Network Working Group
Request for Comments: 5083
Updates: 3852
Category: Standards Track

R. Housley
Vigil Security

November 2007

Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type

Status of This Memo

Abstract

This document describes an additional content type for the Cryptographic Message Syntax (CMS). The authenticated-enveloped-data content type is intended for use with authenticated encryption modes. All of the various key management techniques that are supported in the CMS enveloped-data content type are also supported by the CMS authenticated-enveloped-data content type.

1. Introduction

This document describes an additional content type for the Cryptographic Message Syntax (CMS) [CMS]. The authenticated-enveloped-data content type is intended for use with authenticated encryption modes, where an arbitrary content is both authenticated and encrypted. Also, some associated data in the form of authenticated attributes can also be authenticated. All of the various key management techniques that are supported in the CMS enveloped-data content type are also supported by the CMS authenticated-enveloped-data content type.

The conventions for using the Advanced Encryption Standard-Counter with Cipher Block Chaining-Message Authentication Code (AES-CCM) and the AES-Galois/Counter Mode (GCM) authenticated encryption algorithms with the CMS authenticated-enveloped-data content type defined in this document can be found in [AESALGS].

The authenticated-enveloped-data content type, like all of the other CMS content types, employs ASN.1 [X.208-88], and it uses both the Basic Encoding Rules (BER) [X.209-88] and the Distinguished Encoding Rules (DER) [X.509-88].

1.1. Terminology

In this document, the key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL are to be interpreted as described in [STDWORDS].

1.2. Version Numbers

The major data structure (AuthEnvelopedData) includes a version number as the first item in the data structure. The version number is intended to avoid ASN.1 decode errors. Some implementations do not check the version number prior to attempting a decode, and then if a decode error occurs, the version number is checked as part of the error handling routine. This is a reasonable approach; it places error processing outside of the fast path. This approach is also forgiving when an incorrect version number is used by the sender.

Whenever the structure is updated, a higher version number will be assigned. However, to ensure maximum interoperability, the higher version number is only used when the new syntax feature is employed. That is, the lowest version number that supports the generated syntax is used.

2. Authenticated-Enveloped-Data Content Type

The authenticated-enveloped-data content type consists of an authenticated and encrypted content of any type and encrypted content-authenticated-encryption keys for one or more recipients. The combination of the authenticated and encrypted content and one encrypted content-authenticated-encryption key for a recipient is a "digital envelope" for that recipient. Any type of content can be enveloped for an arbitrary number of recipients using any of the supported key management techniques for each recipient. In addition, authenticated but not encrypted attributes may be provided by the originator.

The typical application of the authenticated-enveloped-data content type will represent one or more recipients' digital envelopes on an encapsulated content.

Authenticated-enveloped-data is constructed by the following steps:

1. A content-authenticated-encryption key for a particular content-authenticated-encryption algorithm is generated at random.

2. The content-authenticated-encryption key is encrypted for each recipient. The details of this encryption depend on the key management algorithm used, but four general techniques are supported:

   Key Transport: the content-authenticated-encryption key is encrypted in the recipient's public key;

   Key Agreement: the recipient's public key and the sender's private key are used to generate a pairwise symmetric key-encryption key, then the content-authenticated-encryption key is encrypted in the pairwise symmetric key-encryption key;

   Symmetric Key-Encryption Keys: the content-authenticated-encryption key is encrypted in a previously distributed symmetric key-encryption key; and

   Passwords: the content-authenticated-encryption key is encrypted in a key-encryption key that is derived from a password or other shared secret value.

3. For each recipient, the encrypted content-authenticated-encryption key and other recipient-specific information are collected into a RecipientInfo value, defined in Section 6.2 of [CMS].

4. Any attributes that are to be authenticated but not encrypted are collected in the authenticated attributes.

5. The attributes collected in step 4 are authenticated and the CMS content is authenticated and encrypted with the content-authenticated-encryption key. If the authenticated encryption algorithm requires either the additional authenticated data (AAD) or the content to be padded to a multiple of some block size, then the padding is added as described in Section 6.3 of [CMS].

6. Any attributes that are to be provided without authentication or encryption are collected in the unauthenticated attributes.

7. The RecipientInfo values for all the recipients, the authenticated attributes, the unauthenticated attributes, and the authenticated and encrypted content are collected together to form an AuthEnvelopedData value as defined in Section 2.1.

A recipient opens the digital envelope by decrypting one of the encrypted content-authenticated-encryption keys, and then using the recovered key to decrypt and verify the integrity of the authenticated and encrypted content as well as to verify the integrity of the authenticated attributes.

The recipient MUST verify the integrity of the received content before releasing any information, especially the plaintext of the content. If the integrity verification fails, the receiver MUST destroy all of the plaintext of the content.

This section is divided into three parts. The first part describes the AuthEnvelopedData content type, the second part describes the authentication and encryption process, and the third part describes the key encryption process.

2.1. AuthEnvelopedData Type

The following object identifier identifies the authenticated-enveloped-data content type:

```
id-ct-authEnvelopedData OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
    smime(16) ct(1) 23 }
```

The authenticated-enveloped-data content type MUST have ASN.1 type AuthEnvelopedData:

```
AuthEnvelopedData ::= SEQUENCE {
  version CMSVersion,
  originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
  recipientInfos RecipientInfos,
  authEncryptedContentInfo EncryptedContentInfo,
  authAttrs [1] IMPLICIT AuthAttributes OPTIONAL,
  mac MessageAuthenticationCode,
  unauthAttrs [2] IMPLICIT UnauthAttributes OPTIONAL }
```

The fields of type AuthEnvelopedData have the following meanings:

version is the syntax version number. It MUST be set to 0.

originatorInfo optionally provides information about the originator. It is present only if required by the key management algorithm. It may contain certificates and Certificate Revocation Lists (CRLs), and the OriginatorInfo type is defined in Section 6.1 of [CMS].

recipientInfos is a collection of per-recipient information. There MUST be at least one element in the collection. The RecipientInfo type is defined in Section 6.2 of [CMS].

authEncryptedContentInfo is the authenticated and encrypted content. The CMS enveloped-data content type uses the same type to carry the encrypted content. The EncryptedContentInfo type is defined in Section 6.1 of [CMS].

authAttrs optionally contains the authenticated attributes. The CMS authenticated-data content type uses the same type to carry authenticated attributes. The authAttrs MUST be present if the content type carried in EncryptedContentInfo is not id-data. AuthAttributes MUST be DER encoded, even if the rest of the AuthEnvelopedData structure is BER encoded. The AuthAttributes type is defined in Section 9.1 of [CMS]; however, in this case, the message-digest attribute SHOULD NOT be included. Useful attribute types are defined in Section 11 of [CMS].

mac is the integrity check value (ICV) or message authentication code (MAC) that is generated by the authenticated encryption algorithm. The CMS authenticated-data content type uses the same type to carry a MAC. In this case, the MAC covers the authenticated attributes and the content directly, and a digest algorithm is not used. The MessageAuthenticationCode type is defined in Section 9.1 of [CMS].

unauthAttrs optionally contains the unauthenticated attributes. The CMS authenticated-data content type uses the same type to carry unauthenticated attributes. The UnauthAttributes type is defined in Section 9.1 of [CMS]. Useful attribute types are defined in Section 11 of [CMS].

2.2. Authentication and Encryption Process

The content-authenticated-encryption key for the desired content-authenticated-encryption algorithm is randomly generated.

If the authenticated encryption algorithm requires the content to be padded to a multiple of some block size, then the padding MUST be added as described in Section 6.3 of [CMS]. This padding method is well defined if and only if the block size is less than 256 octets.

If optional authenticated attributes are present, then they are DER encoded. A separate encoding of the authAttrs field is performed to construct the authenticated associated data (AAD) input to the authenticated encryption algorithm. For the purposes of constructing the AAD, the IMPLICIT [1] tag in the authAttrs field is not used for

the DER encoding: rather a universal SET OF tag is used. That is, the DER encoding of the SET OF tag, rather than of the IMPLICIT [1] tag, is to be included in the construction for the AAD along with the length and content octets of the authAttrs value. If the authenticated encryption algorithm requires the AAD to be padded to a multiple of some block size, then the padding MUST be added as described in Section 6.3 of [CMS]. This padding method is well defined if and only if the block size is less than 256 octets.

If optional authenticated attributes are absent, then zero bits of input are provided for the AAD input to the authenticated encryption algorithm.

The inputs to the authenticated encryption algorithm are the content (the data, which is padded if necessary), the DER-encoded authenticated attributes (the AAD, which is padded if necessary), and the content-authenticated-encryption key. Under control of a content-authenticated-encryption key, the authenticated encryption operation maps an arbitrary string of octets (the data) to another string of octets (the ciphertext) and it computes an authentication tag over the AAD and the data. The encrypted data is included in the AuthEnvelopedData authEncryptedContentInfo encryptedContent as an OCTET STRING, and the authentication tag is included in the AuthEnvelopedData mac.

2.3. Key Encryption Process

The input to the key encryption process -- the value supplied to the recipient's key-encryption algorithm -- is just the "value" of the content-authenticated-encryption key.

Any of the aforementioned key management techniques can be used for each recipient of the same encrypted content.

3. Security Considerations

This specification defines an additional CMS content type. The security considerations provided in [CMS] apply to this content type as well.

Many authenticated encryption algorithms make use of a block cipher in counter mode to provide encryption. When used properly, counter mode provides strong confidentiality. Bellare, Desai, Jokipii, and Rogaway show in [BDJR] that the privacy guarantees provided by counter mode are at least as strong as those for Cipher Block Chaining (CBC) mode when using the same block cipher.

Unfortunately, it is easy to misuse counter mode. If counter block values are ever used for more that one encryption operation with the same key, then the same key stream will be used to encrypt both plaintexts, and the confidentiality guarantees are voided.

Fortunately, the CMS authenticated-enveloped-data content type provides all of the tools needed to avoid misuse of counter mode. All of the existing key management techniques permit a fresh content-encryption key to be generated for each content. In addition, existing authenticated encryption algorithms that make use of counter mode support the use of an unpredictable nonce value in the counter block. This unpredictable nonce value (sometimes called a "salt") should be carried in an algorithm identifier parameter.

Implementations must randomly generate content-authenticated-encryption keys, padding, and unpredictable nonce values. Also, the generation of public/private key pairs relies on a random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate cryptographic keys can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, and then searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult. RFC 4086 [RANDOM] offers important guidance in this area.

If the message-digest attribute is included in the AuthAttributes, then the attribute value will contain the unencrypted one-way hash value of the plaintext of the content. Disclosure of this hash value enables content tracking, and it can be used to determine if the plaintext matches one or more candidate contents. For these reasons, the AuthAttributes SHOULD NOT contain the message-digest attribute.

CMS is often used to provide encryption in messaging environments. In messaging environments, various forms of unsolicited messages (such as spam and phishing) represent a significant volume of unwanted traffic. Present mitigation strategies for unwanted message traffic involve analysis of message plaintext. When recipients accept unsolicited encrypted messages, they become even more vulnerable to unwanted traffic since many present mitigation strategies will be unable to access the plaintext. Therefore, software that receives messages that have been encrypted using CMS needs to provide one or more mechanisms to handle the unwanted message traffic. One approach that does not require disclosure of keying material to a server is to reject or discard encrypted messages unless they purport to come from a member of a white list.

4. ASN.1 Module

```
CMS-AuthEnvelopedData-2007
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0) cms-authEnvelopedData(31) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS All
-- The types and values defined in this module are exported for use
-- in the other ASN.1 modules.  Other applications may use them for
-- their own purposes.

IMPORTS

  -- Imports from RFC 3852 [CMS], Section 12.1
        AuthAttributes, CMSVersion, EncryptedContentInfo,
        MessageAuthenticationCode, OriginatorInfo, RecipientInfos,
        UnauthAttributes
           FROM CryptographicMessageSyntax2004
                { iso(1) member-body(2) us(840) rsadsi(113549)
                  pkcs(1) pkcs-9(9) smime(16) modules(0)
                  cms-2004(24) } ;


id-ct-authEnvelopedData OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
    smime(16) ct(1) 23 }

AuthEnvelopedData ::= SEQUENCE {
  version CMSVersion,
  originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
  recipientInfos RecipientInfos,
  authEncryptedContentInfo EncryptedContentInfo,
  authAttrs [1] IMPLICIT AuthAttributes OPTIONAL,
  mac MessageAuthenticationCode,
  unauthAttrs [2] IMPLICIT UnauthAttributes OPTIONAL }

END -- of CMS-AuthEnvelopedData-2007
```

5. References

5.1. Normative References

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

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

[X.208-88] CCITT. Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1). 1988.

[X.209-88] CCITT. Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). 1988.

[X.509-88] CCITT. Recommendation X.509: The Directory-Authentication Framework. 1988.

5.2. Informative References

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

[BDJR] Bellare, M., Desai, A., Jokipii, E., and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation", Proceedings 38th Annual Symposium on Foundations of Computer Science, 1997.

[RANDOM] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

Author's Address

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA
EMail: housley@vigilsec.com

Full Copyright Statement

Intellectual Property