Network Working Group
Request for Comments: 5288
Category: Standards Track

J. Salowey
A. Choudhury
D. McGrew
Cisco Systems, Inc.

August 2008

AES Galois Counter Mode (GCM) Cipher Suites for TLS

Status of This Memo

Abstract

This memo describes the use of the Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM) as a Transport Layer Security (TLS) authenticated encryption operation. GCM provides both confidentiality and data origin authentication, can be efficiently implemented in hardware for speeds of 10 gigabits per second and above, and is also well-suited to software implementations. This memo defines TLS cipher suites that use AES-GCM with RSA, DSA, and Diffie-Hellman-based key exchange mechanisms.

Table of Contents

1. Introduction

This document describes the use of AES [AES] in Galois Counter Mode (GCM) [GCM] (AES-GCM) with various key exchange mechanisms as a cipher suite for TLS. AES-GCM is an authenticated encryption with associated data (AEAD) cipher (as defined in TLS 1.2 [RFC5246]) providing both confidentiality and data origin authentication. The following sections define cipher suites based on RSA, DSA, and Diffie-Hellman key exchanges; ECC-based (Elliptic Curve Cryptography) cipher suites are defined in a separate document [RFC5289].

AES-GCM is not only efficient and secure, but hardware implementations can achieve high speeds with low cost and low latency, because the mode can be pipelined. Applications that require high data throughput can benefit from these high-speed implementations. AES-GCM has been specified as a mode that can be used with IPsec ESP [RFC4106] and 802.1AE Media Access Control (MAC) Security [IEEE8021AE].

2. Conventions Used in This Document

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].

3. AES-GCM Cipher Suites

The following cipher suites use the new authenticated encryption modes defined in TLS 1.2 with AES in Galois Counter Mode (GCM) [GCM]:

```
CipherSuite TLS_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0x9C}
CipherSuite TLS_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0x9D}
CipherSuite TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0x9E}
CipherSuite TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0x9F}
CipherSuite TLS_DH_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0xA0}
CipherSuite TLS_DH_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0xA1}
CipherSuite TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 = {0x00,0xA2}
CipherSuite TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 = {0x00,0xA3}
CipherSuite TLS_DH_DSS_WITH_AES_128_GCM_SHA256 = {0x00,0xA4}
CipherSuite TLS_DH_DSS_WITH_AES_256_GCM_SHA384 = {0x00,0xA5}
CipherSuite TLS_DH_anon_WITH_AES_128_GCM_SHA256 = {0x00,0xA6}
CipherSuite TLS_DH_anon_WITH_AES_256_GCM_SHA384 = {0x00,0xA7}
```

These cipher suites use the AES-GCM authenticated encryption with associated data (AEAD) algorithms AEAD_AES_128_GCM and AEAD_AES_256_GCM described in [RFC5116]. Note that each of these AEAD algorithms uses a 128-bit authentication tag with GCM (in particular, as described in Section 3.5 of [RFC4366], the

"truncated_hmac" extension does not have an effect on cipher suites that do not use HMAC). The "nonce" SHALL be 12 bytes long consisting of two parts as follows: (this is an example of a "partially explicit" nonce; see Section 3.2.1 in [RFC5116]).

```
struct {
   opaque salt[4];
   opaque nonce_explicit[8];
} GCMNonce;
```

The salt is the "implicit" part of the nonce and is not sent in the packet. Instead, the salt is generated as part of the handshake process: it is either the client_write_IV (when the client is sending) or the server_write_IV (when the server is sending). The salt length (SecurityParameters.fixed_iv_length) is 4 octets.

The nonce_explicit is the "explicit" part of the nonce. It is chosen by the sender and is carried in each TLS record in the GenericAEADCipher.nonce_explicit field. The nonce_explicit length (SecurityParameters.record_iv_length) is 8 octets.

Each value of the nonce_explicit MUST be distinct for each distinct invocation of the GCM encrypt function for any fixed key. Failure to meet this uniqueness requirement can significantly degrade security. The nonce_explicit MAY be the 64-bit sequence number.

The RSA, DHE_RSA, DH_RSA, DHE_DSS, DH_DSS, and DH_anon key exchanges are performed as defined in [RFC5246].

The Pseudo Random Function (PRF) algorithms SHALL be as follows:

For cipher suites ending with _SHA256, the PRF is the TLS PRF [RFC5246] with SHA-256 as the hash function.

For cipher suites ending with _SHA384, the PRF is the TLS PRF [RFC5246] with SHA-384 as the hash function.

Implementations MUST send TLS Alert bad_record_mac for all types of failures encountered in processing the AES-GCM algorithm.

4. TLS Versions

These cipher suites make use of the authenticated encryption with additional data defined in TLS 1.2 [RFC5246]. They MUST NOT be negotiated in older versions of TLS. Clients MUST NOT offer these cipher suites if they do not offer TLS 1.2 or later. Servers that select an earlier version of TLS MUST NOT select one of these cipher suites. Because TLS has no way for the client to indicate that it

supports TLS 1.2 but not earlier, a non-compliant server might potentially negotiate TLS 1.1 or earlier and select one of the cipher suites in this document. Clients MUST check the TLS version and generate a fatal "illegal_parameter" alert if they detect an incorrect version.

5. IANA Considerations

IANA has assigned the following values for the cipher suites defined in this document:

```
CipherSuite TLS_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0x9C}
CipherSuite TLS_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0x9D}
CipherSuite TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0x9E}
CipherSuite TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0x9F}
CipherSuite TLS_DH_RSA_WITH_AES_128_GCM_SHA256 = {0x00,0xA0}
CipherSuite TLS_DH_RSA_WITH_AES_256_GCM_SHA384 = {0x00,0xA1}
CipherSuite TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 = {0x00,0xA2}
CipherSuite TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 = {0x00,0xA3}
CipherSuite TLS_DH_DSS_WITH_AES_128_GCM_SHA256 = {0x00,0xA4}
CipherSuite TLS_DH_DSS_WITH_AES_256_GCM_SHA384 = {0x00,0xA5}
CipherSuite TLS_DH_anon_WITH_AES_128_GCM_SHA256 = {0x00,0xA6}
CipherSuite TLS_DH_anon_WITH_AES_256_GCM_SHA384 = {0x00,0xA7}
```

6. Security Considerations

The security considerations in [RFC5246] apply to this document as well. The remainder of this section describes security considerations specific to the cipher suites described in this

document.

6.1. Counter Reuse

AES-GCM security requires that the counter is never reused. The IV construction in Section 3 is designed to prevent counter reuse.

Implementers should also understand the practical considerations of IV handling outlined in Section 9 of [GCM].

6.2. Recommendations for Multiple Encryption Processors

If multiple cryptographic processors are in use by the sender, then the sender MUST ensure that, for a particular key, each value of the nonce_explicit used with that key is distinct. In this case, each encryption processor SHOULD include, in the nonce_explicit, a fixed value that is distinct for each processor. The recommended format is

```
nonce_explicit = FixedDistinct || Variable
```


where the FixedDistinct field is distinct for each encryption processor, but is fixed for a given processor, and the Variable field is distinct for each distinct nonce used by a particular encryption processor. When this method is used, the FixedDistinct fields used by the different processors MUST have the same length.

In the terms of Figure 2 in [RFC5116], the Salt is the Fixed-Common part of the nonce (it is fixed, and it is common across all encryption processors), the FixedDistinct field exactly corresponds to the Fixed-Distinct field, the Variable field corresponds to the Counter field, and the explicit part exactly corresponds to the nonce_explicit.

For clarity, we provide an example for TLS in which there are two distinct encryption processors, each of which uses a one-byte FixedDistinct field:

```
Salt          = eedc68dc
FixedDistinct = 01       (for the first encryption processor)
FixedDistinct = 02       (for the second encryption processor)
```

The GCMnonces generated by the first encryption processor, and their

corresponding nonce_explicit, are:

GCMNonce	nonce_explicit
eedc68dc0100000000000000	0100000000000000
eedc68dc0100000000000001	0100000000000001
eedc68dc0100000000000002	0100000000000002
...	

The GCMnonces generated by the second encryption processor, and their corresponding nonce_explicit, are

GCMNonce	nonce_explicit
eedc68dc0200000000000000	0200000000000000
eedc68dc0200000000000001	0200000000000001
eedc68dc0200000000000002	0200000000000002
...	

7. Acknowledgements

This document borrows heavily from [RFC5289]. The authors would like to thank Alex Lam, Simon Josefsson, and Pasi Eronen for providing useful comments during the review of this document.

8. References

8.1. Normative References

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

[GCM] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", National Institute of Standards and Technology SP 800-38D, November 2007.

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

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

8.2. Informative References

[IEEE8021AE] Institute of Electrical and Electronics Engineers, "Media Access Control Security", IEEE Standard 802.1AE, August 2006.

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, June 2005.

[RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

[RFC5289] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode", RFC 5289, August 2008.

Authors' Addresses

Joseph Salowey
Cisco Systems, Inc.
2901 3rd. Ave
Seattle, WA 98121
USA

EMail: jsalowey@cisco.com

Abhijit Choudhury
Cisco Systems, Inc.
3625 Cisco Way
San Jose, CA 95134
USA

EMail: abhijitc@cisco.com

David McGrew
Cisco Systems, Inc.
170 W Tasman Drive
San Jose, CA 95134
USA

EMail: mcgrew@cisco.com

Intellectual Property