Algorithms for Cryptographic Message Syntax (CMS)
Key Package Receipt and Error Content Types

Abstract

This document describes the conventions for using several cryptographic algorithms with the Cryptographic Message Syntax (CMS) key package receipt and error content types. 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/rfc7192.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

This document describes the conventions for using several cryptographic algorithms with the Cryptographic Message Syntax (CMS) key package receipt and error content types [RFC7191]. Specifically, it includes conventions necessary to implement SignedData [RFC5652], EnvelopedData [RFC5652], EncryptedData [RFC5652], and AuthEnvelopedData [RFC5083].

This document does not define any new algorithms; instead, it refers to previously defined algorithms. In fact, the algorithm requirements in this document are the same as those in [RFC5959], [RFC6033], [RFC6160], [RFC6161], and [RFC6162] with the following exceptions: the content-encryption algorithm is AES in Cipher Block Chaining (CBC) mode as opposed to AES Key Wrap with Message Length Indicator (MLI) and the key-wrap algorithm is AES Key Wrap as opposed to AES Key Wrap with MLI. The rationale for the difference is that the receipt and error content types are not keys; therefore, AES Key Wrap with MLI is not appropriate for the content-encryption algorithm. If an implementation is not using AES Key Wrap with MLI as the content-encryption algorithm, then there’s no need to keep the key-wrap algorithm the same as the content encryption algorithm.

NOTE: [RFC7191] only requires that the key package receipt be signed.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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]. Implementations can also choose the to support Elliptic Curve Digital Signature Algorithm (ECDSA) [RFC5753] and [RFC6090].
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) [RFC5753] [RFC6090] MAY be supported.

Regardless of the key management technique choice, implementations MUST support AES-128 in CBC mode [AES] as the content-encryption algorithm. Implementations SHOULD support AES-256 in CBC mode [AES] as the content-encryption algorithm.

When key agreement is used, the same length for the underlying block algorithm MUST be used. If the content-encryption algorithm is AES-128 in CBC mode, then the key-wrap algorithm MUST be AES-128 Key Wrap [RFC3394]. If the content-encryption algorithm is AES-256 in CBC mode, then the key-wrap algorithm MUST be AES-256 Key Wrap [RFC3394].

4. EncryptedData

If an implementation supports EncryptedData, then it MUST implement AES-128 in CBC mode [AES] and SHOULD implement AES-256 in CBC mode [AES].

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 the P-256 curve [RFC6090].

7. Security Considerations

The security considerations from [RFC3370], [RFC3394], [RFC3560], [RFC4056], [RFC5084], [RFC5652], [RFC5753], and [RFC5754] apply.

[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. Acknowledgements

I’d like to thank Russ Housley for his early feedback on this document.

9. References

9.1. Normative References


9. Informative References


Author’s Address

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

EMail: turners@ieca.com