Network Working Group
Request for Comments: 5084
Category: Standards Track

R. Housley
Vigil Security

November 2007

Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS)

Status of This Memo

Abstract

This document specifies the conventions for using the AES-CCM and the AES-GCM authenticated encryption algorithms with the Cryptographic Message Syntax (CMS) authenticated-enveloped-data content type.

1. Introduction

This document specifies the conventions for using Advanced Encryption Standard-Counter with Cipher Block Chaining-Message Authentication Code (AES-CCM) and AES-Galois/Counter Mode (GCM) authenticated encryption algorithms as the content-authenticated-encryption algorithm with the Cryptographic Message Syntax [CMS] authenticated-enveloped-data content type [AuthEnv].

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 RFC 2119 [STDWORDS].

1.2. ASN.1

CMS values are generated using ASN.1 [X.208-88], which uses the Basic Encoding Rules (BER) [X.209-88] and the Distinguished Encoding Rules (DER) [X.509-88].

1.3. AES

Dr. Joan Daemen and Dr. Vincent Rijmen, both from Belgium, developed the Rijndael block cipher algorithm, and they submitted it for consideration as the Advanced Encryption Standard (AES). Rijndael

was selected by the National Institute for Standards and Technology (NIST), and it is specified in a U.S. Federal Information Processing Standard (FIPS) Publication [AES]. NIST selected the Rijndael algorithm for AES because it offers a combination of security, performance, efficiency, ease of implementation, and flexibility. Specifically, the algorithm performs well in both hardware and software across a wide range of computing environments. Also, the very low memory requirements of the algorithm make it very well suited for restricted-space environments. The AES is widely used by organizations, institutions, and individuals outside of the U.S. Government.

The AES specifies three key sizes: 128, 192, and 256 bits.

1.4. AES-CCM

The Counter with CBC-MAC (CCM) mode of operation is specified in [CCM]. CCM is a generic authenticated encryption block cipher mode. CCM is defined for use with any 128-bit block cipher, but in this document, CCM is used with the AES block cipher.

AES-CCM has four inputs: an AES key, a nonce, a plaintext, and optional additional authenticated data (AAD). AES-CCM generates two outputs: a ciphertext and a message authentication code (also called an authentication tag).

The nonce is generated by the party performing the authenticated encryption operation. Within the scope of any authenticated-encryption key, the nonce value MUST be unique. That is, the set of nonce values used with any given key MUST NOT contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties.

AAD is authenticated but not encrypted. Thus, the AAD is not included in the AES-CCM output. It can be used to authenticate plaintext packet headers. In the CMS authenticated-enveloped-data content type, authenticated attributes comprise the AAD.

1.5. AES-GCM

The Galois/Counter Mode (GCM) is specified in [GCM]. GCM is a generic authenticated encryption block cipher mode. GCM is defined for use with any 128-bit block cipher, but in this document, GCM is used with the AES block cipher.

AES-GCM has four inputs: an AES key, an initialization vector (IV), a plaintext content, and optional additional authenticated data (AAD). AES-GCM generates two outputs: a ciphertext and message

authentication code (also called an authentication tag). To have a common set of terms for AES-CCM and AES-GCM, the AES-GCM IV is referred to as a nonce in the remainder of this document.

The nonce is generated by the party performing the authenticated encryption operation. Within the scope of any authenticated-encryption key, the nonce value MUST be unique. That is, the set of nonce values used with any given key MUST NOT contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties.

AAD is authenticated but not encrypted. Thus, the AAD is not included in the AES-GCM output. It can be used to authenticate plaintext packet headers. In the CMS authenticated-enveloped-data content type, authenticated attributes comprise the AAD.

2. Automated Key Management

The reuse of an AES-CCM or AES-GCM nonce/key combination destroys the security guarantees. As a result, it can be extremely difficult to use AES-CCM or AES-GCM securely when using statically configured keys. For safety's sake, implementations MUST use an automated key management system [KEYMGMT].

The CMS authenticated-enveloped-data content type supports four general key management techniques:

- 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, then the content-authenticated-encryption key is encrypted in the pairwise symmetric 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.

All of these key management techniques meet the automated key management system requirement as long as a fresh content-authenticated-encryption key is generated for the protection of each content. Note that some of these key management techniques use one key-encryption key to encrypt more than one content-authenticated-

encryption key during the system life cycle. As long as fresh content-authenticated-encryption key is used each time, AES-CCM and AES-GCM can be used safely with the CMS authenticated-enveloped-data content type.

In addition to these four general key management techniques, CMS supports other key management techniques. See Section 6.2.5 of [CMS]. Since the properties of these key management techniques are unknown, no statement can be made about whether these key management techniques meet the automated key management system requirement. Designers and implementers must perform their own analysis if one of these other key management techniques is supported.

3. Content-Authenticated Encryption Algorithms

This section specifies the conventions employed by CMS implementations that support content-authenticated encryption using AES-CCM or AES-GCM.

Content-authenticated encryption algorithm identifiers are located in the AuthEnvelopedData EncryptedContentInfo contentEncryptionAlgorithm field.

Content-authenticated encryption algorithms are used to encipher the content located in the AuthEnvelopedData EncryptedContentInfo encryptedContent field and to provide the message authentication code for the AuthEnvelopedData mac field. Note that the message authentication code provides integrity protection for both the AuthEnvelopedData authAttrs and the AuthEnvelopedData EncryptedContentInfo encryptedContent.

3.1. AES-CCM

The AES-CCM authenticated encryption algorithm is described in [CCM]. A brief summary of the properties of AES-CCM is provided in Section 1.4.

Neither the plaintext content nor the optional AAD inputs need to be padded prior to invoking AES-CCM.

There are three algorithm identifiers for AES-CCM, one for each AES key size:

```
aes OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840)
    organization(1) gov(101) csor(3) nistAlgorithm(4) 1 }

id-aes128-CCM OBJECT IDENTIFIER ::= { aes 7 }

id-aes192-CCM OBJECT IDENTIFIER ::= { aes 27 }

id-aes256-CCM OBJECT IDENTIFIER ::= { aes 47 }
```

With all three AES-CCM algorithm identifiers, the AlgorithmIdentifier parameters field MUST be present, and the parameters field must contain a CCMParameter:

```
CCMParameters ::= SEQUENCE {
  aes-nonce         OCTET STRING (SIZE(7..13)),
  aes-ICVlen        AES-CCM-ICVlen DEFAULT 12 }

AES-CCM-ICVlen ::= INTEGER (4 | 6 | 8 | 10 | 12 | 14 | 16)
```

The aes-nonce parameter field contains 15-L octets, where L is the size of the length field. With the CMS, the normal situation is for the content-authenticated-encryption key to be used for a single content; therefore, L=8 is RECOMMENDED. See [CCM] for a discussion of the trade-off between the maximum content size and the size of the nonce. Within the scope of any content-authenticated-encryption key, the nonce value MUST be unique. That is, the set of nonce values used with any given key MUST NOT contain any duplicate values.

The aes-ICVlen parameter field tells the size of the message authentication code. It MUST match the size in octets of the value in the AuthEnvelopedData mac field. A length of 12 octets is RECOMMENDED.

3.2. AES-GCM

The AES-GCM authenticated encryption algorithm is described in [GCM]. A brief summary of the properties of AES-CCM is provided in Section 1.5.

Neither the plaintext content nor the optional AAD inputs need to be padded prior to invoking AES-GCM.

There are three algorithm identifiers for AES-GCM, one for each AES key size:

```
aes OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840)
    organization(1) gov(101) csor(3) nistAlgorithm(4) 1 }

id-aes128-GCM OBJECT IDENTIFIER ::= { aes 6 }

id-aes192-GCM OBJECT IDENTIFIER ::= { aes 26 }

id-aes256-GCM OBJECT IDENTIFIER ::= { aes 46 }
```

With all three AES-GCM algorithm identifiers, the AlgorithmIdentifier parameters field MUST be present, and the parameters field must contain a GCMParameter:

```
GCMParameters ::= SEQUENCE {
  aes-nonce        OCTET STRING, -- recommended size is 12 octets
  aes-ICVlen       AES-GCM-ICVlen DEFAULT 12 }

AES-GCM-ICVlen ::= INTEGER (12 | 13 | 14 | 15 | 16)
```

The aes-nonce is the AES-GCM initialization vector. The algorithm specification permits the nonce to have any number of bits between 1 and 2^64. However, the use of OCTET STRING within GCMParameters requires the nonce to be a multiple of 8 bits. Within the scope of any content-authenticated-encryption key, the nonce value MUST be unique, but need not have equal lengths. A nonce value of 12 octets can be processed more efficiently, so that length is RECOMMENDED.

The aes-ICVlen parameter field tells the size of the message authentication code. It MUST match the size in octets of the value in the AuthEnvelopedData mac field. A length of 12 octets is RECOMMENDED.

4. Security Considerations

AES-CCM and AES-GCM make use of the AES 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 than 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 AuthEnvelopedData provides all the tools needed to avoid misuse of counter mode. Automated key management is discussed in Section 2.

There are fairly generic precomputation attacks against the use of any block cipher in counter mode that allow a meet-in-the-middle attack against the key [H][B][MF]. AES-CCM and AES-GCM both make use of counter mode for encryption. These precomputation attacks require the creation and searching of huge tables of ciphertext associated with known plaintext and known keys. Assuming that the memory and processor resources are available for a precomputation attack, then the theoretical strength of any block cipher in counter mode is limited to 2^(n/2) bits, where n is the number of bits in the key. The use of long keys is the best countermeasure to precomputation attacks. Use of an unpredictable nonce value in the counter block significantly increases the size of the table that the attacker must compute to mount a successful precomputation attack.

Implementations must randomly generate content-authenticated-encryption keys. 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.

5. References

5.1. Normative References

[AES] NIST, FIPS PUB 197, "Advanced Encryption Standard (AES)", November 2001.

[CCM] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

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

[GCM] Dworkin, M., "NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC." , U.S. National Institute of Standards and Technology http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[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

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

[B] Biham, E., "How to Forge DES-Encrypted Messages in 2^28 Steps", Technion Computer Science Department Technical Report CS0884, 1996.

[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.

[H] Hellman, M. E., "A cryptanalytic time-memory trade-off", IEEE Transactions on Information Theory, July 1980, pp. 401-406.

[KEYMGMT] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[MF] McGrew, D., and S. Fluhrer, "Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security", The Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag, August, 2000.

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

Appendix: ASN.1 Module

```
CMS-AES-CCM-and-AES-GCM
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs-9(9) smime(16) modules(0) cms-aes-ccm-and-gcm(32) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS All

-- Object Identifiers

aes OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840)
    organization(1) gov(101) csor(3) nistAlgorithm(4) 1 }

id-aes128-CCM OBJECT IDENTIFIER ::= { aes 7 }

id-aes192-CCM OBJECT IDENTIFIER ::= { aes 27 }

id-aes256-CCM OBJECT IDENTIFIER ::= { aes 47 }

id-aes128-GCM OBJECT IDENTIFIER ::= { aes 6 }

id-aes192-GCM OBJECT IDENTIFIER ::= { aes 26 }

id-aes256-GCM OBJECT IDENTIFIER ::= { aes 46 }


-- Parameters for AigorithmIdentifier

CCMParameters ::= SEQUENCE {
  aes-nonce         OCTET STRING (SIZE(7..13)),
  aes-ICVlen        AES-CCM-ICVlen DEFAULT 12 }

AES-CCM-ICVlen ::= INTEGER (4 | 6 | 8 | 10 | 12 | 14 | 16)

GCMParameters ::= SEQUENCE {
  aes-nonce        OCTET STRING, -- recommended size is 12 octets
  aes-ICVlen       AES-GCM-ICVlen DEFAULT 12 }

AES-GCM-ICVlen ::= INTEGER (12 | 13 | 14 | 15 | 16)

END
```

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