Internet Engineering Task Force (IETF)
Request for Comments: 5959
Category: Standards Track
ISSN: 2070-1721

S. Turner
IECA

August 2010

Algorithms for Asymmetric Key Package Content Type

Abstract

This document describes the conventions for using several cryptographic algorithms with the EncryptedPrivateKeyInfo structure, as defined in RFC 5958. It also includes conventions necessary to protect the AsymmetricKeyPackage content type with SignedData, EnvelopedData, EncryptedData, AuthenticatedData, 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/rfc5959.

1. Introduction

This document describes the conventions for using several cryptographic algorithms with the EncryptedPrivateKeyInfo structure [RFC5958]. The EncryptedPrivateKeyInfo is used by [P12] to encrypt PrivateKeyInfo [RFC5958]. It is similar to EncryptedData [RFC5652] in that it has no recipients, no originators, and no content encryption keys and requires keys to be managed by other means.

This document also includes conventions necessary to protect the AsymmetricKeyPackage content type [RFC5958] with Cryptographic Message Syntax (CMS) protecting content types: SignedData [RFC5652], EnvelopedData [RFC5652], EncryptedData [RFC5652], AuthenticatedData [RFC5652], and AuthEnvelopedData [RFC5083]. Implementations of AsymmetricKeyPackage do not require support for any CMS protecting content type; however, if the AsymmetricKeyPackage is CMS protected it is RECOMMENDED that conventions defined herein be followed.

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. EncryptedPrivateKeyInfo

The de facto standard used to encrypt the PrivateKeyInfo structure, which is subsequently placed in the EncryptedPrivateKeyInfo encryptedData field, is Password Based Encryption (PBE) based on PKCS #5 [RFC2898] and PKCS #12 [P12]. The major difference between PKCS #5 and PKCS #12 is the supported encoding for the password: ASCII for PKCS #5 and Unicode for PKCS #12, encoded as specified in Section B.1 of [P12]. [RFC2898] specifies two PBE Schemes (PBES) 1 and 2; [RFC2898] recommends PBES2 for new specification. PBES2 with a key derivation algorithm of PBKDF2 using HMAC with SHA-256 [RFC5754] and an encryption algorithm of AES Key Wrap with Padding as defined in [RFC5649] MUST be supported. AES-256 Key Wrap with Padding [RFC5649] MAY also be supported as an encryption algorithm.

3. AsymmetricKeyPackage

As noted in Asymmetric Key Packages [RFC5958], CMS can be used to protect the AsymmetricKeyPackage. The following provides guidance for SignedData [RFC5652], EnvelopedData [RFC5652], EncryptedData

[RFC5652], AuthenticatedData [RFC5652], and AuthEnvelopedData [RFC5083].

3.1. SignedData

If an implementation supports SignedData, then it MUST support the signature scheme RSA [RFC3370] [RFC5754] and SHOULD support the signature schemes RSASSA-PSS [RFC4056] and DSA [RFC3370] [RFC5754]. Additionally, implementations MUST support in concert with these signature schemes the hash function SHA-256 [RFC5754] and SHOULD support the hash function SHA-1 [RFC3370].

3.2. 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 RSAES-OAEP (RSA Encryption Scheme - Optimal Asymmetric Encryption Padding) [RFC3560] SHOULD be supported.

When key agreement is used, Diffie-Hellman (DH) ephemeral-static [RFC3370] MUST be supported.

Since the content type is used to carry a cryptographic key and its attributes, an algorithm that is traditionally used to encrypt one key with another is employed. 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, a key wrap algorithm is also specified to wrap the content encryption key. 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].

3.3. 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. Since the content type is used to carry a cryptographic key and its attributes, an algorithm that is traditionally used to encrypt one key with another is employed.

3.4. AuthenticatedData

If an implementation supports AuthenticatedData, then it MUST implement SHA-256 [RFC5754] and SHOULD support SHA-1 [RFC3370] as the message digest algorithm. Additionally, HMAC with SHA-256 [RFC4231] MUST be supported and HMAC with SHA-1 [RFC3370] SHOULD be supported.

3.5. 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 for AES-CCM [RFC5084].

4. Public Key Sizes

The easiest way to implement the SignedData, EnvelopedData, and AuthEnvelopedData is with public key certificates [RFC5280]. If an implementation support RSA, RSASSA-PSS, DSS, RSAES-OAEP, or DH, then it MUST support key lengths from 1024-bit to 2048-bit, inclusive.

5. SMIMECapabilities Attribute

[RFC5751] defines the SMIMECapabilities attribute as a mechanism for recipients to indicate their supported capabilities including the algorithms they support. The following are values for the SMIMECapabilities attribute for AES Key Wrap with Padding [RFC5649] when used as a content encryption algorithm:

AES-128 KW with Padding: 30 0d 06 09 60 86 48 01 65 03 04 01 08
AES-192 KW with Padding: 30 0d 06 09 60 86 48 01 65 03 04 01 1C
AES-256 KW with Padding: 30 0d 06 09 60 86 48 01 65 03 04 01 30

6. Security Considerations

The security considerations from [RFC3370], [RFC3560], [RFC4056], [RFC4231], [RFC5083], [RFC5084], [RFC5649], [RFC5652], [RFC5754], and [RFC5958] apply.

The strength of any encryption scheme is only as good as its weakest link, which in the case of a PBES is the password. Passwords need to provide sufficient entropy to ensure they cannot be easily guessed. The U.S. National Institute of Standards and Technology (NIST) Electronic Authentication Guidance [SP800-63] provides some information on password entropy. [SP800-63] indicates that a user-chosen 20-character password from a 94-character keyboard with no checks provides 36 bits of entropy. If the 20-character password is randomly chosen, then the amount of entropy is increased to roughly 131 bits of entropy. The amount of entropy in the password does not correlate directly to bits of security but in general the more than the better.

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.

7. References

7.1. Normative References

[P12] RSA Laboratories, "PKCS #12 v1.0: Personal Information Exchange Syntax", June 1999.

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

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

[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.

[RFC4231] Nystrom, M., "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC 4231, December 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.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

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

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, August 2010.

7.2. Informative References

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

[SP800-63] National Institute of Standards and Technology (NIST), Special Publication 800-63: Electronic Authentication Guidance, April 2006.

Author's Address

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

EMail: turners@ieca.com