AuthEnvelopedData is with public key certificates [RFC5280]. If an implementation supports RSA, RSASSA-PSS, DSA, RSAES-OAEP, or Diffie-Hellman, then it MUST support key lengths from 1024-bit to 2048-bit, inclusive. If an implementation supports ECDSA or ECDH, then it MUST support keys on P-256.

7. Security Considerations

The security considerations from [RFC3370], [RFC3560], [RFC4056], [RFC5083], [RFC5084], [RFC5649], [RFC5652], [RFC5753], [RFC5754], and [RFC6031] apply.

The choice of content-encryption algorithms for this document was based on [RFC5649]:

> In the design of some high assurance cryptographic modules, it is desirable to segregate cryptographic keying material from other data. The use of a specific cryptographic mechanism solely for the protection of cryptographic keying material can assist in this goal.

Unfortunately, there is no AES-GCM or AES-CCM mode that provides the same properties. If an AES-GCM and AES-CCM mode that provides the same properties is defined, then this document will be updated to adopt that algorithm.

[SP800-57] provides comparable bits of security for some algorithms and key sizes. [SP800-57] also provides time frames during which certain numbers of bits of security are appropriate, and some environments may find these time frames useful.

8. References

8.1. Normative References

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

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.

[RFC3560] Housley, R., "Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS)", RFC 3560, July 2003.

[RFC4056] Schaad, J., "Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS)", RFC 4056, June 2005.

[RFC5083] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, November 2007.

[RFC5084] Housley, R., "Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS)", RFC 5084, November 2007.

[RFC5280] 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.

[RFC5649] Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, September 2009.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC5753] Turner, S. and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 5753, January 2010.

[RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, January 2010.

[RFC6031] Turner, S. and R. Housley, "Cryptographic Message Syntax (CMS) Symmetric Key Package Content Type", RFC 6031, December 2010.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, February 2011.

8.2. Informative Reference

[SP800-57] National Institute of Standards and Technology (NIST), Special Publication 800-57: Recommendation for Key Management - Part 1 (Revised), March 2007.

Author's Address

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com