AVT Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

D. McGrew
D. Wing
Cisco
J. Foley
Cisco Systems
October 21, 2013

Using Authenticated Encryption with Replay prOtection (AERO) in SRTP
draft-mcgrew-srtp-aero-01

Abstract

Authenticated Encryption with Replay prOtection (AERO) is a cryptographic technique that provides all of the security services that are used in the Secure Real-time Transport Protocol (SRTP). This note describes how to use AERO in SRTP. AERO has minimal data expansion, avoids the need to manage implicit state, and provides strong misuse resistance. These properties make it an ideal cryptographic transform for SRTP, as it enables SRTP to easily handle multiple senders sharing the same key, multiple receivers with late-joiners in a session, decentralized conferences with minimal control, and mixers that selectively forward RTP traffic. RTP architectures that utilize AERO can use the normal SSRC collision detection mechanism, and can ignore problematic SRTP artifacts such as the Roll-Over Counter (ROC) and Initial Sequence Number.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Table of Contents

1. Introduction

RTP is designed to allow decentralized groups with minimal control to establish sessions, such as for multimedia conferences. Unfortunately, Secure RTP (SRTP [RFC3711]) cannot be used in many minimal-control scenarios, because it requires that SSRC values and other data be coordinated among all of the participants in a session. For example, if a participant joins a session that is already in progress, the SRTP Roll-Over Counter (ROC) of each SRTP source in the session needs to be provided to that participant.

The inability of SRTP to work in the absence of central control was well understood during the design of that protocol; that omission was considered less important than optimizations such as bandwidth conservation. Additionally, in many situations SRTP is used in conjunction with a signaling system that can provide much of the central control needed by SRTP. However, there are several cases in which conventional signaling systems cannot easily provide all of the coordination required. It is also desirable to eliminate the layer violations that occur when signaling systems coordinate certain SRTP parameters, such as SSRC values and ROCs.

These issues are due to the particular cryptographic techniques used in SRTP, specifically a partially-implicit sequence number that is utilized for counter mode encryption, for replay protection, and for determining when re-keying should occur. Authenticated Encryption with Replay prOtection (AERO) [I-D.mcgrew-aero] is a cryptographic technique that provides confidentiality, authentication, and replay protection; it is a stateful and self-synchronizing authenticated encryption method. It has minimal data expansion, avoids the need to manage implicit state, and provides strong misuse resistance. This document defines how AERO can be used in SRTP as a replacement for the default transforms of [RFC3711] in a way that avoids all of the issues identified above.

This document is organized as follows. Packet processing and contexts are described in Section 2 and Section 3. A rationale for the design is offered in Section 5. Security Considerations are provided in Section 6, and IANA considerations are provided in Section 7.

1.1. 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].

1.2. History

This section describes the evolution of this document as an Internet Draft, and it should be removed by the RFC Editor prior to publication as an RFC.

This is the initial version of this note. As it uses a cryptographic technique that was published recently, it may evolve over time as that technique is reviewed and experience is gained in its usage. Thus, while we encourage the implementation of the crypto suites specified in this note as the best way to obtain experience with their use in RTP architectures, we also expect that there may be changes to these crypto suites over time.

2. SRTP AERO

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of the Real-time Transport Protocol (RTP) that 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 provides a framework for the encryption and message authentication of RTP and RTCP streams. Default cryptographic transforms are defined in [RFC3711], while the framework supports the addition of new cryptographic transforms.

The note describes how to use Authenticated Encryption with Replay prOtection (AERO) [I-D.mcgrew-aero] in SRTP, which we refer to as SRTP-AERO. All three security services - confidentiality, message authentication, and replay protection - are always provided with SRTP-AERO; none of those services is optional. The SRTP framework includes provisions for separate encryption and authentication transforms; when AERO is used, it is considered an encryption transform and there MUST NOT be a separate authentication transform.

This description mainly consists of how AERO encryption inputs and outputs are mapped to RTP and SRTP packets, respectively, how AERO decryption inputs and outputs are mapped to SRTP and RTP packets, respectively, and how the AERO context is stored in the SRTP context. A similar description is provided for SRTCP. The security considerations are different from those in [RFC3711] (and are rather less stringent).

With SRTP-AERO, the normal RTP SSRC collision detection and repair process can be followed. This is in contrast to SRTP with the default transforms, which relies on external mechanisms to coordinate SSRC values.

AERO is an Authenticated Encryption with Associated Data (AEAD) method, and thus SRTP-AERO processing is similar to that of [I-D.ietf-avtcore-srtp-aes-gcm].

2.1. Encryption

To protect an RTP packet using AERO, the inputs to the encryption operation are as follows:

The associated data A is set to the RTP header, including the CSRC identifiers (if present), and the RTP header extension (if present). (In Figure 1 of [RFC3711], this consists of the part of the Authenticated Portion of the packet that does not overlap with the Encrypted Portion.)

The plaintext is set to the RTP payload, RTP padding, and RTP pad count. (In Figure 1 of [RFC3711], this is the Encrypted Portion.)

The secret key K is set to the AERO key in the SRTP context.

The ciphertext returned by that operation replaces the plaintext in the RTP packet. Note that the ciphertext will be longer than the plaintext.

AERO does not include a separate authentication tag, and thus this field is omitted from the SRTP packet; it MUST NOT be present. This omission simplifies this specification (see Section 5).

A Master Key Indicator (MKI) field MAY be present after the ciphertext; this field is optional in [RFC3711] and its usage is unchanged by this specification.

2.2. Decryption

To decrypt an SRTP packet using AERO, the inputs to the decryption operation are as follows:

The associated data A is set as in SRTP encryption Section 2.1.

The ciphertext C is set as follows. It starts and the end of the RTP header extension, if that field is present. If it is not, then it starts at the end of the last CSRC identifier, if any CSRC identifiers are present. Otherwise, it starts at the end of the SSRC identifier. If an MKI is in use, then the ciphertext ends at the start of the MKI. Otherwise, the ciphertext ends at the end of the RTP packet.

The secret key K is set to the AERO key in the SRTP context.

If the decryption operation returns the symbol FAIL, then the SRTP packet is either a forgery attempt or a replay, and the packet MUST be discarded from further processing and the event MAY be logged. Otherwise, an RTP packet is formed from the SRTP packet and the plaintext returned by the decryption operation, by replacing the ciphertext with the plaintext, and discarding the MKI field if one is present.

Steps 2 and the replay checking in Step 5 of the decryption processing in Section 3.4 of [RFC3711] SHOULD NOT be performed, since it is unnecessary. The SRTP packet index used in Step 3 to determine the master key and salt SHOULD be set to the sequence number returned by the AERO decryption operation.

2.3. Contexts

The AERO context is stored in the transform-dependent parameters of the SRTP context, as the SRTP encryption transform. That is, the identifier for the encryption algorithm describes the particular AERO algorithm that is in use.

When AERO is used, an SRTP implementation MAY omit the following transform-independent parameters:

- the ROC,
- the replay list maintained by an SRTP/SRTCP receiver, and
- the identifier for the message authentication algorithm.

The ROC and replay list are not needed because AERO provides replay protection, and the message authentication algorithm identifier is not needed because AERO provides that security service.

3. Secure RTCP

The use of AERO in SRTCP follows that in SRTP.

3.1. Encryption

To protect an RTCP packet using AERO, the inputs to the encryption operation are as follows:

The associated data A is set to the RTCP header, including all of the fields starting with the Version and up to and including the SSRC of the sender.

The plaintext is set to the remaining part of the RTCP packet.

The secret key K is set to the AERO key in the SRTCP context.

The ciphertext returned by that operation replaces the plaintext in the RTCP packet. Note that the ciphertext will be longer than the plaintext.

AERO does not include a separate authentication tag, and thus this field is omitted from the SRTCP packet; it MUST NOT be present (see Section 5 for the rationale).

A Master Key Indicator (MKI) field MAY be present after the ciphertext; this field is optional in [RFC3711] and its usage is unchanged by this specification.

The "E" flag and the SRTCP index MUST NOT be included in the SRTCP packet.

3.2. Decryption

To decrypt an SRTCP packet using AERO, the inputs to the decryption operation are as follows:

The associated data A is set as in SRTCP encryption Section 2.1.

The ciphertext C consists of all of the data after the associated data, to the end of the packet.

The secret key K is set to the AERO key in the SRTCP context.

If the decryption operation returns the symbol FAIL, then the SRTCP packet is either a forgery attempt or a replay, and the packet MUST be discarded from further processing and the event MAY be logged. Otherwise, an RTCP packet is formed from the SRTCP packet and the

plaintext returned by the decryption operation, by replacing the ciphertext with the plaintext, and discarding the MKI field if one is present.

4. SRTP crypto suites

This section defines some crypto suites for use in SRTP, which can be signaled by DTLS-SRTP [RFC5764], SDP Security Descriptions [RFC4568] or MIKEY [RFC3830]. The use of these crypto suites in SDP Security Descriptions is straightforward; the particular crypto suite to be used is specified by the "crypto-suite" parameter. Their use in DTLS-SRTP is similarly straightforward; each crypto suite is described by an SRTP Protection Profile (as in Section 4.1.2 of [RFC5764]). For MIKEY, the use of a particular crypto suite is specified by both the MIKEY "encryption algorithm" parameter and the "session encryption key length" parameter, as noted below.

4.1. AERO_AES_128_XCB_80

The AERO_AES_128_XCB_80 crypto suite uses the AERO_AES_128_XCB algorithm defined in [I-D.mcgrew-aero], with a sequence number length T of 72 bits. It provides authentication strength equivalent to an ideal message authentication code with a 70-bit tag. The data expansion is 80 bits; the ciphertext will be at most 80 bits longer than the plaintext. The master-key length is 128 bits and has a default lifetime of a maximum of 2^48 SRTP packets or 2^31 SRTCP packets, whichever comes first (see page 39 [RFC3711]).

The SRTP and SRTCP encryption key lengths are 128 bits. The master salt value is 112 bits in length and the session salt value is 112 bits in length. The pseudo-random function (PRF) is the default SRTP pseudo-random function that uses AES Counter Mode with a 128-bit key length.

The length of the base64-decoded key and salt value for this crypto suite MUST be 30 characters (i.e., 240 bits); otherwise, the crypto attribute is considered invalid.

To signal the use of AERO_AES_128_XCB_80 with MIKEY, the following MIKEY SRTP Policy Parameters MUST be used:

An encryption algorithm parameter of TBD Section 7

A session encryption key length of 128 bits.

An authentication tag length of 80 bits.

4.2. AERO_AES_128_XCB_32

This crypto suite is identical to AERO_AES_128_XCB_80, except that it uses a sequence number length T of 24 bits. Its authentication strength is about 24 bits, and its data overhead is at most 32 bits.

To signal the use of AERO_AES_128_XCB_32 with MIKEY, the following MIKEY SRTP Policy Parameters MUST be used:

An encryption algorithm parameter of TBD Section 7

A session encryption key length of 128 bits.

An authentication tag length of 32 bits.

4.3. AERO_AES_256_XCB_128

The AERO_AES_256_XCB_128 crypto suite uses the AERO_AES_256_XCB algorithm defined in [I-D.mcgrew-aero], with a sequence number length T of 128 bits. It provides authentication strength equivalent to an ideal message authentication code with a 121-bit tag. The data expansion is 128 bits; the ciphertext will be exactly 128 bits longer than the plaintext. The master-key length is 128 bits and has a default lifetime of a maximum of 2^48 SRTP packets or 2^31 SRTCP packets, whichever comes first (see page 39 [RFC3711]).

The SRTP and SRTCP encryption key lengths are 256 bits. The master salt value is 112 bits in length and the session salt value is 112 bits in length. The pseudo-random function (PRF) is the AES_256_CM_PRF key derivation function [RFC6188] which uses AES Counter Mode with a 256-bit key length.

The length of the base64-decoded key and salt value for this crypto suite MUST be 46 characters (i.e., 368 bits); otherwise, the crypto attribute is considered invalid.

To signal the use of AERO_AES_128_XCB_80 with MIKEY, the following MIKEY SRTP Policy Parameters MUST be used:

An encryption algorithm parameter of TBD Section 7

A session encryption key length of 256 bits.

An authentication tag length of 128 bits.

5. Rationale

There is no need to allow the SRTP or SRTCP Authentication Tag fields to be present, and thus it is forbidden instead of being optional. Compatibility with [RFC4771], which uses this field as a means of conveying the Roll-Over Counter (ROC), is not needed because AERO removes any need for the ROC. Omitting this field simplifies implementations and avoids confusion.

The SSRC field serves as a Sender Identifier, and meets the requirements described in Section 3.5 of [I-D.mcgrew-aero].

5.1. Comparison to other approaches

There are other approaches that have been used to address the issues identified in Section 1. In this section, we compare them to SRTP AERO.

With the SRTP Integrity Transform Carrying Roll-Over Counter [RFC4771], the sender periodically includes the ROC in the SRTP authentication tag, in which case the authentication process is altered so that the ROC is authenticated. This approach addresses synchronization issues that are due to multiple receivers, such as the problem of "late joiners" in a session. However, it does not address any of the issues that are due to multiple senders using the same encryption key. It also does not change the need for misuse resistance.

With SRTP Encrypted Key Transport (EKT) [SRTP-EKT], the sender periodically includes additional data about the session in its outbound packets. This data includes the value of the master key being used for that SRTP source, encrypted under a master key that is used for all sources in the session. EKT address the issues that arise in multiple-sender sessions by providing a way that each source can use a distinct master key. EKT also includes the initial RTP sequence number, to aid receivers in establishing the appropriate SRTP packet index for the first packet in a session. EKT is more complex than SRTP-AERO, as it requires a separate authenticated encryption method for protecting the data that is conveyed, and it adds complexities to the packet processing rules. It also adds data overhead to SRTP and SRTCP packets in a way that is non-uniform, with some packets growing by a single byte, and others growing by over 24 bytes. In contrast, SRTP-AERO has data overhead that is constant, and need not be greater than a single byte at equivalent security levels.

6. Security Considerations

6.1. SSRC collisions

With SRTP-AERO, SSRC collisions do not undermine security; instead, there is a limited and quantifiable loss of confidentiality, which is described in Section 11.4 of [I-D.mcgrew-aero]. In essence, if there is an SSRC collision between two senders, then the attacker will be able to detect the event that both senders encrypt two distinct packets that happen to have exactly the same plaintext and associated data values. Even if an SSRC collision occurs, it is unlikely that the RTP sequence number, the RTP timestamp, and the plaintexts will all be identical. Thus, in many setting the unpredictability of the RTP header and payload provide additional protection even in the unlikely occurrence of an SSRC collision.

An SSRC collision will not undermine authentication.

The normal RTP mechanism for detection and correction of SSRC collisions MUST be used. In practice, if an SRTP or SRTCP sender receives a valid SRTP or SRTCP packet that it did not itself originate, which has an SSRC value equal to its own, then it MUST stop using that SSRC value, and select a new SSRC value at random. A packet is valid only if its AERO decryption does not return FAIL.

6.2. Key scope

With SRTP-AERO, a single master key MAY be used with multiple SRTP sources or multiple SRTP receivers within a single session. Scenarios in which there may be multiple sources in a single session include multicast SRTP, as well as an RTP mixer that retransmits packets from one selected source to an entire set of sources. In this latter case, a set of participants in a session can all use a single SRTP master key, and a mixer can selectively retransmit packets, e.g. from the "loudest talker", without re-encrypting the packets.

A single master key MUST NOT be used across distinct SRTP sessions. This property is not specific to AERO, but instead is general to all uses of SRTP, and it follows from the fact that RTP and RTCP receivers have no way of distinguishing between the packets from one session and those from another. This fact would allow an attacker to substitute packets from one session to another, if both sessions were using the same master key.

7. IANA Considerations

This section registers with IANA the following SRTP crypto-suite parameters for SDP Security Descriptions [RFC4568]:

- SRTP_AERO_128_XCB_80
- SRTP_AERO_128_XCB_32
- SRTP_AERO_256_XCB_128

They are specified in Section 4 of this note.

We also request the IANA assignment of the following values to the DTLS SRTPProtectionProfile registry:

SRTP_AERO_128_XCB_80 = { TBD1, TBD2 }

SRTP_AERO_128_XCB_32 = { TBD3, TBD4 }

SRTP_AERO_256_XCB_128 = { TBD5, TBD6 }

We also request the following IANA assignment from the MIKEY registry of SRTP policy parameters:

- An Encryption Algorithm parameter value of TBD.

This value indicates the use AERO_AES_XCB in SRTP, and corresponds to either SRTP_AERO_128_XCB_80, SRTP_AERO_128_XCB_32, and SRTP_AERO_256_XCB_128.

8. Acknowledgements

Thanks are due to Nermeen Ismail for insightful discussions on the use of SRTP in telepresence environments.

9. References

9.1. Normative References

[I-D.ietf-avtcore-srtp-aes-gcm]
McGrew, D. and K. Igoe, "AES-GCM and AES-CCM Authenticated Encryption in Secure RTP (SRTP)", draft-ietf-avtcore-srtp-aes-gcm-10 (work in progress), September 2013.

[I-D.mcgrew-aero]
McGrew, D. and J. Foley, "Authenticated Encryption with Replay prOtection (AERO)", draft-mcgrew-aero-00 (work in progress), October 2013.

[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.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "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.

[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.

[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.

9.2. Informative References

[SRTP-EKT]
McGrew, D., Andreason, F., Wing, D., and K. Fisher, "Encrypted Key Transport for Secure RTP", Internet Draft; Work In Progress. <draft-ietf-avtcore-srtp-ekt-00.txt>.

Authors' Addresses

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

Dan Wing
Cisco Systems, Inc.
510 McCarthy Blvd.
Milpitas, CA 95035
US

Phone: (408) 853 4197
Email: dwing@cisco.com

John Foley
Cisco Systems
7025-2 Kit Creek Road
Research Triangle Park, NC 14987
US

Email: foleyj@cisco.com