Network Working Group
Internet Draft
Intended Status: Informational
Expires: March 22, 2013

D. McGrew
Cisco Systems, Inc.
K. Igoe
National Security Agency
September 18, 2012

AES-GCM and AES-CCM Authenticated Encryption in Secure RTP (SRTP)
draft-ietf-avtcore-srtp-aes-gcm-03

Status of this Memo

Copyright Notice

Abstract

This document defines how AES-GCM and AES-CCM Authenticated Encryption with Associated Data algorithms can be used to provide confidentiality and data authentication 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). 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. 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.

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. Overview of the SRTP/SRTCP Security Architecture

SRTP/SRTCP 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-CCM or 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 sections 4.3.1 and 4.3.3 of [RFC3711]. Since AEAD algorithms such as AES-CCM and AES-GCM combine encryption and authentication into a single process, AEAD algorithms do not make use of the authentication keys. The master key MUST be at least as large as the encryption key derived from it.

4. Terminology

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 endpoint in the SRTP session and possession by each endpoint of a shared secret master key. Various encryption keys, authentication keys and salts are derived from the master key. Aside from making modifications to IANA registries to allow AES-GCM and AES-CCM to work with SDES, DTLS and MIKEY, the details of how the master key is established are outside the scope of this document. Similarly any mechanism for rekeying an existing Cipher Context is outside the scope of the document.

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 for SRTP and one for SRTCP.

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 9.2, 10.2 and 10.3.

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	Bit string of variable length
Plaintext	Bit string of variable length
Tag_Size_Flag (CCM only*)	One Octet
Outputs	
Ciphertext	Bit string, length = length(Plaintext)+tag_length

(*) For GCM, the algorithm choice determines the tag size.

AES-CCM uses a Tag_Size_Flag that 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.

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
Tag_Size_Flag (CCM only*)	One octet
Outputs	
Plaintext	Bit string, length = length(Ciphertext)-tag_length
Validity_Flag	Boolean, TRUE if valid, FALSE otherwise

(*) For GCM, the algorithm choice determines the tag size.

The Tag_Size_Flag, used in AES-CCM authentication, 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.

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, when GCM is being used, 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

In both GCM and CCM, 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 informing the authentication tag and keystream of octets

which is XORed to the plaintext to form cipher.

When GCM is used, the concatenation of a 12-octet IV, with a 4-octet

block counter forms the input to AES. This is used to build a key_stream as follows:

```
 def GCM_keystream( Plaintext, IV, Encryption_key ):
     assert len(plaintext)  <= (2**36) - 32
     key_stream = ""
     block_counter = 1
     first_key_block = AES_ENC( data=IV||block_counter,
                                key=Encryption_key        )
     while len(key_stream) < len(Plaintext):
         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, len(Plaintext) )
     return (first_key_block, key_stream )
```

In AES-CCM counter mode encryption, the AES data input consists of the concatenaton of a 1-octet flag,, a 12-octet IV, and a 3-octet block counter. Note that in this apllication the flag octet will always have the value 0x02 (see section 2.3 of [RFC3610]). A (first_key_block, keystream) pair is formed as follows:

```
 def CCM_keystream( Plaintext, IV, Encryption_key ):
     assert len(Plaintext) <= (2**28)-16
     key_stream = ""
     block_counter = 0
     first_key_block = AES_ENC( data=0x02||IV||block_counter,
                                key=Encryption_key            )
     while len(key_stream)<len(Plaintext):
         block_counter = block_counter + 1
         key_block = AES_ENC( data=0x02||IV||block_counter,
                              key=Encryption_key            )
         key_stream  = key_stream || key_block
     key_stream = truncate( key_stream, len(Plaintext) )
     return (first_key_block, key_stream )
```

These keystream generation processes allows for a keystream of length of up to (2^24)-16 octets for AES-CCM and up to (2^36)-32 octets for AES-GCM.

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

AES-GCM, the consequences of such a reuse are even worse than explained in RFC 3711 because it would completely compromise the AES-GCM authentication mechanism.

7. AEAD_AES_128_CCM_12 and AEAD_AES_256_CCM_12

AEAD_AEC_128_CCM and AEAD_AEC_256_CCM are defined in [RFC5116] with an authentication tag length of 16-octets. AEAD_AEC_128_CCM_8 and AEAD_AEC_256_CCM_8 are defined in [RFC6655] with an authentication tag length of 8-octets. We require two new variants, AEAD_AES_128_CCM_12 and AEAD_AES_256_CCM_12, with 12-octet authentication tags. In each case the authentication tag is formed by taking the 12 most significant octets (in network order) of the AEAD_AES_128/256_CCM authentication tag:

Name	Key Size	tag size (t)
AEAD_AEC_256_CCM_12	256 bits	12 octets
AEAD_AEC_128_CCM_12	128 bits	12 octets

8. Unneeded SRTP/SRTCP Fields

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

8.1. 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.

8.2. RTP Padding

Neither AES-GCM not AES-CCM 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 RECOMENDED that the RTP padding mechanism not be used unless it is necessary to disguise the length of the

underlying plaintext.

9. AES-GCM/CCM processing for SRTP

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

The 12 octet initialization vector used by both AES-GCM and AES-CCM SRTP is formed by first concatenating 2-octets of zeroes, the 4-octet SSRC, the 4-octer 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.

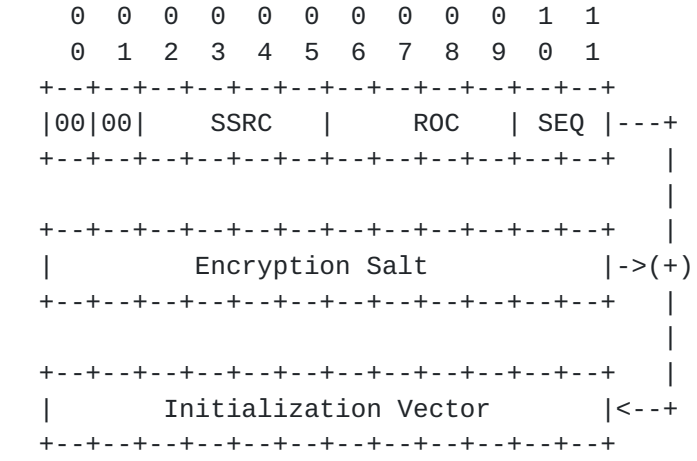


Figure 1: AES-GCM and AES-CCM SRTP Initialization Vector formation.

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 octets are the invocation field.

9.2. Data Types in SRTP Packets

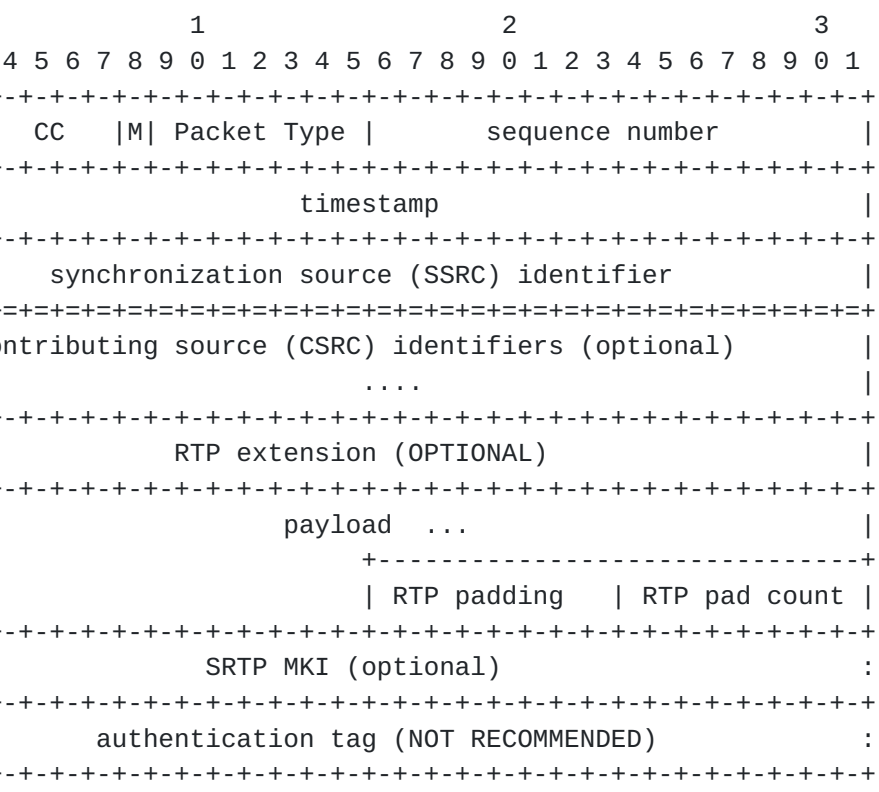
All SRTP packets MUST be both authenticated and encrypted. The data fields within the SRTP packets are broken into Associated Data, Plaintext and Raw Data as follows (see figure 2):

Associated Data — The version (2 bits), padding flag (1 bit), extension flag (1 bit), CSRC count (4 bits), 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 32-bit SRTP MKI and the 32-bit SRTP authentication tag (whose use is NOT

RECOMMENDED).



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

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

Figure 2: AEAD inputs from an SRTP packet

Since the AEAD cipher 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 SRTP MKI and SRTP authentication tag fields.

9.3. 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.
SSRC Management	The set of all SSRC values must be partitioned into disjoint pools, one pool for each endpoint using the master key to originate outbound data. Each such endpoint MUST only issue SSRC values from the pool it has been assigned. Further, each 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 its pool of SSRC values is exhausted.

10. AES-GCM/CCM 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)

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

The 12 octet initialization vector used by both AES-GCM and AES-CCM 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.

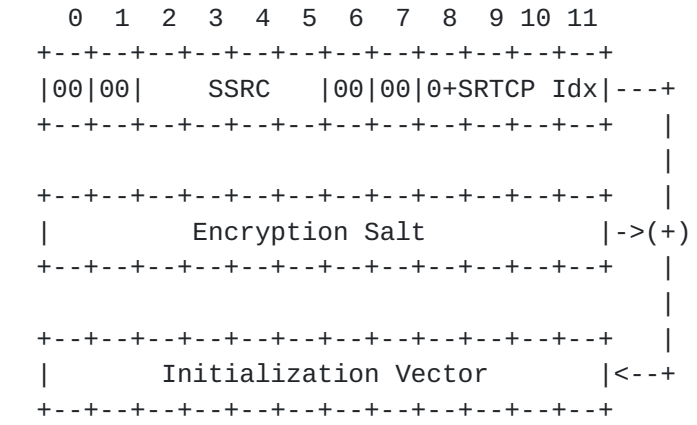


Figure 3: SRTCP Initialization Vector formation

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 octets are the invocation field.

10.2. Data Types in Encrypted SRTCP Compound Packets

When the encryption flag is set to 1, the SRTCP packet is broken into plaintext, associated data, and raw (untouched) data as listed below (see figure 4):

Associated Data	The packet version (2 bits), padding flag (1 bit), reception report count (5 bits), packet type (8 bits), length (2 octets), SSRC (4 octets), encryption flag (1 bit) and SRTCP index (31 bits).
Raw Data	The 32-bit optional SRTCP MKI index and 32-bit 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.

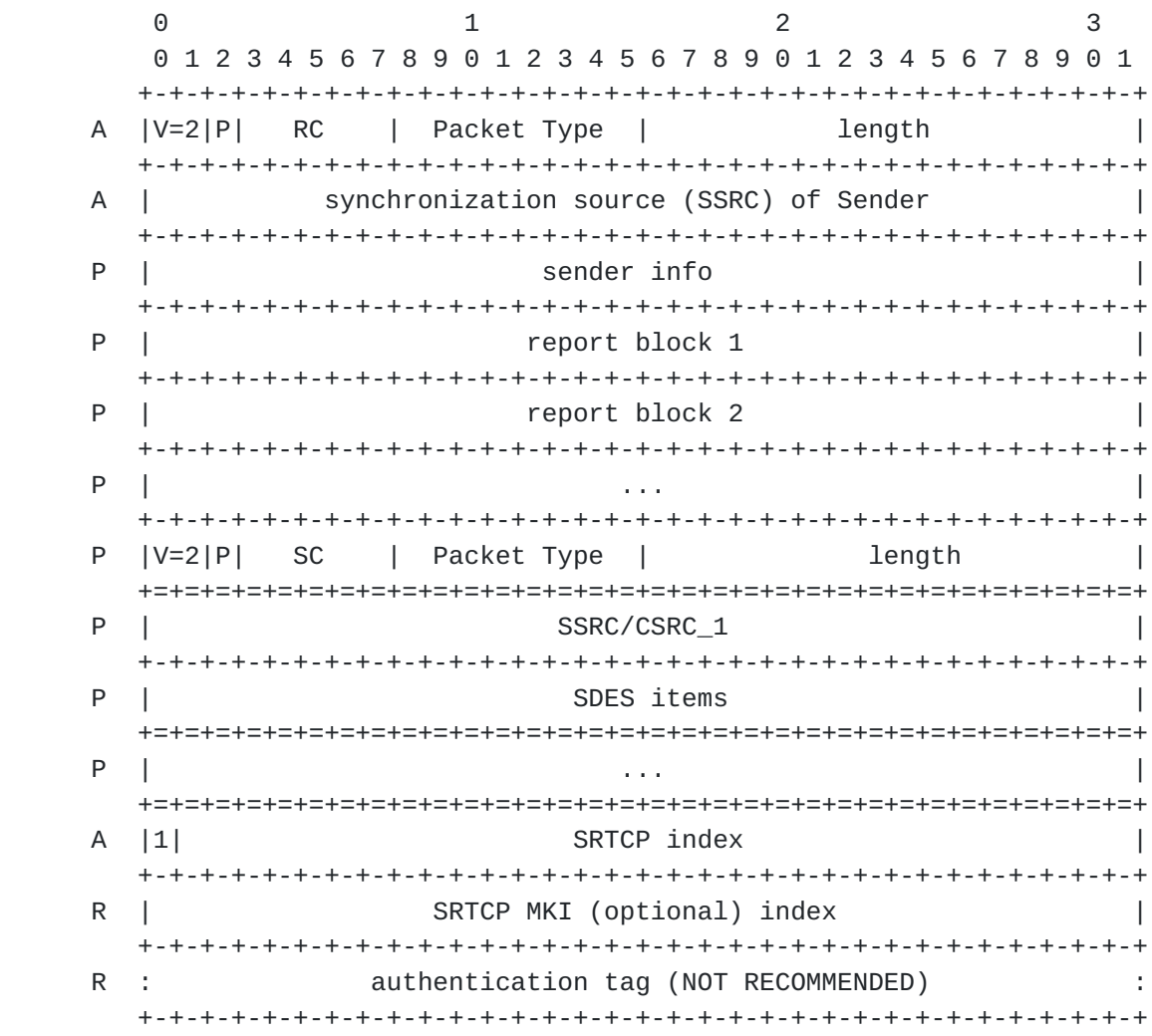


Figure 4: AEAD SRTCP inputs when encryption flag = 1.

10.3. Data Types in Unencrypted SRTCP Compound Packets

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 5):

Plaintext	None.
Raw Data	The 32-bit optional SRTCP MKI index and 32-bit SRTCP authentication tag (whose use is NOT RECOMMENDED).

Associated Data All other data.

Even though there is no plaintext 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.

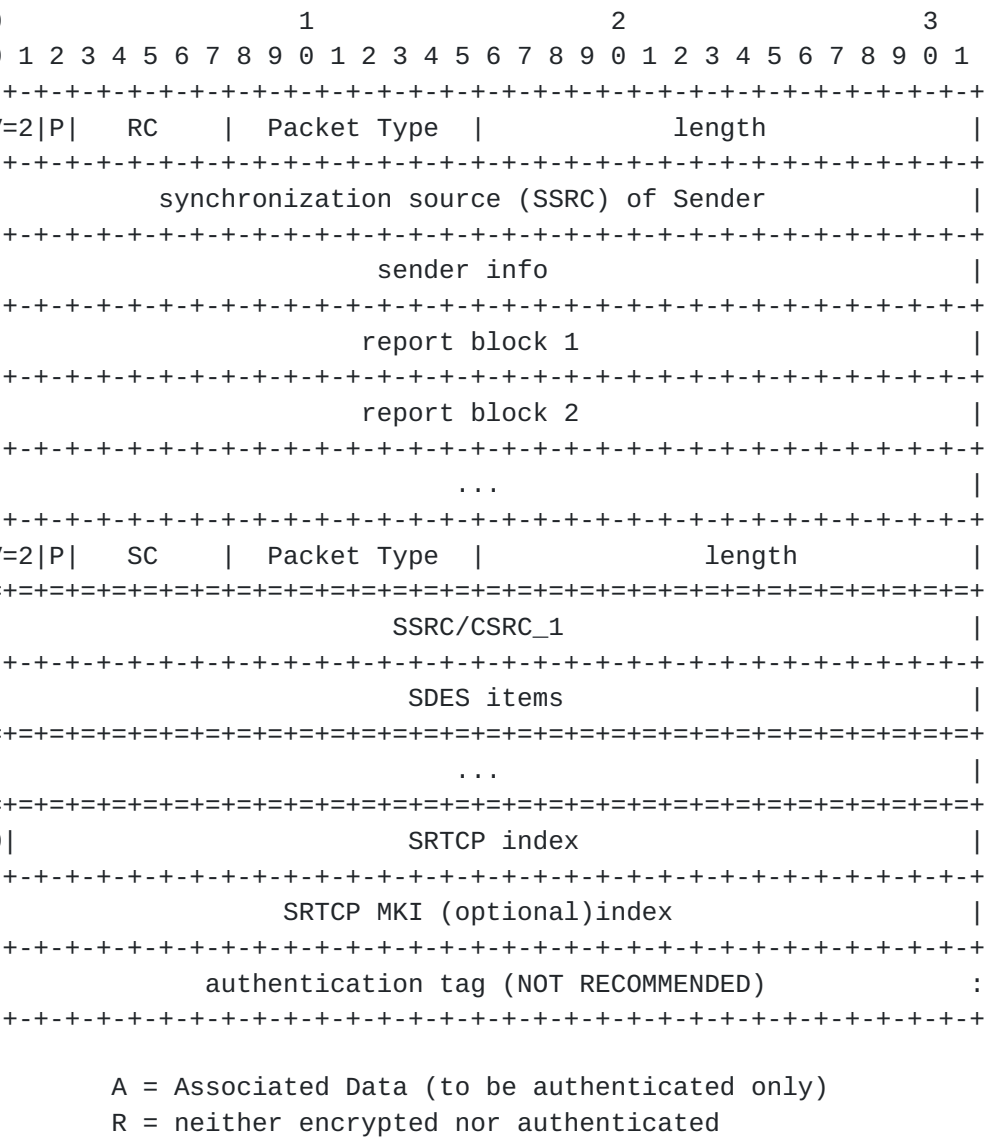


Figure 5: AEAD SRTCP inputs when encryption flag = 0

10.4. 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 performed should be performed and a new master key put in place well before the SRTCP index overflows.

The comments on SSRC management in section 9.3 also apply.

11. 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 two specific AEAD algorithms, namely AES-GCM and AES-CCM.

11.1. 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	GCM: MUST be at most 2^36-16 octets CCM: MUST be at most 2^24+16 octets C_MAX values less than 2232 are discouraged

The upper bound on C_MAX are based on purely cryptographic considerations. The lower bound on C_MAX is obtained by subtracting away a 20-octet IP header, 8-octet UDP header, and 12-octet RTP header from the maximum transmission unit (MTU) of 2272.

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
Block Counter size	MUST be 24 bits for CCM, MUST be 32 bits for GCM

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

11.2. 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 table 1.

11.3. AES-CCM for SRTP/SRTCP

AES-CCM is another family of AEAD algorithms built around the AES block cipher algorithm. AES-CCM 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]. Four of the six CCM algorithms used in this document are defined in previous RFCs, while two, AEAD_AES_128_CCM_12 and AEAD_AES_256_CCM_12, are defined in section 7 of this document.

Table 2: AES-CCM algorithms for SRTP/SRTCP

Name	Key Size	Auth. Tag Size	Reference
AEAD_AES_128_CCM	128 bits	16 octets	[RFC5116]
AEAD_AES_256_CCM	256 bits	16 octets	[RFC5116]
AEAD_AES_128_CCM_12	128 bits	12 octets	see section 7
AEAD_AES_256_CCM_12	256 bits	12 octets	see section 7
AEAD_AES_128_CCM_8	128 bits	8 octets	[RFC6655]
AEAD_AES_256_CCM_8	256 bits	8 octets	[RFC6655]

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

In addition to the flag octet used in counter node encryptionn, AES-CCM authentications also useus a flag octet that conveys information about the length of the authentication tag, length of the block counter, and presence of additional authenticated data (see section 2.2 of [RFC 3610]). 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.

12. 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 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].

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 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 for GCM and 2^24 for CCM.
- Each time a rekey occurs, the initial values of the SRTCP index and the values of all the SEQ counters 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.

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 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 for a given number of forgery attempts		
	prob=2^-30	prob=2^-20	prob=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 3: Probability of a compromise occurring for a given number of forgery attempts and tag size.

14. IANA Considerations

14.1. SDES

Security description [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"    /
```

14.2. DTLS

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, we request IANA register the following SRTP Protection Profiles:

```
AEAD_AES_128_CCM
     cipher:               AES_128_CCM
     cipher_key_length:    128 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_256_CCM
     cipher:               AES_256_CCM
     cipher_key_length:    256 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_128_CCM_8
     cipher:               AES_128_CCM
     cipher_key_length:    128 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_256_CCM_8
     cipher:               AES_256_CCM
     cipher_key_length:    256 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_128_CCM_12
     cipher:               AES_128_CCM
     cipher_key_length:    128 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_256_CCM_12
     cipher:               AES_256_CCM
     cipher_key_length:    256 bits
```

```
        cipher_salt_length:   96 bits
        maximum lifetime:     at most 2^31 SRTCP packets and
                              at most 2^48 SRTP packets
```

```
AEAD_AES_128_GCM
     cipher:               AES_128_GCM
     cipher_key_length:    128 bits
     cipher_salt_length:   96 bits
     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
     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
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_256_GCM_8
     cipher:               AES_256_GCM
     cipher_key_length:    256 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_128_GCM_12
     cipher:               AES_128_GCM
     cipher_key_length:    128 bits
     cipher_salt_length:   96 bits
     maximum lifetime:     at most 2^31 SRTCP packets and
                           at most 2^48 SRTP packets

AEAD_AES_256_GCM_12
     cipher:               AES_256_GCM
     cipher_key_length:    256 bits
     cipher_salt_length:   96 bits
     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.

14.3. MIKEY

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

On the SRTP policy Type/Value list (derived from Table 6.10.1.a of [RFC3830]) we request the following addition:

Type	Meaning	Possible values
TBD	AEAD authentication tag length	8, 12, or 16 (in octets)

On the Encryption Algorithm List (derived from Table 6.10.1.b of [RFC3830]) we request the following additions:

SRTP encr alg.	Value	Default Session Encr. Key Length
AES-CCM	TBD	16 octets
AES-GCM	TBD	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 15.

14.4. AEAD registry

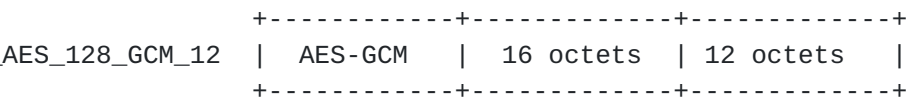
We request that IANA make the following additions to the AEAD registry:

```
AEAD_AES_128_CCM_12     = TBD
AEAD_AES_256_CCM_12     = TBD
```

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	AES-GCM	16 octets	16 octets
AEAD_AES_128_CCM	AES-CCM	16 octets	16 octets



AEAD_AES_128_GCM_12	AES-GCM	16 octets	12 octets

AEAD_AES_128_CCM_12	AES-CCM	16 octets	12 octets
AEAD_AES_128_GCM_8	AES-GCM	16 octets	8 octets
AEAD_AES_128_CCM_8	AES-CCM	16 octets	8 octets
AEAD_AES_256_GCM	AES-GCM	32 octets	16 octets
AEAD_AES_256_CCM	AES-CCM	32 octets	16 octets
AEAD_AES_256_GCM_12	AES-GCM	32 octets	12 octets
AEAD_AES_256_CCM_12	AES-CCM	32 octets	12 octets
AEAD_AES_256_GCM_8	AES-GCM	32 octets	8 octets
AEAD_AES_256_CCM_8	AES-CCM	32 octets	8 octets

Table 4: Mapping MIKEY parameters to AEAD algorithm

Section 12 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 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 and many other reviewers who provided valuable comments on earlier drafts of this document.

17. References

17.1. 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.

[RFC3610] Whiting,D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, March 2004.

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

RFC 6188, March 2011

[RFC6655] McGrew,D. and D. Bailey,"AES-CCM Cipher Suites for Transport

Layer Security (TLS)", July 2012

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.

[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 Security, 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).