Network Working Group D. McGrew
Internet Draft Cisco Systems, Inc.
Intended Status: Informational K.M. Igoe
Expires: August 20, 2012 National Security Agency
February 17, 2012

AES-GCM and AES-CCM Authenticated Encryption in Secure RTP (SRTP)
draft-ietf-avtcore-srtp-aes-gcm-00

Abstract

This document defines how AES-GCM, AES-CCM, and other Authenticated Encryption with Associated Data (AEAD) algorithms, can be used to provide confidentiality and data authentication mechanisms in the SRTP protocol.

Table of Contents

1. Introduction

The Secure Real-time Transport Protocol (SRTP) is a profile of the Real-time Transport Protocol (RTP), which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the control traffic for RTP, the Real-time Transport Control Protocol (RTCP).

SRTP/SRTCP assumes that both the sender and recipient have a shared secret master key and a shared master salt. As described in sections 4.3.1 and 4.3.3 of [RFC3711], a Key Derivation Function is applied to these values to obtain separate encryption keys, authentication keys and salting keys for SRTP and for SRTCP. (Note: As will be explained below, AEAD SRTP/SRTCP does not make use of these authentication keys.)

Authenticated encryption [BN00] is a form of encryption that, in addition to providing confidentiality for the plaintext that is encrypted, provides a way to check its integrity and authenticity. Authenticated Encryption with Associated Data, or AEAD [R02], adds the ability to check the integrity and authenticity of some Associated Data (AD), also called "additional authenticated data", that is not encrypted. This specification makes use of the interface to a generic AEAD algorithm as defined in [RFC5116].

The Advanced Encryption Standard (AES) is a block cipher that provides a high level of security, and can accept different key sizes. Two families of AEAD algorithm families, AES Galois/Counter Mode (AES-GCM) and AES Counter with Cipher Block Chaining-Message Authentication Code (AES-CCM), are based upon AES. This specification makes use of the AES versions that use 128-bit and 256-bit keys, which we call AES-128 and AES-256, respectively.

The Galois/Counter Mode of operation (GCM) and the Counter with Cipher Block Chaining-Message Authentication Code mode of operation (CCM) are both AEAD modes of operation for block ciphers. Both use counter mode to encrypt the data, an operation that can be efficiently pipelined. Further, GCM authentication uses operations that are particularly well suited to efficient implementation in hardware, making it especially appealing for high-speed implementations, or for implementations in an efficient and compact circuit. CCM is well suited for use in compact software implementations. This specification uses GCM and CCM with both AES-128 and AES-256.

In summary, this document defines how to use AEAD algorithms, particularly AES-GCM and AES-CCM, to provide confidentiality and message authentication within SRTP and SRTCP packets.

1.1. Conventions Used In This Document

The following terms have very specific meanings in the context of this RFC:

Crypto Context For the purposes of this document a crypto context is the outcome of any process which results in authentication of each participant in the SRTP session and possession by each partcipant of a shared secret master key and a shared master salt. Details of how the maser key and master salt are established are outside the scope of this document. Similarly any mechanism for rekeying an existing Ciper Contest is outside the scope of the document. The master key MUST be at least as large as the encryption key. The SRTP/SRTCP Key

Derivation Function (KDF) defined in [RFC3711] is applied to the master key and master SALT to derive the SRTP_encr_key, SRTCP_encr_key, SRTP_SALT, and SRTCP_SALT. Authentication keys are not used in AEAD.

Instantiation: Once keys have been established, an instance of the AEAD algorithm is created using the appropriate key and salt. In a point-to-point scenario, each participant in the SRTP/SRTCP session will need four instantiations of the AEAD algorithm; one for inbound SRTP traffic, one for outbound SRTP traffic source, one for inbound SRTCP traffic, and one for outbound SRTCP traffic source. See section 1.2 for details on what is required of each instantiation.

Invocation: SRTP/SRTCP data streams are broken into packets. Each packet is processed by a single invocation of the appropriate instantiation of the AEAD algorithm.

Each AEAD instantiation has its own key, a 48-bit zero-based packet counter that is incremented after that particular instantiation has been invoked to process an SRTP packet, and a 31-bit zero-based SRTCP index that is incremented each time an SRTCP packet is processed. A 32-bit Block Counter is incremented each time a block of key is produced and is reset (to zero for CCM and to one for GCM) at the start of each packet. As we shall see in sections 1.3 and 1.4, the packet counter and SRTCP counter play a crucial role in the formation of each packet's IV.

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].

1.2. AEAD processing for SRTP

We first define how to use a generic AEAD algorithm in SRTP, then we describe the specific use of the AES-128-GCM and AES-256-GCM algorithms.

The use of an AEAD algorithm is defined by expressing the AEAD encryption algorithm inputs in terms of SRTP fields and data structures. The AEAD encryption inputs are as follows:

Key: This input is the SRTP encryption key (SRTP_encr_key) produced from the shared secret master key using the key derivation process. (Note that the SRTP_auth_key is not used).

Associated Data — This is data that is to be authenticated but not encrypted. In SRTP, the associated data consists of the entire RTP header, including the list of CSRC identifiers (if present) and the RTP header extension (if present), as shown in Figure 3.

Plaintext — Data that is to be both encrypted and authenticated. In SRTP this consists of the RTP payload, the RTP padding and the RTP pad count fields (if the latter two fields are present) as shown in Figure 3. The padding service provided by RTP is not needed by the AEAD encryption algorithm, so the RTP padding and RTP pad count fields SHOULD be omitted.

Initialization Vector — Each SRTP/SRTCP packet has its own 12-octet initialization vector (IV). Construction of this IV is covered in more detail below.

The AEAD encryption algorithm accepts these four inputs and returns a Ciphertext field.

1.2.1. AEAD versus SRTP/SRTCP Authentication

The reader is reminded that in addition to providing confidentiality for the plaintext that is encrypted, an AEAD algorithm also provides a mechanism that allows the intended recipient to check the data integrity and authenticity of the plaintext and associated data. The AEAD authentication tag is incorporated into the Ciphertext field by RFC 5116, thus AEAD does not make use of the SRTP/SRTCP Authentication Tag fields defined in RFC 3711. (Note that this means that the cipher text will be longer than the plain text by precisely the length of the AEAD authentication tag.)

The AEAD message authentication mechanism MUST be the primary message authentication mechanism for AEAD SRTP/SRTCP. Additional SRTP/SRTCP authentication mechanisms SHOULD NOT be used with any AEAD algorithm and the optional SRTP/SRTCP Authentication Tags are NOT RECOMMENDED and SHOULD NOT be present. Note that this contradicts section 3.4 of [RFC3711] which makes the use of the SRTCP Authentication field mandatory, but the presence of the AEAD authentication renders the older authentication methods redundant.

Rationale. Some applications use the SRTP/SRTCP Authentication

Tag as a means of conveying additional information, notably [RFC4771]. This document retains the Authentication Tag field primarily to preserve compatibility with these applications.

1.2.2. Values used to form the Initialization Vector (IV)

The initialization vector for an SRTP packet is formed from the:

- **SSRC** — The 4-octet Synchronization Source identifier (SSRC), found in the RTP header.

- **Packet Counter** — Each AEAD instantiation MUST maintain a 6 octet zero-based packet counter which is incremented each time an SRTP packet is sent. As we shall see below, the packet counter is used to insure each SRTP packet gets a unique initialization vector.

- **Sequence Number** — The 2-octet RTP Sequence Number (SEQ), found in the RTP header. The SEQ is just the 16 least significant bits of the packet counter.

- **Rollover Counter** — A 4-octet Rollover Counter (ROC), maintained independently by both sides of the link, incremented each time the Sequence Number cycles back to 0. The ROC is just the 32 most significant bits of the Packet Counter.

- **SRTCP index** — The SRTCP index is a 31-bit counter that plays the same role for SRTCP packets that the Packet Counter does for SRTP packets. Unlike the Packet Counter, the SRTCP index is explicitly included in each STRCP packet. The sender MUST increment the SRTCP index by one after each SRTP packet is sent.

- **SALT** — A 12-octet SRTP session encryption salt produced by the SRTP Key Derivation Function (KDF) (see section 2.4).

The reader is reminded that both SRTP and SRTCP allow packets to arrive out of order, presenting the receiver with a synchronization problem. The 31-bit SRTCP index is contained in the unencrypted (but authenticated) portion of the SRTCP header, allowing the recipient to read the SRTCP index directly from the header. But only the low 16 bits of the SRTP Packet counter are contained in the SRTP header (in the sequence number field). Section 3.3.1 of [RFC3711] explains in great detail how the 16-bit sequence number and 32-bit Rollover Counter are to be used to recover the 48-bit Packet Counter.

1.3. SRTP IV formation for AES-GCM and AES-CCM

AES-GCM and AES-CCM SRTP use a 12 byte initialization vector which is formed as follows. A 12-octet string is formed by concatenating 2-octets of zeroes, the 4-octet SSRC, and the 6-octet Packet Counter. The resulting string is bitwise exclusive-ored with the 12-octet salt to form the 12-octet IV.

```
                0  0  0  0  0  0  0  0  0  0  1  1
                0  1  2  3  4  5  6  7  8  9  0  1
              +--+--+--+--+--+--+--+--+--+--+--+--+
              |00|00|    SSRC   |  Packet_Counter |---+
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                      |
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
              |         Encryption Salt           |->(+)
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                      |
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
              |       Initialization Vector       |<--+
              +--+--+--+--+--+--+--+--+--+--+--+--+
```

Figure 1: AES-GCM and AES-CCM SRTP Initialization Vector formation.

Using the terminology of section 8.2.1. of [GCM], the first six octets of the IV are the fixed field and the last six bytes are the invocation field.

1.4. SRTCP IV formation for AES-GCM and AES-CCM

The initialization vector for an SRTCP packet is formed from the 4-octet Synchronization Source identifier (SSRC), 31-bit SRTCP Index (packed zero-filled, right justified into a 4-octet field), and a 12-octet SRTCP session encryption salt produced by the SRTP Key Derivation Function (KDF).

```
                0  1  2  3  4  5  6  7  8  9 10 11
              +--+--+--+--+--+--+--+--+--+--+--+--+
              |00|00|    SSRC   |00|00|SRTCP Index|---+
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                      |
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
              |         Encryption Salt           |->(+)
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                      |
              +--+--+--+--+--+--+--+--+--+--+--+--+   |
              |       Initialization Vector       |<--+
              +--+--+--+--+--+--+--+--+--+--+--+--+
```

Figure 2: SRTCP Initialization Vector formation.

As shown if figure 2, a 12-octet string is formed by concatenating in order 2-octets of zeroes, the 4-octet SSRC, 2 more zero octets, and the 4-octet SRTCP index. The resulting 12-octet string is bitwise exclusive-ored into salt; the output of that process is the IV. The IV is always exactly 12 octets in length. Using the terminology of section 8.2.1. of [GCM], the first eight octets of the IV are the fixed field and the last four bytes are the invocation field.

1.5. AEAD Processing of SRTP Packets

All SRTP packets MUST be authenticated and encrypted. Figure 3 below shows which fields of AEAD SRTP packet are to be treated as plaintext and which are to be treated as additional authenticated data.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |V=2|P|X|  CC   |M| Packet Type |       sequence number         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                           timestamp                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |           synchronization source (SSRC) identifier            |
    +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 A  |      contributing source (CSRC) identifiers (optional)        |
 A  |                               ....                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                    RTP extension (OPTIONAL)                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P  |                          payload  ...                         |
 P  |                               +-------------------------------+
 P  |                               | RTP padding   | RTP pad count |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 X  :              authentication tag (NOT RECOMMENDED)             :
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

            P = Plaintext (to be encrypted and authenticated)
            A = Associated Data (to be authenticated only)
            X = neither encrypted nor authenticated

     Note: The RTP padding and RP padding count fields are optional
           and are not recommended
```

Figure 3: AEAD inputs from an SRTP packet.

1.6. AEAD Processing of SRTCP Packets

All SRTCP packets MUST be authenticated, but unlike SRTP, SRTCP packet encryption is optional. A sender can select which packets to encrypt, and indicates this choice with a 1-bit encryption flag (located in the leftmost bit of the 32-bit word that contains the

SRTCP index)

1.6.1. Encrypted SRTCP packets

When the encryption flag is set to 1, the first 8-octets, the encryption flag and 31-bit SRTCP index MUST be treated as AAD. The remaining data MUST be treated as plaintext, and hence is to be both encrypted and AEAD authenticated, save for the optional SRTCP MKI index and optional SRTCP authentication tag, which MUST be neither encrypted nor AEAD authenticated. Figure 4 below shows how fields of an RTCP packet are to be treated when the encryption flag is set to 1.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A   |V=2|P|   RC    |  Packet Type  |             length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A   |           synchronization source (SSRC) of Sender             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                         sender info                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                       report block 1                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                       report block 2                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                              ...                              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |V=2|P|   SC    |  Packet Type  |               length          |
     +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 P   |                          SSRC/CSRC_1                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                           SDES items                          |
     +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 P   |                              ...                              |
     +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 A   |1|                         SRTCP index                         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 X   |                  SRTCP MKI (optional)index                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 X   :              authentication tag (NOT RECOMMENDED)             :
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                P = Plaintext (to be encrypted and authenticated)
                A = Associated Data (to be authenticated only)
                X = neither encrypted nor authenticated
```

Figure 4: AEAD SRTCP inputs when encryption flag = 1.

1.6.2. Unencrypted SRTCP packets

When the encryption flag is set to 0, all of the data up to and including the SRTCP index is treated as AAD. Figure 5 shows how the fields of an RTCP packet are to be treated when the encryption flag is set to 0.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |V=2|P|   RC    |  Packet Type  |            length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |           synchronization source (SSRC) of Sender             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                         sender info                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                        report block 1                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                        report block 2                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                              ...                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |V=2|P|   SC    |  Packet Type  |              length           |
    +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 A  |                          SSRC/CSRC_1                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 A  |                           SDES items                          |
    +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 A  |                              ...                              |
    +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 A  |0|                         SRTCP index                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 X  |                  SRTCP MKI (optional)index                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 X  :               authentication tag (NOT RECOMMENDED)            :
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

             A = Associated Data (to be authenticated only)
             X = neither encrypted nor authenticated
```

Figure 5: AEAD SRTCP inputs when encryption flag = 0.

1.7. Initialization of the Counters

When an AEAD Crypto Context is first established, both the SRTCP index and the rollover counter are set to zero. The Sequence Number is set to a value passed to it by RTP. When the context is rekeyed these counters keep their current values and are not reset to zero. These conventions assist in making a seamless transition from the old key (if any) to the new key despite of the fact that packets are allowed to arrive out of order.

As mentioned in section 1.1, AES-GCM and AES-CCM both use a Block Counter which is reset at the start of each packet. For AES-CCM it is reset to 0 and for AES-GCM it is reset to 1.

1.8. Prevention of IV Reuse

For a given key it is critical that the IV not repeat. This reduces to the problem of insuring neither the Packet Counter nor the SRTCP index do not repeat before the AEAD instantiation is rekeyed.

Processing MUST cease if either the 48-bit Packet Counter or the 31-bit SRTCP index cycles back to their initial value. Processing MUST NOT resume until a new SRTP/SRTCP session has been established using a new shared secret master key and shares master salt. Ideally, a rekey should be done well before either of these counters cycle.

2. AEAD parameters for SRTP and SRTCP

In general, any AEAD algorithm can accept inputs with varying lengths, but each algorithm can accept only a limited range of lengths for a specific parameter. In this section, we describe the constraints on the parameter lengths that any AEAD algorithm must support to be used in AEAD-SRTP. Additionally we specify a complete parameter set for two specific AEAD algorithms, namely AES-GCM and AES-CCM.

2.1. Generic AEAD Parameter Constraints

All AEAD algorithms used with SRTP/SRTCP MUST satisfy the three constraints listed below:

PARAMETER	Meaning	Value
A_MAX	maximum additional authenticated data length	MUST be at least 12 octets
N_MIN	minimum nonce (IV) length	MUST be no more than 12 octets
N_MAX	maximum nonce (IV) length	MUST be at least 12 octets
C_MAX	maximum ciphertext length per invocation	MUST be at most 2^16-40 octets SHOULD be at least 2232

The upper bound on C_MAX is obtained by subtracting away a 20-octet IP header, an 8-octet UDP header, and a 12-octet RTP header out of the largest possible IP packet, the total length of which is 2^16 octets.

Similarly the lower bound on C_MAX is based on the maximum transmission unit (MTU) of 2272 octets in IEEE 802.11. Because many RTP applications use very short payloads (for example, the G.729 codec used in VoIP can be as short as 20 octets), implementations that only support a maximum ciphertext length smaller than 2232 octets are permitted under this RFC. However, in the interest of maximizing interoperability between various AEAD implementations, the use of C_MAX values less than 2232 is discouraged.

For sake of clarity we specify two additional parameters:

Authentication Tag Length	MUST be either 8, 12, or 16 octets
Maximum number of invocations for a given instantiation	MUST be at most 2^48 for SRTP MUST be at most 2^31 for SRTCP

The reader is reminded that the plaintext is shorter than the ciphertext by exactly the length of the AEAD authentication tag.

2.2. AES-GCM for SRTP/SRTCP

AES-GCM is a family of AEAD algorithms built around the AES block cipher algorithm. AES-GCM uses AES counter mode for encryption and Galois Message Authentication Code (GMAC) for authentication. A detailed description of the AES-GCM family can be found in [RFC5116]. The following members of the AES-GCM family may be used with SRTP/SRTCP:

Table 1: AES-GCM algorithms for SRTP/SRTCP

Name	Key Size	Auth. Tag Size	Reference
AEAD_AES_128_GCM	16 octets	16 octets	[RFC5116]
AEAD_AES_256_GCM	32 octets	16 octets	[RFC5116]
AEAD_AES_128_GCM_8	16 octets	8 octets	[RFC5282]
AEAD_AES_256_GCM_8	32 octets	8 octets	[RFC5282]
AEAD_AES_128_GCM_12	16 octets	12 octets	[RFC5282]
AEAD_AES_256_GCM_12	32 octets	12 octets	[RFC5282]

Any implementation of AES-GCM SRTP SHOULD support both AEAD_AES_128_GCM_8 and AEAD_AES_256_GCM_8, and it MAY support the four other variants shown in the table.

In addition to the Packet Counter used in the formation of IVs, each instantiation of AES-GCM has a block counter which is incremented each time AES is called to produce a 16-octet output block. The block counter is reset to "1" each time AES-GCM is invoked to process a new packet. The 128-bit concatentation of the IV and the block counter is input to AES and the output is used as a block of key that is XORed to the next block of data to be encrypted/decypted.

```
                                              1   1   1   1   1   1
       0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
     ----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                   initialization                |     block     |
    |                       vector                    |    counter    |
     ----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 6: AES Inputs for Counter Mode Encryption

2.3. AES-CCM for SRTP/SRTCP

AES-CCM is another family of AEAD algorithms built around the AES block cipher algorithm. AES-GCM uses AES counter mode for encryption and AES Cipher Block Chaining Message Authentication Code (CBC MAC) for authentication. A detailed description of the AES-CCM family can be found in [RFC5116]. The following members of the AES-CCM family may be used with SRTP/SRTCP:

Table 2: AES-CCM algorithms for SRTP/SRTCP

Name	Key Size	Auth. Tag Size	Reference
AEAD_AES_128_CCM	16 octets	16 octets	[RFC5116]
AEAD_AES_256_CCM	32 octets	16 octets	[RFC5116]

Any implementation of AES-CCM SRTP/SRTCP SHOULD support both AEAD_AES_128_CCM and AEAD_AES_256_CCM.

In addition to the Packet Counter used in the formation of IVs, each instantiation of AES-CCM has a block counter which is incremented each time AES is called to produce a 16-octet output block. The block counter is reset to "0" each time AES-CCM is invoked to process a new packet. As with AES-GCM, the 128-bit concatentation of the IV abd the block counter is input to AES to produce a block of key that is XORed to the next block of data to be encrypted/decypted.

AES-CCM uses a flag octet that conveys information about the length of the authentication tag, length of the block counter, and presence of additional authenticated data. For AES-CCM in SRTP/SRTCP, the flag octet has the hex value 5A if an 8-octet authentication tag is used, 6A if a 12-octet authentication tag is used, and 7A if a 16-octet authentication tag is used. The flag octet is one of the inputs to AES during the counter mode encryption of the plaintext.

2.4. Key Derivation Functions

A Key Derivation Function (KDF) is used to derive all of the required encryption and authentication keys from a secret value shared by the two endpoints. Both the AEAD_AES_128_GCM algorithms and the AEAD_AES_128_CCM algorithms MUST use the (128-bit) AES_CM_PRF Key Derivation Function described in [RFC3711]. Both the AEAD_AES_256_GCM algorithms and the AEAD_AES_256_CCM algorithms MUST use the AES_256_CM_PRF Key Derivation Function described in [RFC 6188].

3. Security Considerations

3.1. Handling of Security Critical Parameters

As with any security process, the implementer must take care to ensure cryptographically sensitive parameters are properly handled. Many of these recommendations hold for all SRTP cryptographic algorithms, but we include them here to emphasize their importance.

- If the master salt is to be kept secret it MUST be properly erased when no longer needed.
- The secret master key and all keys derived from it MUST be kept secret. All keys MUST be properly erased when no longer needed.
- Packets that fail the authentication check SHOULD be silently discarded.
- The sender MUST increment the Packet Counter after each SRTP packet is processed.
- The sender MUST increment the SRTCP index after each SRTCP packet is processed.
- At the start of each packet the block counter MUST be reset (to 0 for CCM, to 1 for GCM). The block counter is incremented after each block key has been produced, but it MUST NOT be allowed to exceed 2^32-1.
- Each time a rekey occurs the initial values of the invocation counter and SRTCP index MUST be saved.
- Processing MUST cease if the 48-bit Packet Counter or the 31-bit SRTCP index cycles back to its initial value. Processing MUST NOT resume until a new SRTP/SRTCP session has been established using a new SRTP master key. Ideally, a rekey should be done well before either of these counters cycle.

3.2. Size of the Authentication Tag

We require that the AEAD authentication tag must be at least 8 octets, significantly reducing the probability of an adversary successfully introducing fraudulent data. The goal of an authentication tag is to minimize the probability of a successful forgery occurring anywhere in the network we are attempting to defend. There are three relevant factors: how low we wish the probability of successful forgery to be (prob_success), how many attempts the adversary can make (N_tries) and the size of the authentication tag in bits (N_tag_bits). Then

```
prob_success < expected number of successes
             = N_tries * 2^-N_tag_bits.
```

Suppose an adversary wishes to introduce a forged or altered packet into a target network by randomly selecting an authentication value until by chance they hit a valid authentication tag. The table below summarizes the relationship between the number of forged packets the adversary has tried, the size of the authentication tag, and the probability of a compromise occurring (i.e. at least one of the attempted forgeries having a valid authentication tag). The reader is reminded that the forgery attempts can be made over the entire network, not just a single link, and that frequently changing the key does not decrease the probability of a compromise occurring.

Authentication Tag Size (octets)	Probability of a Compromise Occurring		
	2^-30	2^-20	2^-10
4	2^2 tries	2^12 tries	2^22 tries
8	2^34 tries	2^44 tries	2^54 tries
12	2^66 tries	2^76 tries	2^86 tries
16	2^98 tries	2^108 tries	2^118 tries

Table 1: Probability of a compromise occurring for a given number of forgery attempts and tag size.

4. IANA Considerations

RFC 4568 defines SRTP "crypto suites"; a crypto suite corresponds to a particular AEAD algorithm in SRTP. In order to allow SDP to signal the use of the algorithms defined in this document, IANA will register the following crypto suites into the subregistry for SRTP crypto suites under the SRTP transport of the SDP Security Descriptions:

```
srtp-crypto-suite-ext = "AEAD_AES_128_GCM"    /
                        "AEAD_AES_256_GCM"    /
                        "AEAD_AES_128_GCM_8"  /
                        "AEAD_AES_256_GCM_8"  /
                        "AEAD_AES_128_GCM_12" /
```

```
                             "AEAD_AES_256_GCM_12" /
                             "AEAD_AES_128_CCM"    /
                             "AEAD_AES_256_CCM"    /
```

srtp-crypto-suite-ext

DTLS-SRTP [RFC5764] defines a DTLS-SRTP "SRTP Protection Profile"; it also corresponds to the use of an AEAD algorithm in SRTP. In order to allow the use of the algorithms defined in this document in DTLS-SRTP, IANA will also register the following SRTP Protection Profiles:

```
SRTP_AEAD_AES_128_GCM
SRTP_AEAD_AES_256_GCM
SRTP_AEAD_AES_128_GCM_8
SRTP_AEAD_AES_256_GCM_8
SRTP_AEAD_AES_128_GCM_12
SRTP_AEAD_AES_256_GCM_12
SRTP_AEAD_AES_128_CCM
SRTP_AEAD_AES_256_CCM
```

5. Acknowledgements

The authors would like to thank Michael Peck, Michael Torla, Qin Wu, and many other reviewers who provided valuable comments on earlier drafts of this document.

6. References

6.1. Normative References

[CCM] Dworkin, M., "NIST Special Publication 800-38C: The CCM Mode for Authentication and Confidentiality", U.S. National Institute of Standards and Technology http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C.pdf.

[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/SP800-38D.pdf.

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

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4658] Andreasen, F., Baugher, M., and D.Wing, "Session Description Protocol (SDP): Security Descriptions for Media Streams", RFC 4568, July 2006.

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

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

[RFC5282] McGrew, D. and D. Black, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", RFC 5282, August 2008.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[RFC6188] McGrew,D.,"The Use of AES-192 and AES-256 in Secure RTP" RFC 6811, March 2011

6.2. Informative References

[BN00] Bellare, M. and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm", Proceedings of ASIACRYPT 2000, Springer-Verlag, LNCS 1976, pp. 531-545 http://www-cse.ucsd.edu/users/mihir/papers/oem.html.

[BOYD] Boyd, C. and A. Mathuria, "Protocols for Authentication and Key Establishment", Springer, 2003 .

[CMAC] "NIST Special Publication 800-38B", http://csrc.nist.gov/CryptoToolkit/modes/800-38_Series_Publications/SP800-38B.pdf.

[EEM04] Bellare, M., Namprempre, C., and T. Kohno, "Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm", ACM Transactions on Information and System Secu rity, http://www-cse.ucsd.edu/users/tkohno/papers/TISSEC04/.

[GR05] Garfinkel, T. and M. Rosenblum, "When Virtual is Harder than Real: Security Challenges in Virtual Machine Based Computing Environments", Proceedings of the 10th Workshop on Hot Topics in Operating Systems http://www.stanford.edu/~talg/papers/HOTOS05/virtual-harder-hotos05.pdf.

[J02] Jonsson, J., "On the Security of CTR + CBC-MAC", Proceedings of the 9th Annual Workshop on Selected Areas on Cryptography, http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/ccm/ccm-ad1.pdf, 2002.

[MODES] Dworkin, M., "NIST Special Publication 800-38: Recommendation for Block Cipher Modes of Operation", U.S. National Institute of Standards and Technology http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

[MV04] McGrew, D. and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM)", Proceedings of INDOCRYPT '04, http://eprint.iacr.org/2004/193, December 2004.

[R02] Rogaway, P., "Authenticated encryption with Associated-Data", ACM Conference on Computer and Communication Security (CCS'02), pp. 98-107, ACM Press, 2002. http://www.cs.ucdavis.edu/~rogaway/papers/ad.html.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, December 2005.

[RFC4771] Lehtovirta, V., Naslund, M., and K. Norrman, "Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)", RFC 4771, January 2007.

Author's Address

David A. McGrew
Cisco Systems, Inc.
510 McCarthy Blvd.
Milpitas, CA 95035
US
Phone: (408) 525 8651
Email: mcgrew@cisco.com
URI: http://www.mindspring.com/~dmcgrew/dam.htm

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

Acknowledgement

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