Network Working Group D. McGrew
Internet Draft Cisco Systems, Inc.
Intended Status: Standards Track K. Igoe
Expires: October 16, 2015 National Security Agency
April 14, 2015

AES-GCM Authenticated Encryption in Secure RTP (SRTP)
draft-ietf-avtcore-srtp-aes-gcm-15

Status of this Memo

Copyright Notice

Abstract

This document defines how the AES-GCM Authenticated Encryption with Associated Data family of algorithms can be used to provide confidentiality and data authentication in the SRTP protocol. Note: this is an intermediate draft, awaiting the inclusion of test vectors. Care is being taken to ensure these test vectors will be correct, always a desirable property.

Table of Contents

1. Introduction

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of the Real-time Transport Protocol (RTP) [RFC3550], 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). It is important to note that the outgoing SRTP packets from a single endpoint may be originating from several independent data sources.

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. AES Galois/Counter Mode (AES-GCM) [GCM] is a family of AEAD algorithms 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.

Any AEAD algorithm provides an intrinsic authentication tag. In many applications the authentication tag is truncated to less than full length. In this specification the authentication tag MUST be either 8 octets or 16 octets in length, and the 8 byte authentication tag can only be used with AES-128. Thus when used in SRTP, GCM will have three configurations:

AEAD_AES_128_GCM_8	AES-128 with an 8 byte authentication tag
AEAD_AES_128_GCM	AES-128 with a 16 byte authentication tag
AEAD_AES_256_GCM	AES-256 with a 16 byte authentication tag

The key size and the length of the authentication tag are set when the session is initiated and SHOULD NOT be altered.

The Galois/Counter Mode of operation (GCM) ia an AEAD mode of operation for block ciphers. GCM 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.

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

2. Conventions Used In This Document

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 [RFC2119].

3. Overview of the SRTP/SRTCP AEAD security Architecture

SRTP/SRTCP AEAD security is based upon the following principles:

a) Both privacy and authentication are based upon the use of symmetric algorithms. An AEAD algorithm such as AES-GCM combines privacy and authentication into a single process.

b) A secret master key is shared by all participating endpoints, both those originating SRTP/SRTCP packets and those receiving these packets. Any given master key MAY be used simultaneously by several endpoints to originate SRTP/SRTCP packets (as well one or more endpoints using this master key to process inbound data).

c) A Key Derivation Function is applied to the shared master key value to form separate encryption keys, authentication keys and salting keys for SRTP and for SRTCP (a total of six keys). This process is described in section 4.3 of [RFC3711]. The master key MUST be at least as large as the encryption key derived from it. Since AEAD algorithms such as AES-GCM combine encryption and authentication into a single process, AEAD algorithms do not make use of separate authentication keys.

d) Aside from making modifications to IANA registries to allow AES-GCM to work with SDES, DTLS-SRTP and MIKEY, the details of how the master key is established and shared between the participants are outside the scope of this document. Similarly any mechanism for rekeying an existing session is outside the scope of the document.

e) Each time an instantiation of AES-GCM is invoked to encrypt and authenticate an SRTP or SRTCP data packet a new IV is used. SRTP combines the 4-octet synchronization source (SSRC) identifier, the 4-octet rollover counter (ROC), and the 2-octet sequence number (SEQ) with the 12-octet encryption salt to form a 12-octet IV (see section 8.1). SRTCP combines the SSRC and 31-bit SRTCP index with the encryption salt to form a 12-octet IV (see section 9.1).

4. Terminology

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

Instantiation: In AEAD, an instantiation is an (Encryption_key, salt) pair together with all of the data structures (for example, counters) needed for it to function properly. In SRTP/SRTCP, each endpoint will need two instantiations of the AEAD algorithm for each master key in its possession, one instantiation for SRTP traffic and one instantiation for SRTCP traffic.

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.

In many applications, each endpoint will have one master key for processing outbound data but may have one or more separate master keys for processing inbound data.

5. Generic AEAD Processing

5.1. Types of Input Data

Associated Data: This is data that is to be authenticated but not encrypted.

Plaintext: Data that is to be both encrypted and authenticated.

Raw Data: Data that is to be neither encrypted nor authenticated.

Which portions of SRTP/SRTCP packets that are to be treated as associated data, which are to be treated as plaintext, and which are to be treated as raw data are covered in sections 8.2, 9.2 and 9.3.

5.2. AEAD Invocation Inputs and Outputs

5.2.1. Encrypt Mode

Inputs:

Encryption_key	Octet string, either 16 or 32 octets long

Initialization_Vector	Octet string, 12 octets long
Associated_Data	Octet string of variable length
Plaintext	Octet string of variable length

Outputs

Ciphertext*	Octet string, length = length(Plaintext)+tag_length

(*): In AEAD the authentication tag in embedded in the cipher text. When GCM is being used the ciphertext consists of the encrypted plain text followed by the authentication tag.

5.2.2. Decrypt Mode

Inputs:

Encryption_key	Octet string, either 16 or 32 octets long
Initialization_Vector	Octet string, 12 octets long
Associated_Data	Octet string of variable length
Ciphertext	Octet string of variable length

Outputs

Plaintext	Octet string, length = length(Ciphertext)-tag_length
Validity_Flag	Boolean, TRUE if valid, FALSE otherwise

5.3. Handling of AEAD Authentication

AEAD requires that all incoming packets MUST pass AEAD authentication before any other action takes place. Plaintext and associated data MUST NOT be released until the AEAD authentication tag has been validated. Further the ciphertext MUST NOT be decrypted until the AEAD tag has been validated.

Should the AEAD tag prove to be invalid, the packet in question is to be discarded and a Validation Error flag raised. Local policy determines how this flag is to be handled and is outside the scope of this document.

6. Counter Mode Encryption

Each outbound packet uses a 12-octet IV and an encryption key to form two outputs, a 16-octet first_key_block which is used in forming the authentication tag and a key stream of octets, formed in blocks of 16-octets each. The first 16-octet block of key is saved for use in forming the authentication tag, and the of remainder of the key stream is XORed to the plaintext to form cipher. This key stream is formed one block at a time by inputting the concatenation of a 12-octet IV (see sections 8.1 and 9.1) with a 4-octet block to AES. The pseudo-code below illustrates this process:

```
def GCM_keystream( Plaintext_len, IV, Encryption_key ):
    assert Plaintext_len  <= (2**36) - 32 ## measured in octets
    key_stream = ""
    block_counter = 1
    first_key_block = AES_ENC( data=IV||block_counter,
                               key=Encryption_key        )
    while len(key_stream) < Plaintext_len:
        block_counter = block_counter + 1
        key_block = AES_ENC( data=IV||block_counter,
                             key=Encryption_key        )
        key_stream  = key_stream || key_block
    key_stream = truncate( key_stream, Plaintext_len )
    return (first_key_block, key_stream )
```

In theory this keystream generation process allows for the encryption of up to (2^36)-32 octets per invocation (i.e. per packet), far longer than is actually required.

With any counter mode, if the same (IV, Encryption_key) pair is used twice, precisely the same keystream is formed. As explained in section 9.1 of RFC 3711, this is a cryptographic disaster. For GCM the consequences are even worse since such a reuse compromises GCM's integrity mechanism not only for the current packet stream but for all future uses of the current encryption_key.

7. Unneeded SRTP/SRTCP Fields

AEAD counter mode encryption removes the need for certain existing SRTP/SRTCP mechanisms.

7.1. SRTP/SRTCP Authentication Field

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.

7.2. RTP Padding

AES-GCM does not requires that the data be padded out to a specific block size, reducing the need to use the padding mechanism provided by RTP. It is RECOMMENDED that the RTP padding mechanism not be used unless it is necessary to disguise the length of the underlying plaintext.

8. AES-GCM processing for SRTP

8.1. SRTP IV formation for AES-GCM

```
               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   |     ROC   | SEQ |---+
             +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                     |
             +--+--+--+--+--+--+--+--+--+--+--+--+   |
             |         Encryption Salt           |->(+)
             +--+--+--+--+--+--+--+--+--+--+--+--+   |
                                                     |
             +--+--+--+--+--+--+--+--+--+--+--+--+   |
             |       Initialization Vector       |<--+
             +--+--+--+--+--+--+--+--+--+--+--+--+
```

Figure 1: AES-GCM SRTP Initialization
Vector formation.

The 12 octet initialization vector used by AES-GCM SRTP is formed by first concatenating 2-octets of zeroes, the 4-octet SSRC, the 4-octet Rollover Counter (ROC) and the two octet sequence number SEQ. The resulting 12-octet value is then XORed to the 12-octet salt to form the 12-octet IV.

8.2. Data Types in SRTP Packets

All SRTP packets MUST be both authenticated and encrypted. The data fields within the SRTP packets are broken into Associated Data, Plaintext and Raw Data as follows (see Figure 2):

Associated Data: The version V (2 bits), padding flag P (1 bit),

extension flag X (1 bit), CSRC count CC (4 bits),
marker M (1 bit), the Payload Type PT (7 bits),
the sequence number (16 bits), timestamp (32

bits), SSRC (32 bits), optional contributing source identifiers (CSRCs, 32 bits each), and optional RTP extension (variable length).

Plaintext: The RTP payload (variable length), RTP padding (if used, variable length), and RTP pad count (if used, 1 octet).

Raw Data: The optional variable length SRTP MKI and SRTP authentication tag (whose use is NOT RECOMMENDED). These fields are appended after encryption has been performed.

```
        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|     PT      |       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 |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

              P = Plaintext (to be encrypted and authenticated)
              A = Associated Data (to be authenticated only)
```

Figure 2: Structure of an SRTP packet before Authenticated Encryption

Since the AEAD ciphertext is larger than the plaintext by exactly the length of the AEAD authentication tag, the corresponding SRTP encrypted packet replaces the plaintext field by a slightly larger field containing the cipher. Even if the plaintext field is empty, AEAD encryption must still be performed, with the resulting cipher consisting solely of the authentication tag. This tag is to be placed immediately before the optional variable length SRTP MKI and SRTP authentication tag fields.

```
        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|     PT      |       sequence number         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    A  |                           timestamp                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    A  |           synchronization source (SSRC) identifier            |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
    A  |      contributing source (CSRC) identifiers (optional)        |
    A  |                               ....                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    A  |                   RTP extension (OPTIONAL)                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    C  |                             cipher                            |
    C  |                               ...                             |
    C  |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    R  :                     SRTP MKI (OPTIONAL)                       :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    R  :           SRTP authentication tag (NOT RECOMMENDED)           :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                C = Ciphertext (encrypted and authenticated)
                A = Associated Data (authenticated only)
                R = neither encrypted nor authenticated, added
                    after authenticated encryption completed
```

Figure 3: Structure of an SRTP packet after Authenticated Encryption

8.3. Handling Header Extensions

RTP header extensions were first defined in RFC 3550. RFC 6904 [RFC6904] describes how these header extensions are to be encrypted in SRTP.

When RFC 6904 is in use, a separate keystream is generated to encrypt selected RTP header extension elements. For the AEAD_AES_128_GCM and AEAD_AES_128_GCM_8 algorithms, this keystream MUST be generated in the manner defined in [RFC6904] using the AES_128_CM transform. For the AEAD_AES_256_GCM algorithm, the keystream MUST be generated in the manner defined for the AES_256_CM transform. The originator must perform any required header extension encryption before the AEAD algorithm is invoked.

As with the other fields contained within the RTP header, both encrypted and unencrypted header extensions are to be treated by the AEAD algorithm as Associated Data (AD). Thus the AEAD algorithm does not provide any additional privacy for the header extensions, but does provide integrity and authentication.

8.4. Prevention of SRTP IV Reuse

In order to prevent IV reuse, we must ensure that the (ROC,SEQ,SSRC) triple is never used twice with the same master key. There are two phases to this issue.

Counter Management: A rekey MUST be performed to establish a new master key before the (ROC,SEQ) pair cycles back to its original value. Note that implicitly assumes that either the outgoing RTP process is trusted to not attempt to repeat a (ROC,SEQ) value, or that the encryption process ensures that the both the SEQ and ROC numbers of the packets presented to it are always incremented in the proper fashion. This is particularly important for GCM since using the same (ROC,SEQ) value twice compromises the authentication mechanism. For GCM, the (ROC,SEQ) and SSRC values used MUST either be generated or checked by the SRTP implementation, or by a module (e.g. the RTP application) that can be considered equally trusted as the SRTP implementation. While [RFC3711] allows detecting SSRC collisions after they happen, SRTP using GCM with shared master keys MUST prevent SSRC collision from happening even once.

SSRC Management: For a given master key, the set of all SSRC values used with that master key must be partitioned into disjoint pools, one pool for each endpoint using that master key to originate outbound data. Each such originating endpoint MUST only issue SSRC values from the pool it has been assigned. Further, each originating endpoint MUST maintain a history of outbound SSRC identifiers that it has issued within the lifetime of the current master key, and when a new synchronization source requests an SSRC identifier it MUST NOT be given an identifier that has been previously issued. A rekey MUST be performed before any of the originating endpoints using that master key exhausts its pool of SSRC values. Further, the identity of the entity giving out SSRC values MUST be verified, and the SSRC signaling MUST be integrity protected.

9. AES-GCM Processing of SRTCP Compound Packets

All SRTCP compound 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 just before the 31-bit SRTCP index)

9.1. SRTCP IV formation for AES-GCM

The 12-octet initialization vector used by AES-GCM SRTCP is formed by first concatenating 2-octets of zeroes, the 4-octet Synchronization Source identifier (SSRC), 2-octets of zeroes, a single zero bit, and the 31-bit SRTCP Index. The resulting 12-octet value is then XORed to the 12-octet salt to form the 12-octet IV.

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

Figure 4: SRTCP Initialization Vector formation

9.2. Data Types in Encrypted SRTCP Compound Packets

```
        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                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    R  |                  SRTCP MKI (optional) index                   :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    R  :           SRTCP authentication tag (NOT RECOMMENDED)          :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                P = Plaintext (to be encrypted and authenticated)
                A = Associated Data (to be authenticated only)
                R = neither encrypted nor authenticated, added after
                    encryption
```

Figure 5: AEAD SRTCP inputs when encryption flag = 1.

When the encryption flag is set to 1, the SRTCP packet is broken into plaintext, associated data, and raw (untouched) data (as shown above in figure 5):

Associated Data: The packet version V (2 bits), padding flag P (1 bit), reception report count RC (5 bits), packet type (8 bits), length (2 octets), SSRC (4 octets), encryption flag (1 bit) and SRTCP index

(31 bits).

Raw Data: The optional variable length SRTCP MKI and SRTCP authentication tag (whose use is NOT

```
                       RECOMMENDED).

     Plaintext:        All other data.
```

Note that the plaintext comes in one contiguous field. Since the AEAD cipher is larger than the plaintext by exactly the length of the AEAD authentication tag, the corresponding SRTCP encrypted packet replaces the plaintext field with a slightly larger field containing the cipher. Even if the plaintext field is empty, AEAD encryption must still be performed, with the resulting cipher consisting solely of the authentication tag. This tag is to be placed immediately before the encryption flag and SRTCP index.

9.3. Data Types in Unencrypted SRTCP Compound Packets

```
        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                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     R |                  SRTCP MKI (optional) index                   :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     R :              authentication tag (NOT RECOMMENDED)             :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                A = Associated Data (to be authenticated only)
                R = neither encrypted nor authenticated, added after
```

```
                    encryption
```

Figure 6: AEAD SRTCP inputs when encryption flag = 0

When the encryption flag is set to 0, the SRTCP compound packet is broken into plaintext, associated data, and raw (untouched) data as follows (see figure 6):

Plaintext: None.

Raw Data: The variable length optional SRTCP MKI and SRTCP authentication tag (whose use is NOT RECOMMENDED).

Associated Data: All other data.

Even though there is no ciphertext in this RTCP packet, AEAD encryption returns a cipher field which is precisely the length of the AEAD authentication tag. This cipher is to be placed before the Encryption flag and the SRTCP index in the authenticated SRTCP packet.

9.4. Prevention of SRTCP IV Reuse

A new master key MUST be established before the 31-bit SRTCP index cycles back to its original value. Ideally, a rekey should be performed and a new master key put in place well before the SRTCP cycles back to the starting value.

The comments on SSRC management in section 8.4 also apply.

10. Constraints on AEAD 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 one specific fasmily of AEAD algorithms, namely AES-GCM.

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

PARAMETER	Meaning	Value
A_MAX	maximum associated data length	MUST be at least 12 octets.
N_MIN	minimum nonce (IV) length	MUST be 12 octets.
N_MAX	maximum nonce (IV)	MUST be 12 octets.

	length	
P_MAX	maximum plaintext length per invocation	GCM: MUST be <= 2^36-32 octets.

C_MAX	maximum ciphertext length per invocation	GCM: MUST be <= 2^36-16 octets.

For sake of clarity we specify two additional parameters:

AEAD Authentication Tag Length	MUST be 8 or 16 octets,
Maximum number of invocations for a given instantiation	SRTP: MUST be at most 2^48, SRTCP: MUST be at most 2^31.
Block Counter size	GCM: MUST be 32 bits.

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

11. 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 endpoints. Both AEAD_AES_128_GCM and AEAD_AES_128_GCM_8 algorithms MUST use the (128-bit) AES_CM_PRF Key Derivation Function described in [RFC3711]. AEAD_AES_256_GCM MUST use the AES_256_CM_PRF Key Derivation Function described in [RFC6188].

12. Summary of AES-GCM in SRTP/SRTCP

For convenience, much of the information about the use of AES-GCM family of algorithms in SRTP is collected in the tables contained in this section.

The AES-GCM 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:

Name	Key Size	AEAD Tag Size	Reference
AEAD_AES_128_GCM_8	16 octets	8 octets	[RFC5282]
AEAD_AES_128_GCM	16 octets	16 octets	[RFC5116]
AEAD_AES_256_GCM	32 octets	16 octets	[RFC5116]

Table 1: AES-GCM algorithms for SRTP/SRTCP

Any implementation of AES-GCM SRTP MUST support both AEAD_AES_128_GCM and AEAD_AES_256_GCM (the versions with 16 octet AEAD authentication tags), and it MAY support AEAD_AES_128_GCM_8. Below we summarize parameters associated with these three GCM algorithms:

Parameter	Value
Master key length	128 bits
Master salt length	96 bits
Key Derivation Function	AES_CM_PRF [RFC3711]
Maximum key lifetime (SRTP)	2^48 packets
Maximum key lifetime (SRTCP)	2^31 packets
Cipher (for SRTP and SRTCP)	AEAD_AES_128_GCM_8
AEAD authentication tag length	64 bits

Table 2: The AEAD_AES_128_GCM_8 Crypto Suite

Parameter	Value
Master key length	128 bits
Master salt length	96 bits
Key Derivation Function	AES_CM_PRF [RFC3711]
Maximum key lifetime (SRTP)	2^48 packets
Maximum key lifetime (SRTCP)	2^31 packets
Cipher (for SRTP and SRTCP)	AEAD_AES_128_GCM
AEAD authentication tag length	128 bits

Table 3: The AEAD_AES_128_GCM Crypto Suite

Parameter	Value
Master key length	256 bits
Master salt length	96 bits
Key Derivation Function	AES_256_CM_PRF [RFC6188]
Maximum key lifetime (SRTP)	2^48 packets
Maximum key lifetime (SRTCP)	2^31 packets
Cipher (for SRTP and SRTCP)	AEAD_AES_256_GCM
AEAD authentication tag length	128 bits

Table 4: The AEAD_AES_256_GCM Crypto Suite

13. Security Considerations

13.1. 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.
- At the start of each packet, the block counter MUST be reset to 1. The block counter is incremented after each block key has been produced, but it MUST NOT be allowed to exceed 2^32-1 for GCM. Note that even though the block counter is reset at the start of each packet, IV uniqueness is ensured by the inclusion of SSRC/ROC/SEQ or SRTCP Index in the IV. (The reader is reminded that the first block of key produced is reserved for use in authenticating the packet and is not used to encrypt plaintext.)
- Each time a rekey occurs, the initial values of both the 31-bit SRTCP index and the 48-bit SRTP packet index (ROC||SEQ) MUST be saved in order to prevent IV reuse.
- Processing MUST cease if either the 31-bit SRTCP index or the 48-bit packet index ROC||SEQ cycles back their initial values . 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 any of these counters cycle.

13.2. 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 reduce 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.
```

When the expected number of successes is much less than one, the probability of success is well approximated by the expected number of successes.

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.

It should be noted that the cryptographic properties of the GHASH algorithm used in GCM reduces the effective authentication tag size (in bits) by the log base 2 of the of blocks of encrypted and/or authenticated data in a packet. In practice an SRTP payload will be less than 2^16 bytes, because of the 16-bit IPv4 and UDP length fields. The exception to this case is IPv6 jumbograms [RFC2675], which is unlikely to be used for RTP-based multimedia traffic [RFC3711]. This corresponds to 2^12 blocks of data, so the effective GCM authentication tag size is reduced by at most 12 bits.

Auth. Tag Size (bytes)	Effective Tag Size (bits)	Number of Forgery Attempts Needed to Achieve a Given Probability of Success		
		prob=2^-30	prob=2^-20	prob=2^-10
4	20 (GCM)	1 try	1 try	2^10 tries
8	52 (GCM)	2^22 tries	2^32 tries	2^42 tries
12	84 (GCM)	2^54 tries	2^64 tries	2^74 tries
16	116 (GCM)	2^86 tries	2^96 tries	2^106 tries

Table 5: Number of forgery attempts needed to achieve a given probability of success for various tag sizes.

14. IANA Considerations

14.1. SDES

SDP Security Descriptions [RFC4568] defines SRTP "crypto suites". A crypto suite corresponds to a particular AEAD algorithm in SRTP. In order to allow Security Descriptions to signal the use of the algorithms defined in this document, IANA will register the following crypto suites into the "SRTP Crypto Suite Registrations" subregistry of the "Session Description Protocol (SDP) Security Descriptions" registry.

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

14.2. DTLS-SRTP

DTLS-SRTP [RFC5764] defines a DTLS-SRTP "SRTP Protection Profile". These also correspond 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, we request IANA register the following SRTP Protection Profiles:

```
      AEAD_AES_128_GCM    = {TBD, TBD }
      AEAD_AES_128_GCM_8  = {TBD, TBD }
      AEAD_AES_256_GCM    = {TBD, TBD }
```

Below we list the SRTP transform parameters for each of these protection profile. Unless separate parameters for SRTCP and SRTCP are explicitly listed, these parameters apply to both SRTP and SRTCP.

```
AEAD_AES_128_GCM
     cipher:                  AES_128_GCM
     cipher_key_length:       128 bits
     cipher_salt_length:      96 bits
     aead_auth_tag_length:    16 octets
     auth_function:           NULL
     auth_key_length:         N/A
     auth_tag_length:         N/A
     maximum lifetime:        at most 2^31 SRTCP packets and
                              at most 2^48 SRTP packets
AEAD_AES_128_GCM_8
     cipher:                  AES_128_GCM
     cipher_key_length:       128 bits
     cipher_salt_length:      96 bits
     aead_auth_tag_length:    8 octets
     auth_function:           NULL
     auth_key_length:         N/A
     auth_tag_length:         N/A
     maximum lifetime:        at most 2^31 SRTCP packets and
                              at most 2^48 SRTP packets
```

AEAD_AES_256_GCM

cipher:	AES_256_GCM
cipher_key_length:	256 bits

cipher_salt_length:	96 bits
aead_auth_tag_length:	16 octets
auth_function:	NULL
auth_key_length:	N/A
auth_tag_length:	N/A
maximum lifetime:	at most 2^31 SRTCP packets and at most 2^48 SRTP packets

Note that these SRTP Protection Profiles do not specify an auth_function, auth_key_length, or auth_tag_length because all of these profiles use AEAD algorithms, and thus do not use a separate auth_function, auth_key, or auth_tag. The term aead_auth_tag_length is used to emphasize that this refers to the authentication tag provided by the AEAD algorithm and that this tag is not located in the authentication tag field provided by SRTP/SRTCP.

14.3. MIKEY

In accordance with "MIKEY: Multimedia Internet KEYing" [RFC3830], IANA maintains several subregitries under "Multimedia Internet KEYing (MIKEY) Payload Name Spaces". This document requires additions to two of the MIKEY subregistries.

In the "MIKEY Security Protocol Parameters" subregistry we request the following addition:

Type	Meaning	Possible values
TBD	AEAD authentication tag length	8 octets or 16 octets

This list is, of course, intended for use with GCM. It is conceivable that new AEAD algorithms introduced at some point in the future may require a different set of Authentication tag lengths.

In the "Encryption Algorithm" subregistry (derived from Table 6.10.1.b of [RFC3830]) we request the following addition:

SRTP encr Algorithm	Value	Default Session Encr. Key Length	Default Auth. Tag Length
AES-GCM	TBD	16 octets	16 octets

The SRTP encryption algorithm, session encryption key length, and AEAD authentication tag values received from MIKEY fully determine the AEAD algorithm (e.g., AEAD_AES_256_GCM_8). The exact mapping is described in section 16.

15. Parameters for use with MIKEY

MIKEY specifies the algorithm family separately from the key length (which is specified by the Session Encryption key length) and the authentication tag length (specified by AEAD Auth. tag length).

	Encryption Algorithm	Encryption Key Length	AEAD Auth. Tag Length
AEAD_AES_128_GCM_8	AES-GCM	16 octets	8 octets
AEAD_AES_128_GCM	AES-GCM	16 octets	16 octets
AEAD_AES_256_GCM	AES-GCM	32 octets	16 octets

Table 6: Mapping MIKEY parameters to AEAD algorithm

Section 11 in this document restricts the choice of Key Derivation Function for AEAD algorithms. To enforce this restriction in MIKEY, we require that the SRTP PRF has value AES-CM whenever an AEAD algorithm is used. Note that, according to Section 6.10.1 in [RFC3830], the input key length of the Key Derivation Function (i.e. the SRTP master key length) is always equal to the session encryption key length. This means, for example, that AEAD_AES_256_GCM will use AES_256_CM_PRF as the Key Derivation Function.

16. Acknowledgements

The authors would like to thank Michael Peck, Michael Torla, Qin Wu, Magnus Westerland, Oscar Ohllson, Woo-Hwan Kim, John Mattsson, Richard Barnes, John Mattisson, Morris Dworkin, Stehen Farrell and many other reviewers who provided valuable comments on earlier drafts of this document.

17. References

17.1. Normative References

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

[RFC3550] Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, September 2003.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M.,and Norrman, K, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4568] 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.

[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] D. McGrew, "The Use of AES-192 and AES-256 in Secure RTP", RFC 6188, March 2011.

[RFC6904] J. Lennox, "Encryption of Header Extensions in the Secure Real-Time Transport Protocol (SRTP)", January 2013.

, January 2013.

[RFC6904] J. Lennox, "Encryption of Header Extensions in the Secure Real-Time Transport Protocol (SRTP)", January 2013.

17.2. 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.

[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.

[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.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[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).