ï»¿

Internet Engineering Task Force (IETF)   R. Housley
Request for Comments: 9044   Vigil Security
Category: Standards Track   June 2021
ISSN: 2070-1721

Using the AES-GMAC Algorithm with the Cryptographic Message Syntax (CMS)

Abstract

This document specifies the conventions for using the AES-GMAC Message Authentication Code algorithm with the Cryptographic Message Syntax (CMS) as specified in RFC 5652.

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 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9044.

Copyright Notice

Table of Contents

1. Introduction

This document specifies the conventions for using the AES-GMAC [AES] [GCM] Message Authentication Code (MAC) algorithm with the Cryptographic Message Syntax (CMS) [RFC5652].

2. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Message Authentication Code Algorithms

This section specifies the conventions employed by CMS [RFC5652] implementations that support the AES-GMAC [AES] [GCM] Message Authentication Code (MAC) algorithm.

MAC algorithm identifiers are located in the AuthenticatedData macAlgorithm field.

MAC values are located in the AuthenticatedData mac field.

3.1. AES-GMAC

The AES-GMAC [AES] [GCM] Message Authentication Code (MAC) algorithm uses one of the following algorithm identifiers in the AuthenticatedData macAlgorithm field; the choice depends on the size of the AES key, which is either 128 bits, 192 bits, or 256 bits:

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

id-aes128-GMAC OBJECT IDENTIFIER ::= { aes 9 }

id-aes192-GMAC OBJECT IDENTIFIER ::= { aes 29 }

id-aes256-GMAC OBJECT IDENTIFIER ::= { aes 49 }
```

For all three of these algorithm identifier values, the AlgorithmIdentifier parameters field MUST be present, and the parameters MUST contain GMACParameters:

```
GMACParameters ::= SEQUENCE {
   nonce        OCTET STRING, -- recommended size is 12 octets
   length       MACLength DEFAULT 12 }

MACLength ::= INTEGER (12 | 13 | 14 | 15 | 16)
```

The GMACParameters nonce field is the GMAC initialization vector. The nonce may have any number of bits between 8 and (2^64)-1, but it MUST be a multiple of 8 bits. Within the scope of any content-authentication key, the nonce value MUST be unique. A nonce value of 12 octets can be processed more efficiently, so that length for the nonce value is RECOMMENDED.

The GMACParameters length field tells the size of the message authentication code. It MUST match the size in octets of the value in the AuthenticatedData mac field. A length of 12 octets is RECOMMENDED.

4. Implementation Considerations

An implementation of the Advanced Encryption Standard (AES) Galois/Counter Mode (GCM) authenticated encryption algorithm is specified in [GCM]. An implementation of AES-GCM can be used to compute the GMAC message authentication code by providing the content-authentication key as the AES key, the nonce as the initialization vector, a zero-length plaintext content, and the content to be authenticated as the additional authenticated data (AAD). The result of the AES-GCM invocation is the AES-GMAC authentication code, which is called the "authentication tag" in some implementations. In AES-GCM, the encryption step is skipped when no input plaintext is provided; therefore, no ciphertext is produced.

The DEFAULT and RECOMMENDED values in GMACParameters were selected to align with the parameters defined for AES-GCM in Section 3.2 of

[RFC5084].

5. ASN.1 Module

The following ASN.1 module uses the definition for MAC-ALGORITHM from [RFC5912].

```
CryptographicMessageSyntaxGMACAlgorithms
    { iso(1) member-body(2) us(840) rsadsi(113549)
      pkcs(1) pkcs-9(9) smime(16) modules(0)
      id-mod-aes-gmac-alg-2020(72) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS All

IMPORTS
  AlgorithmIdentifier{}, MAC-ALGORITHM
  FROM AlgorithmInformation-2009 -- from [RFC5912]
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-algorithmInformation-02(58)} ;

-- 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-GMAC OBJECT IDENTIFIER ::= { aes 9 }

id-aes192-GMAC OBJECT IDENTIFIER ::= { aes 29 }

id-aes256-GMAC OBJECT IDENTIFIER ::= { aes 49 }

-- GMAC Parameters

GMACParameters ::= SEQUENCE {
   nonce        OCTET STRING, -- recommended size is 12 octets
   length       MACLength DEFAULT 12 }

MACLength ::= INTEGER (12 | 13 | 14 | 15 | 16)

-- Algorithm Identifiers

maca-aes128-GMAC MAC-ALGORITHM ::= {
   IDENTIFIER id-aes128-GMAC
   PARAMS TYPE GMACParameters ARE required
   IS-KEYED-MAC TRUE }

maca-aes192-GMAC MAC-ALGORITHM ::= {
   IDENTIFIER id-aes192-GMAC
   PARAMS TYPE GMACParameters ARE required
   IS-KEYED-MAC TRUE }

maca-aes256-GMAC MAC-ALGORITHM ::= {
   IDENTIFIER id-aes256-GMAC
   PARAMS TYPE GMACParameters ARE required
   IS-KEYED-MAC TRUE }

END -- of CryptographicMessageSyntaxGMACAlgorithms
```

6. IANA Considerations

IANA has registered the object identifier shown in Table 1 in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry.

Decimal	Description	References

72	id-mod-aes-gmac-alg-2020	RFC 9044

Table 1

7. Security Considerations

The CMS provides a method for authenticating data. This document identifies the conventions for using the AES-GMAC algorithm with the CMS.

The key management technique employed to distribute message-authentication keys must itself provide authentication; otherwise, the content is delivered with integrity from an unknown source.

When more than two parties share the same message-authentication key, data origin authentication is not provided. Any party that knows the message-authentication key can compute a valid MAC; therefore, the content could originate from any one of the parties.

Within the scope of any content-authentication key, the AES-GMAC nonce value MUST be unique. Use of a nonce value more than once allows an attacker to generate valid AES-GMAC authentication codes for arbitrary messages, resulting in the loss of authentication as described in Appendix A of [GCM].

Within the scope of any content-authentication key, the authentication tag length (MACLength) MUST be fixed.

If AES-GMAC is used as a building block in another algorithm (e.g., as a pseudorandom function), AES-GMAC MUST be used only one time by that algorithm. For instance, AES-GMAC MUST NOT be used as the pseudorandom function for PBKDF2.

When initialization vector (IV) lengths other than 96 bits are used, the GHASH function is used to process the provided IV, which introduces a potential for IV collisions. However, IV collisions are not a concern with CMS AuthenticatedData because a fresh content-authentication key is usually generated for each message.

The probability of a successful forgery is close to 2^(-t), where t is the number of bits in the authentication tag length (MACLength*8). This nearly ideal authentication protection is achieved for CMS AuthenticatedData when a fresh content-authentication key is generated for each message. However, the strength of GMAC degrades slightly as a function of the length of the message being authenticated [F2005] [MV2005]. Implementations SHOULD use 16-octet authentication tags for messages over 2^64 octets.

Implementations must randomly generate message-authentication keys. The use of inadequate pseudorandom number generators (PRNGs) to generate keys can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute-force searching the whole key space. The generation of quality random numbers is difficult. [RFC4086] offers important guidance in this area.

Implementers should be aware that cryptographic algorithms become weaker with time. As new cryptanalysis techniques are developed and computing performance improves, the work factor to break a particular cryptographic algorithm will reduce. Therefore, cryptographic algorithm implementations should be modular, allowing new algorithms to be readily inserted. That is, implementers should be prepared to regularly update the set of algorithms in their implementations. More information is available in BCP 201 [RFC7696].

8. References

8.1. Normative References

[AES] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, DOI 10.6028/NIST.FIPS.197, November 2001, <https://doi.org/10.6028/NIST.FIPS.197>.

[GCM] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-38D, DOI 10.6028/NIST.SP.800-38D, November 2007, <https://doi.org/10.6028/NIST.SP.800-38D>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[F2005] Ferguson, N., "Authentication weaknesses in GCM", May 2005, <https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf>.

[MV2005] McGrew, D. and J. Viega, "GCM Update", May 2005, <https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/CWC-GCM/gcm-update.pdf>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC5084] Housley, R., "Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS)", RFC 5084, DOI 10.17487/RFC5084, November 2007, <https://www.rfc-editor.org/info/rfc5084>.

[RFC7696] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

Acknowledgements

Many thanks to Hans Aschauer, Hendrik Brockhaus, Quynh Dang, Roman Danyliw, Tim Hollebeek, Ben Kaduk, Mike Ounsworth, and Magnus Westerlund for their careful review and thoughtful improvements.

Author’s Address

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA 20170
United States of America

Email: housley@vigilsec.com