             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.

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

   [AES]   National Institute of Standards and Technology, FIPS Pub
              197: Advanced Encryption Standard (AES), 26 November 2001.

   [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.

   [RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard
              (AES) Key Wrap Algorithm", RFC 3394, September 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.

   [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.

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

   [RFC7191]   Housley, R., "Cryptographic Message Syntax (CMS) Key
               Package Receipt and Error Content Types", RFC 7191, April
               2014.

9.   Informative References

   [RFC5959]   Turner, S., "Algorithms for Asymmetric Key Package Content
               Type", RFC 5959, August 2010.

   [RFC6033]   Turner, S., "Algorithms for Cryptographic Message Syntax
               (CMS) Encrypted Key Package Content Type", RFC 6033,
               December 2010.

   [RFC6160]   Turner, S., "Algorithms for Cryptographic Message Syntax
               (CMS) Protection of Symmetric Key Package Content Types",
               RFC 6160, April 2011.

   [RFC6161]   Turner, S., "Elliptic Curve Algorithms for Cryptographic
               Message Syntax (CMS) Encrypted Key Package Content Type",
               RFC 6161, April 2011.

   [RFC6162] Turner, S., "Elliptic Curve Algorithms for Cryptographic
              Message Syntax (CMS) Asymmetric Key Package Content Type",
              RFC 6162, April 2011.

   [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