TLS Working Group J. Salowey
Internet-Draft A. Choudhury
Intended status: Standards Track D. McGrew
Expires: December 20, 2007 Cisco Systems, Inc.
June 18, 2007

RSA based AES-GCM Cipher Suites for TLS
draft-ietf-tls-rsa-aes-gcm-00

Status of this Memo

Copyright Notice

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 ciphersuites that use AES-GCM with RSA.

Table of Contents

1. Introduction

This document describes the use of AES [AES]in Galois Counter Mode (GCM) [GCM] (AES-GCM) with RSA based key exchange as a ciphersuite for TLS. This mechanism is not only efficient and secure, hardware implementations can achieve high speeds with low cost and low latency, because the mode can be pipelined. Applications like CAPWAP, which uses DTLS, can benefit from the high-speed implementations when wireless termination points (WTPs) and controllers (ACs) have to meet requirements to support higher throughputs in the future. AES-GCM has been specified as a mode that can be used with IPsec ESP [RFC4106] and 802.1AE MAC Security [IEEE8021AE]. It also is part of the NSA suite B ciphersuites for TLS [I-D.rescorla-tls-suiteb]; however in order to meet Suite B requirements these ciphersuites require ECC base key exchange and TLS 1.2. The ciphersuites defined in this document are based on RSA which allows the use of AES-GCM in environments that have not deployed ECC algorithms and do not need to meet NSA Suite B requirements. AES-GCM is an authenticated encryption with associated data (AEAD) operation, as used in TLS 1.2[I-D.ietf-tls-rfc4346-bis]. The ciphersuites defined in this draft may be used with DTLS defined in [RFC4347] and [I-D.ietf-tls-ecc-new-mac]. This memo uses GCM in a way similar to [I-D.rescorla-tls-suiteb].

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. RSA Based AES-GCM Cipher Suites

The ciphersuites defined in this document are based on the the AES-GCM authenticated encryption with associated data (AEAD) algorithms

AEAD_AES_128_GCM and AEAD_AES_256_GCM described in [I-D.mcgrew-auth-enc]. Note that this specification uses a 128-bit authentication tag with GCM. The following ciphersuites are defined:

```
CipherSuite TLS_RSA_WITH_AES_128_GCM_SHA256 = {TBD1,TBD1}
CipherSuite TLS_RSA_WITH_AES_256_GCM_SHA384 = {TBD2,TBD2}
CipherSuite TLS_RSA_DHE_WITH_AES_128_GCM_SHA256 = {TBD3,TBD3}
CipherSuite TLS_RSA_DHE_WITH_AES_256_GCM_SHA384 = {TBD4,TBD4}
```

The "nonce" SHALL be 12 bytes long and it is "partially implicit" (see section 3.2.1 in [I-D.mcgrew-auth-enc]); that is, part of the nonce is explicitly carried in each packet, and part of the nonce is

implicit. The nonce is constructed from a salt and an explicit Counter, sent as part of the packet, as follows:

```
Struct{
   opaque salt[4];
   opaque explicit_nonce_part[8];
} GCMNonce
```

The salt is the "implicit" part of the nonce and is not sent in the packet. It is either the client_write_IV if the client is sending or the server_write_IV if the server is sending. These IVs SHALL be 4 bytes long.

The explicit_nonce_part is chosen by the sender and included in the packet. Each value of the explicit_nonce_part MUST be distinct for each distinct invocation of GCM encrypt function using a particular fixed key. Failure to meet this uniqueness requirement can significantly degrade security. The explicit_nonce_part is carried in the IV field of the GenericAEADCipher structure. Therefore, for all the algorithms defined in this section, SecurityParameters.iv_length=8.

In the case of TLS the explicit_nonce_part MAY be the 64-bit sequence number. In the case of DTLS the explicit_nonce_part MAY be the 16-bit epoch with the concatenated 48-bit DTLS seq_num.

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

```
explicit_nonce_part = 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 [I-D.mcgrew-auth-enc], 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, and the Variable field corresponds to the Counter field, and the explicit part exactly corresponds to the explicit_nonce_part.

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 explicit_nonce_parts, are:

```
GCMNonce                    explicit_nonce_part
------------------------    ----------------------------
eedc68dc0100000000000000    0100000000000000
eedc68dc0100000000000001    0100000000000001
eedc68dc0100000000000002    0100000000000002
...
```

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

```
GCMNonce                    explicit_nonce_part
```

```
------------------------   ----------------------------
eedc68dc0200000000000000   0200000000000000
eedc68dc0200000000000001   0200000000000001
eedc68dc0200000000000002   0200000000000002
...
```

The RSA and RSA-DHE key exchange is performed as defined in [I-D.ietf-tls-rfc4346-bis].

Recall that an AEAD operation replaces the use of HMAC as a MAC, but not as a PRF for the purposes of key derivation. Consequently, the hash function for the TLS PRF must be explicitly specified by the ciphersuite. The PRF algorithms SHALL be as follows:

For TLS_RSA_WITH_AES_128_GCM_SHA256 and TLS_RSA_DHE_WITH_AES_128_GCM_SHA256 the hash function is SHA256.

For TLS_RSA_WITH_AES_256_GCM_SHA384 and TLS_RSA_DHE_WITH_AES_256_GCM_SHA384 the hash function is SHA384.

4. TLS Versions

These ciphersuites make use of the authenticated encryption with additional data defined in TLS 1.2 [I-D.ietf-tls-rfc4346-bis]. 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 which 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 ciphersuites defined in this draft:

```
CipherSuite TLS_RSA_WITH_AES_128_GCM_SHA256 = {TBD1,TBD1}
CipherSuite TLS_RSA_WITH_AES_256_GCM_SHA384 = {TBD2,TBD2}
CipherSuite TLS_RSA_DHE_WITH_AES_128_GCM_SHA256 = {TBD3,TBD3}
CipherSuite TLS_RSA_DHE_WITH_AES_256_GCM_SHA384 = {TBD4,TBD4}
```

6. Security Considerations

6.1. Perfect Forward Secrecy

The perfect forward secrecy properties of RSA based TLS ciphersuites are discussed in the security analysis of [RFC4346]. It should be noted that only the ephemeral Diffie-Hellman based ciphersuites are capable of providing perfect forward secrecy.

6.2. Counter Reuse

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

7. Acknowledgements

This draft borrows heavily from [I-D.ietf-tls-ecc-new-mac] and [I-D.rescorla-tls-suiteb]

8. References

8.1. Normative References

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

[GCM] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation:

Galois Counter Mode (GCM) for Confidentiality and Authentication", SP 800-38D, April 2006.

[I-D.ietf-tls-rfc4346-bis]
Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.2", draft-ietf-tls-rfc4346-bis-03 (work in progress), March 2007.

[I-D.mcgrew-auth-enc]
McGrew, D., "An Interface and Algorithms for Authenticated Encryption", draft-mcgrew-auth-enc-02 (work in progress), March 2007.

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

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

8.2. Informative References

[I-D.ietf-tls-ecc-new-mac]
Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode", draft-ietf-tls-ecc-new-mac-01 (work in progress), June 2007.

[I-D.rescorla-tls-suiteb]
Salter, M. and E. Rescorla, "Suite B Cipher Suites for TLS", draft-rescorla-tls-suiteb-01 (work in progress), June 2007.

[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.

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

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).