Network Working Group
Internet Draft
Intended Status: Informational
Expires: June 08, 2009

K.M. Igoe
National Security Agency
December 05, 2008
J.A. Solinas
National Security Agency
December 05, 2008

AES Galois Counter Mode for the Secure Shell Transport Layer Protocol

draft-igoe-secsh-aes-gcm-01

Status of this Memo

Copyright Notice

Abstract

Secure Shell (SSH, RFC 4251) is a secure remote-login protocol. SSH provides for algorithms that provide authentication, key agreement, confidentiality and data integrity services. The purpose of this document is to show how the AES Galois/Counter Mode can be used to provide both confidentiality and data integrity to the SSH Transport Layer

Table of Contents

1. Introduction

Galois/Counter Mode (GCM) is a block cipher mode of operation that provides both confidentiality and data integrity services. The purpose of this document is to show how AES-GCM can be integrated into the Secure Shell Transport Layer Protocol [RFC4253].

2. Requirements 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 [RFC2119].

3. Applicability Statement

Using AES-GCM to provide both confidentiality and data integrity is generally more efficient than using two separate algorithms to provide these security services.

4. Review of Secure Shell

The goal of secure shell is to establish two secure tunnels between a client and a server, one tunnel carrying client-to-server communications and the other server-to-client communications. Each tunnel is encrypted and a message authentications code is used to insure data integrity.

4.1. Key Exchange

These tunnels are initialized using the secure shell key exchange protocol as described in section 7 of [RFC 4253]. This protocol negotiates a mutually acceptable set of cryptographic algorithms, and produces secret value K and an exchange hash H shared by the client and server. The initial value of H is saved for use as the session_id.

If AES-GCM is selected as the encryption algorithm for a given tunnel, AES-GCM MUST also be selected as the mac algorithm. Conversely, if AES-GCM is selected as the mac algorithm, it MUST also be selected as the encryption algorithm.

As described in section 7.2 of [RFC 4253], a hash based key derivation function (KDF) is applied to the shared secret value K to to generate the required symmetric keys. Each tunnel gets a distinct set of symmetric keys. The keys are generated as shown in figure 1. The sizes of these keys varies depending upon which cryptographic algorithms are being used.

```
   Initial IV
```

```
      Client-to-Sever     HASH( K || H ||"A"|| session_id)
      Server-to-Client    HASH( K || H ||"B"|| session_id)
   Encryption Key
      Client-to-Sever     HASH( K || H ||"C"|| session_id)
      Server-to-Client    HASH( K || H ||"D"|| session_id)
   Integrity Key
      Client-to-Sever     HASH( K || H ||"E"|| session_id)
      Server-to-Client    HASH( K || H ||"F"|| session_id)
```

Figure 1: Key Derivation in Secure Shell

As we shall see below, SSH AES-GCM requires a 12-octet Initial IV and an encryption key of either 16 or 32 octets. Because an AEAD algorithm such as AES-GCM uses the encryption key to provide both confidentiality and data integrity, the integrity key is not used with AES-GCM.

Either the server or client may at any time request that the secure shell session be rekeyed. The shared secret value K, the exchange hash H, and all the above symmetric keys will be updated. Only the session_id will remain unchanged.

4.2. Secure Shell Binary Packets

Upon completion of the key exchange protocol, all further secure shell traffic is parsed into a data structure known as a secure shell binary packet as shown below in Figure 2 (see also section 6 of [RFC 4253]).

```
   uint32    packet_length;  // 0 <= packet_length < 2^32
   byte      padding_length; // 4 <= padding_length < 256
   byte[n1]  payload;        // n1 = packet_length-padding_length-1
   byte[n2]  random_padding; // n2 = padding_length
   byte[m]   mac;            // m  = mac_length
```

Figure 2: Structure of a Secure Shell Binary Packet

Following the usage of [GCM], an AEAD algorithm incorporates the data integrity into the cipher rather than producing a (cipher, mac) pair. But because AES-GCM places the Galois message authentication code (GMAC) at the end of the cipher block, this is logically equivalent to having the mac field at the end of the binary packet as required by RFC 4253.

5. Two New AEAD Algorithms

5.1. aead-aes-128-gcm-ssh

aead-aes-128-gcm-ssh is a variant of the algorithm AEAD_AES_128_GCM specified in section 5.1 of [RFC5116]. The only differences between the two algorithms are in the input and output lengths. Using the notation defined in [RFC5116], the input and output lengths for aead-aes-128-gcm-ssh are as follows:

PARAMETER	Meaning	Value
K_LEN	AES key length	16 octets
P_MAX	maximum plaintext length	2^32 - 256 octets
A_MAX	maximum additional authenticated data length	0 octets
N_MIN	minimum nonce (IV) length	12 octets
N_MAX	maximum nonce (IV) length	12 octets
C_MAX	maximum cipher length	2^32 - 128 octets

Test cases are provided in the appendix of [GCM].

The reader is reminded that due to the presence of length fields and padding in SSH packets, the plaintext length is not the same as the payload length. See section 4.2 above.

5.2. aead-aes-256-gcm-ssh

aead-aes-256-gcm-ssh is a variant of the algorithm AEAD_AES_256_GCM specified in section 5.2 of [RFC5116]. The only differences between the two algorithms are in the input and output lengths. Using the notation defined in [RFC5116], the input and output lengths for aead-aes-256-gcm-ssh are as follows:

PARAMETER	Meaning	Value
K_LEN	AES key length	32 octets

P_MAX	maximum plaintext length	2^32 - 256 octets
A_MAX	maximum additional authenticated data length	0 octets
N_MIN	minimum nonce (IV) length	12 octets
N_MAX	maximum nonce (IV) length	12 octets
C_MAX	maximum cipher length	2^32 -128 octets

Test cases are provided in the appendix of [GCM].

The reader is reminded that due to the presence of length fields and padding in SSH packets, the plaintext length is not the same as the payload length. See section 4.2 above.

6. Nonce and Counter Management

With AES-GCM, the 12 octet Initial Initialization Vector is broken into two fields: an 4 octet fixed field and an 8 octet invocation counter field. The invocation field is treated as a 64-bit integer and is incremented after each invocation of AES-GCM to process a binary packet.

```
uint32  fixed;                  // 4 octets
uint64  invocation_counter;     // 8 octets
```

Figure 3: Structure of an SSH AES-GCM nonce

AES-GCM produces a keystream in blocks of 16-octets which is used to encrypt the plaintext. This keystream is produced by encrypting the following 16-octet data structure:

```
uint32  fixed;                  // 4 octets
uint64  invocation_counter;     // 8 octets
uint32  block_counter;          // 4 octets
```

Figure 4: Structure of an AES input for SSH AES-GCM

The block_counter is initial set to one (1) and incremented as each block of key is produced.

The reader is reminded that SSH requires that the data to be encrypted MUST be padded out to a multiple of the block size

(16-octets for AES-GCM).

RFC 4253 requires that the formation of the mac involve the packet sequence_number, a 32-bit value that counts the number of binary packets that have been sent on a given SSH tunnel. An AEAD algorithm usses a single call to the AEAD encryption algorithm to produce cipher with an embdedded integrity tag:

```
cipher = AEAD_ENCRYPT( nonce, unencrypted_packet ).
```

The presence of the invocation_counter field in the SSH AES-GCM nonce insures that the sequence_number is indeed involved in the formation of the integrity tag, though this involvement differs slightly from the requirements in section 6.4 of RFC 4253.

7. Size of the Message Authentication Code

Both aead-aes-128-gcm-ssh and aead-aes-256-gcm-ssh produce a 16-octet message authentication code. ([RFC5116] calls this an "authentication tag" rather than a "message authentication code".)

8. Security Considerations

The security considerations in [SSH-Arch] apply.

9. IANA Considerations

IANA will add the following two entries to the AEAD Registry described in [RFC5116]:

Name	Reference	Proposed Numeric Identifier
aead-aes-128-gcm-ssh	Section 5.1	5
aead-aes-256-gcm-ssh	Section 5.2	6

IANA will add the following two entries to the Secure Shell Encryption Algorithm name Registry described in [RF4250]:

Name	Reference
aead-aes-128-gcm-ssh	Section 5.1
aead-aes-256-gcm-ssh	Section 5.2

IANA will add the following two entries to the Secure Shell MAC Algorithm name Registry described in [RF4250]:

Name	Reference
aead-aes-128-gcm-ssh	Section 5.1
aead-aes-256-gcm-ssh	Section 5.2

10. References

10.1. Normative References

[GCM] Dworkin, M, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-30D, November 2007.

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

[RFC4250] Lehtinen, S. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Assigned Numbers", RFC 4250, January 2006.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006

[RFC4344] Bellare, M., Kohno, T., and C. Namprempre, "The Secure Shell (SSH) Transport Layer Encryption Modes", RFC 4344, January 2006.

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

Author's Addresses

Kevin M. Igoe
NSA/CSS Commercial Solutions Center
National Security Agency
EMail: kmigoe@nsa.gov

Jerome A. Solinas
National Information Assurance Research Laboratory
National Security Agency
EMail: jasolin@orion.ncsc.mil

Full Copyright Statement

Intellectual Property

Acknowledgement

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