Internet Engineering Task Force (IETF)
Request for Comments: 8269
Category: Informational
ISSN: 2070-1721

W. Kim
J. Lee
J. Park
D. Kwon
NSRI
D. Kim
Kookmin Univ.

October 2017

The ARIA Algorithm and Its Use with the Secure Real-Time Transport Protocol (SRTP)

Abstract

This document defines the use of the ARIA block cipher algorithm within the Secure Real-time Transport Protocol (SRTP). It details two modes of operation (CTR and GCM) and the SRTP key derivation functions for ARIA. Additionally, this document defines DTLS-SRTP protection profiles and Multimedia Internet KEYing (MIKEY) parameter sets for use with ARIA.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines the use of the ARIA block cipher algorithm [RFC5794] in the Secure Real-time Transport Protocol (SRTP) [RFC3711] for providing confidentiality for Real-time Transport Protocol (RTP) [RFC3550] traffic and for RTP Control Protocol (RTCP) [RFC3550] traffic.

1.1. ARIA

ARIA is a general-purpose block cipher algorithm developed by Korean cryptographers in 2003. It is an iterated block cipher with 128-, 192-, and 256-bit keys and encrypts 128-bit blocks in 12, 14, and 16 rounds, depending on the key size. It is secure and suitable for most software and hardware implementations on 32-bit and 8-bit processors. It was established as a Korean standard block cipher algorithm in 2004 [ARIAKS] and has been widely used in Korea, especially for government-to-public services. It was included in Public-Key Cryptography Standards (PKCS) #11 in 2007 [ARIAPKCS]. The algorithm specification and object identifiers are described in [RFC5794].

1.2. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Cryptographic Transforms

Block ciphers ARIA and AES share common characteristics including mode, key size, and block size. ARIA does not have any restrictions for modes of operation that are used with this block cipher. We define two modes of running ARIA within SRTP: (1) ARIA in Counter Mode (ARIA-CTR) and (2) ARIA in Galois/Counter Mode (ARIA-GCM).

2.1. ARIA-CTR

Section 4.1.1 of [RFC3711] defines AES-128 counter mode encryption, which it refers to as "AES_CM". Section 2 of [RFC6188] defines "AES_256_CM" in SRTP. ARIA counter modes are defined in the same manner except that each invocation of AES is replaced by that of ARIA [RFC5794] and are denoted by ARIA_128_CTR and ARIA_256_CTR, respectively, according to the key lengths. The plaintext inputs to the block cipher are formed as in AES-CTR (AES_CM, AES_256_CM) and the block cipher outputs are processed as in AES-CTR. Note that, ARIA-CTR MUST be used only in conjunction with an authentication transform.

Section 3.2 of [RFC6904] defines AES-CTR for SRTP header extension keystream generation. When ARIA-CTR is used, the header extension keystream SHALL be generated in the same manner except that each invocation of AES is replaced by that of ARIA [RFC5794].

2.2. ARIA-GCM

Galois/Counter Mode [GCM] [RFC5116] is an Authenticated Encryption with Associated Data (AEAD) block cipher mode. A detailed description of ARIA-GCM is defined similarly as AES-GCM found in [RFC5116] and [RFC5282].

[RFC7714] describes the use of AES-GCM with SRTP. The use of ARIA-GCM with SRTP is defined the same as AES-GCM except that each invocation of AES is replaced by ARIA [RFC5794]. When encryption of header extensions [RFC6904] is in use, a separate keystream to encrypt selected RTP header extension elements MUST be generated in the same manner defined in [RFC7714] except that AES-CTR is replaced by ARIA-CTR.

3. Key Derivation Functions

Section 4.3.3 of [RFC3711] defines the AES-128 counter mode key derivation function, which it refers to as "AES-CM PRF". Section 3 of [RFC6188] defines the AES-256 counter mode key derivation function, which it refers to as "AES_256_CM_PRF". The ARIA-CTR Pseudorandom Function (PRF) is defined in a same manner except that each invocation of AES is replaced by that of ARIA. According to the key lengths of the underlying encryption algorithm, ARIA-CTR PRFs are denoted by "ARIA_128_CTR_PRF" and "ARIA_256_CTR_PRF". The usage requirements of [RFC6188] and [RFC7714] regarding the AES-CM PRF apply to the ARIA-CTR PRF as well.

4. Protection Profiles

This section defines SRTP protection profiles that use the ARIA transforms and key derivation functions defined in this document. The following list indicates the SRTP transform parameters for each protection profile. Those are described for use with DTLS-SRTP [RFC5764].

The parameters cipher_key_length, cipher_salt_length, auth_key_length, and auth_tag_length express the number of bits in the values to which they refer. The maximum_lifetime parameter indicates the maximum number of packets that can be protected with each single set of keys when the parameter profile is in use. All of these parameters apply to both RTP and RTCP, unless the RTCP parameters are separately specified.

```
SRTP_ARIA_128_CTR_HMAC_SHA1_80
        cipher:                    ARIA_128_CTR
        cipher_key_length:         128 bits
        cipher_salt_length:        112 bits
        key derivation function:  ARIA_128_CTR_PRF
        auth_function:             HMAC-SHA1
        auth_key_length:           160 bits
        auth_tag_length:           80 bits
        maximum_lifetime:          at most 2^31 SRTCP packets and
                                   at most 2^48 SRTP packets

SRTP_ARIA_128_CTR_HMAC_SHA1_32
        cipher:                    ARIA_128_CTR
        cipher_key_length:         128 bits
        cipher_salt_length:        112 bits
        key derivation function:  ARIA_128_CTR_PRF
        auth_function:             HMAC-SHA1
        auth_key_length:           160 bits
        SRTP auth_tag_length:      32 bits
        SRTCP auth_tag_length:     80 bits
        maximum_lifetime:          at most 2^31 SRTCP packets and
                                   at most 2^48 SRTP packets

SRTP_ARIA_256_CTR_HMAC_SHA1_80
        cipher:                    ARIA_256_CTR
        cipher_key_length:         256 bits
        cipher_salt_length:        112 bits
        key derivation function:  ARIA_256_CTR_PRF
        auth_function:             HMAC-SHA1
        auth_key_length:           160 bits
        auth_tag_length:           80 bits
        maximum_lifetime:          at most 2^31 SRTCP packets and
                                   at most 2^48 SRTP packets
```

```
SRTP_ARIA_256_CTR_HMAC_SHA1_32
        cipher:                   ARIA_256_CTR
        cipher_key_length:        256 bits
        cipher_salt_length:       112 bits
        key derivation function:  ARIA_256_CTR_PRF
        auth_function:            HMAC-SHA1
        auth_key_length:          160 bits
        SRTP auth_tag_length:     32 bits
        SRTCP auth_tag_length:    80 bits
        maximum_lifetime:         at most 2^31 SRTCP packets and
                                  at most 2^48 SRTP packets

SRTP_AEAD_ARIA_128_GCM
        cipher:                   ARIA_128_GCM
        cipher_key_length:        128 bits
        cipher_salt_length:       96 bits
        aead_auth_tag_length:     128 bits
        auth_function:            NULL
        auth_key_length:          N/A
        auth_tag_length:          N/A
        key derivation function:  ARIA_128_CTR_PRF
        maximum_lifetime:         at most 2^31 SRTCP packets and
                                  at most 2^48 SRTP packets

SRTP_AEAD_ARIA_256_GCM
        cipher:                   ARIA_256_GCM
        cipher_key_length:        256 bits
        cipher_salt_length:       96 bits
        aead_auth_tag_length:     128 bits
        auth_function:            NULL
        auth_key_length:          N/A
        auth_tag_length:          N/A
        key derivation function:  ARIA_256_CTR_PRF
        maximum_lifetime:         at most 2^31 SRTCP packets and
                                  at most 2^48 SRTP packets
```

The ARIA-CTR protection profiles use the same authentication transform that is mandatory to implement in SRTP: HMAC-SHA1 with a 160-bit key.

Note that SRTP protection profiles that use AEAD algorithms do not specify an auth_function, auth_key_length, or auth_tag_length, since they 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.

The PRFs for ARIA protection profiles are defined by ARIA-CTR PRF of the equal key length with the encryption algorithm (see Section 2). SRTP_ARIA_128_CTR_HMAC and SRTP_AEAD_ARIA_128_GCM MUST use the ARIA_128_CTR_PRF key derivation function. And SRTP_ARIA_256_CTR_HMAC and SRTP_AEAD_ARIA_256_GCM MUST use the ARIA_256_CTR_PRF key derivation function.

MIKEY specifies the SRTP protection profile definition separately from the key length (which is specified by the session encryption key length) and the authentication tag length. The DTLS-SRTP [RFC5764] protection profiles are mapped to MIKEY parameter sets as shown below.

	Encryption Algorithm	Encryption Key Length	Auth. Tag Length
SRTP_ARIA_128_CTR_HMAC_80	ARIA-CTR	16 octets	10 octets
SRTP_ARIA_128_CTR_HMAC_32	ARIA-CTR	16 octets	4 octets
SRTP_ARIA_256_CTR_HMAC_80	ARIA-CTR	32 octets	10 octets
SRTP_ARIA_256_CTR_HMAC_32	ARIA-CTR	32 octets	4 octets

Figure 1: Mapping MIKEY Parameters to ARIA-CTR with the HMAC Algorithm

	Encryption Algorithm	Encryption Key Length	AEAD Auth. Tag Length
SRTP_AEAD_ARIA_128_GCM	ARIA-GCM	16 octets	16 octets
SRTP_AEAD_ARIA_256_GCM	ARIA-GCM	32 octets	16 octets

Figure 2: Mapping MIKEY Parameters to the ARIA-GCM Algorithm

5. Security Considerations

At the time of publication of this document, no security problem has been found on ARIA. Previous security analysis results are summarized in [ATY].

The security considerations in [GCM], [RFC3711], [RFC5116], [RFC6188], [RFC6904], and [RFC7714] apply to this document as well. This document includes crypto suites with authentication tags of a length less than 80 bits. These suites MAY be used for certain application contexts where longer authentication tags may be undesirable, for example, those mentioned in [RFC3711], Section 7.5.

Otherwise, short authentication tags SHOULD NOT be used, since they may reduce authentication strength. See [RFC3711], Section 9.5 for a discussion of risks related to weak authentication in SRTP.

At the time of publication of this document, SRTP recommends HMAC-SHA1 as the default and mandatory-to-implement MAC algorithm. All currently registered SRTP crypto suites except the GCM-based ones use HMAC-SHA1 as their HMAC algorithm to provide message authentication. Due to security concerns with SHA-1 [RFC6194], the IETF is gradually moving away from SHA-1 and towards stronger hash algorithms such as SHA-2 or SHA-3 families. For SRTP, however, SHA-1 is only used in the calculation of an HMAC, and no security issue is known for this usage at the time of this publication.

6. IANA Considerations

6.1. DTLS-SRTP

DTLS-SRTP [RFC5764] defines a DTLS-SRTP "SRTP protection profile". In order to allow the use of the algorithms defined in this document in DTLS-SRTP, IANA has added the following protection profiles below to the "DTLS-SRTP Protection Profiles" registry (see <http://www.iana.org/assignments/srtp-protection/>) created by [RFC5764]:

```
SRTP_ARIA_128_CTR_HMAC_SHA1_80 = {0x00, 0x0B}
SRTP_ARIA_128_CTR_HMAC_SHA1_32 = {0x00, 0x0C}
SRTP_ARIA_256_CTR_HMAC_SHA1_80 = {0x00, 0x0D}
SRTP_ARIA_256_CTR_HMAC_SHA1_32 = {0x00, 0x0E}
SRTP_AEAD_ARIA_128_GCM = {0x00, 0x0F}
SRTP_AEAD_ARIA_256_GCM = {0x00, 0x10}
```

6.2. MIKEY

[RFC3830] and [RFC5748] define encryption algorithms and PRFs for the SRTP policy in MIKEY. In order to allow the use of the algorithms defined in this document in MIKEY, IANA has updated the "Multimedia Internet KEYing (MIKEY) Payload Name Spaces" registry (see <http://www.iana.org/assignments/mikey-payloads/>.)

IANA has registered the following two encryption algorithms in the "Encryption algorithm (Value 0)" subregistry within the "MIKEY Security Protocol Parameters" registry:

SRTP encr alg	Value
ARIA-CTR	7
ARIA-GCM	8

The default session encryption key length is 16 octets.

IANA has registered the following PRF in the "SRTP Pseudo Random Function (Value 5)" subregistry within the "MIKEY Security Protocol Parameters" registry:

SRTP PRF	Value
ARIA-CTR	2

7. References

7.1. Normative References

[GCM] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special publication 800-38D, DOI 10.6028/NIST.SP.800-38D, November 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, DOI 10.17487/RFC3830, August 2004, <https://www.rfc-editor.org/info/rfc3830>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <https://www.rfc-editor.org/info/rfc5116>.

[RFC5282] Black, D. and D. McGrew, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", RFC 5282, DOI 10.17487/RFC5282, August 2008, <https://www.rfc-editor.org/info/rfc5282>.

[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, DOI 10.17487/RFC5764, May 2010, <https://www.rfc-editor.org/info/rfc5764>.

[RFC5794] Lee, J., Lee, J., Kim, J., Kwon, D., and C. Kim, "A Description of the ARIA Encryption Algorithm", RFC 5794, DOI 10.17487/RFC5794, March 2010, <https://www.rfc-editor.org/info/rfc5794>.

[RFC6188] McGrew, D., "The Use of AES-192 and AES-256 in Secure RTP", RFC 6188, DOI 10.17487/RFC6188, March 2011, <https://www.rfc-editor.org/info/rfc6188>.

[RFC6904] Lennox, J., "Encryption of Header Extensions in the Secure Real-time Transport Protocol (SRTP)", RFC 6904, DOI 10.17487/RFC6904, April 2013, <https://www.rfc-editor.org/info/rfc6904>.

[RFC7714] McGrew, D. and K. Igoe, "AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP)", RFC 7714, DOI 10.17487/RFC7714, December 2015, <https://www.rfc-editor.org/info/rfc7714>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[ARIAKS] Korean Agency for Technology and Standards, "128 bit block encryption algorithm ARIA - Part 1: General (in Korean)", KS X 1213-1:2014, December 2014.

[ARIAPKCS] RSA Laboratories, "Additional PKCS #11 Mechanisms", PKCS #11 v2.20, Amendment 3, Revision 1, January 2007.

[ATY] Abdelkhalek, A., Tolba, M., and A. Youssef, "Improved Linear Cryptanalysis of Round-Reduced ARIA", Information Security - ISC 2016, Lecture Notes in Computer Science (LNCS), Vol. 9866, pp. 18-34, DOI 10.1007/978-3-319-45871-7_2, September 2016.

[RFC5748] Yoon, S., Jeong, J., Kim, H., Jeong, H., and Y. Won, "IANA Registry Update for Support of the SEED Cipher Algorithm in Multimedia Internet KEYing (MIKEY)", RFC 5748, DOI 10.17487/RFC5748, August 2010, <https://www.rfc-editor.org/info/rfc5748>.

[RFC6194] Polk, T., Chen, L., Turner, S., and P. Hoffman, "Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithms", RFC 6194, DOI 10.17487/RFC6194, March 2011, <https://www.rfc-editor.org/info/rfc6194>.

Appendix A. Test Vectors

All values are in hexadecimal and represented by the network order (called big endian).

A.1. ARIA-CTR Test Vectors

Common values are organized as follows:

```
Rollover Counter:          00000000
Sequence Number:           315e
SSRC:                      20e8f5eb
Authentication Key:        f93563311b354748c978913795530631
                           16452309
Session Salt:              cd3a7c42c671e0067a2a2639b43a
Initialization Vector:     cd3a7c42e69915ed7a2a263985640000
RTP Header:                8008315ebf2e6fe020e8f5eb
RTP Payload:               f57af5fd4ae19562976ec57a5a7ad55a
                           5af5c5e5c5fdf5c55ad57a4a7272d572
                           62e9729566ed66e97ac54a4a5a7ad5e1
                           5ae5fdd5fd5ac5d56ae56ad5c572d54a
                           e54ac55a956afd6aed5a4ac562957a95
                           16991691d572fd14e97ae962ed7a9f4a
                           955af572e162f57a956666e17ae1f54a
                           95f566d54a66e16e4afd6a9f7ae1c5c5
                           5ae5d56afde916c5e94a6ec56695e14a
                           fde1148416e94ad57ac5146ed59d1cc5
```

Note:
SSRC = Synchronization Source

A.1.1. SRTP_ARIA_128_CTR_HMAC_SHA1_80

```
Session Key:               0c5ffd37a11edc42c325287fc0604f2e

Encrypted RTP Payload:     1bf753f412e6f35058cc398dc851aae3
                           a6ccdcb463fbed9cfb3de2fb76fdffa9
                           e481f5efb64c92487f59dabbc7cc72da
                           092485f3fbad87888820b86037311fa4
                           4330e18a59a1e1338ba2c21458493a57
                           463475c54691f91cec785429119e0dfc
                           d9048f90e07fecd50b528e8c62ee6e71
                           445de5d7f659405135aff3604c2ca4ff
                           4aaca40809cb9eee42cc4ad232307570
                           81ca289f2851d3315e9568b501fdce6d
```

```
   Authenticated Portion || Rollover Counter:
                              8008315ebf2e6fe020e8f5eb1bf753f4
                              12e6f35058cc398dc851aae3a6ccdcb4
                              63fbed9cfb3de2fb76fdffa9e481f5ef
                              b64c92487f59dabbc7cc72da092485f3
                              fbad87888820b86037311fa44330e18a
                              59a1e1338ba2c21458493a57463475c5
                              4691f91cec785429119e0dfcd9048f90
                              e07fecd50b528e8c62ee6e71445de5d7
                              f659405135aff3604c2ca4ff4aaca408
                              09cb9eee42cc4ad23230757081ca289f
                              2851d3315e9568b501fdce6d00000000

   Authentication Tag:        f9de4e729054672b0e35
```

A.1.2. SRTP_ARIA_256_CTR_HMAC_SHA1_80

```
   Session Key:               0c5ffd37a11edc42c325287fc0604f2e
                              3e8cd5671a00fe3216aa5eb105783b54

   Encrypted RTP Payload:     c424c59fd5696305e5b13d8e8ca76566
                              17ccd7471088af9debf07b55c750f804
                              a5ac2b737be48140958a9b420524112a
                              e72e4da5bca59d2b1019ddd7dbdc30b4
                              3d5f046152ced40947d62d2c93e7b8e5
                              0f02db2b6b61b010e4c1566884de1fa9
                              702cdf8157e8aedfe3dd77c76bb50c25
                              ae4d624615c15acfdeeb5f79482aaa01
                              d3e4c05eb601eca2bd10518e9d46b021
                              16359232e9eac0fabd05235dd09e6dea

   Authenticated Portion || Rollover Counter:
                              8008315ebf2e6fe020e8f5ebc424c59f
                              d5696305e5b13d8e8ca7656617ccd747
                              1088af9debf07b55c750f804a5ac2b73
                              7be48140958a9b420524112ae72e4da5
                              bca59d2b1019ddd7dbdc30b43d5f0461
                              52ced40947d62d2c93e7b8e50f02db2b
                              6b61b010e4c1566884de1fa9702cdf81
                              57e8aedfe3dd77c76bb50c25ae4d6246
                              15c15acfdeeb5f79482aaa01d3e4c05e
                              b601eca2bd10518e9d46b02116359232
                              e9eac0fabd05235dd09e6dea00000000

   Authentication Tag:        192f515fab04bbb4e62c
```

A.2. ARIA-GCM Test Vectors

Common values are organized as follows:

```
Rollover Counter:           00000000
Sequence Number:            315e
SSRC:                       20e8f5eb
Encryption Salt:            000000000000000000000000

Initialization Vector:      000020e8f5eb00000000315e
RTP Payload:                f57af5fd4ae19562976ec57a5a7ad55a
                            5af5c5e5c5fdf5c55ad57a4a7272d572
                            62e9729566ed66e97ac54a4a5a7ad5e1
                            5ae5fdd5fd5ac5d56ae56ad5c572d54a
                            e54ac55a956afd6aed5a4ac562957a95
                            16991691d572fd14e97ae962ed7a9f4a
                            955af572e162f57a956666e17ae1f54a
                            95f566d54a66e16e4afd6a9f7ae1c5c5
                            5ae5d56afde916c5e94a6ec56695e14a
                            fde1148416e94ad57ac5146ed59d1cc5
Associated Data:            8008315ebf2e6fe020e8f5eb
```

The encrypted RTP payload is longer than the RTP payload by exactly the GCM authentication tag length (16 octets).

A.2.1. SRTP_AEAD_ARIA_128_GCM

```
Key:                        e91e5e75da65554a48181f3846349562

Encrypted RTP Payload:      4d8a9a0675550c704b17d8c9ddc81a5c
                            d6f7da34f2fe1b3db7cb3dfb9697102e
                            a0f3c1fc2dbc873d44bceeae8e444297
                            4ba21ff6789d3272613fb9631a7cf3f1
                            4bacbeb421633a90ffbe58c2fa6bdca5
                            34f10d0de0502ce1d531b6336e588782
                            78531e5c22bc6c85bbd784d78d9e680a
                            a19031aaf89101d669d7a3965c1f7e16
                            229d7463e0535f4e253f5d18187d40b8
                            ae0f564bd970b5e7e2adfb211e89a953
                            5abace3f37f5a736f4be984bbffbedc1
```

A.2.2. SRTP_AEAD_ARIA_256_GCM

```
Key:                       0c5ffd37a11edc42c325287fc0604f2e
                           3e8cd5671a00fe3216aa5eb105783b54

Encrypted RTP Payload:     6f9e4bcbc8c85fc0128fb1e4a0a20cb9
                           932ff74581f54fc013dd054b19f99371
                           425b352d97d3f337b90b63d1b082adee
                           ea9d2d7391897d591b985e55fb50cb53
                           50cf7d38dc27dda127c078a149c8eb98
                           083d66363a46e3726af217d3a00275ad
                           5bf772c7610ea4c23006878f0ee69a83
                           97703169a419303f40b72e4573714d19
                           e2697df61e7c7252e5abc6bade876ac4
                           961bfac4d5e867afca351a48aed52822
                           e210d6ced2cf430ff841472915e7ef48
```

A.3. Key Derivation Test Vectors

This section provides test vectors for the default key derivation function that uses ARIA in Counter Mode. In the following, we walk through the initial key derivation for the ARIA Counter Mode cipher that requires a session encryption key of 16/24/32 octets according to the session encryption key length, a 14-octet session salt, and an authentication function that requires a 94-octet session authentication key. These values are called the cipher key, the cipher salt, and the auth key in the following. The test vectors are generated in the same way with the test vectors of key derivation functions in [RFC3711] and [RFC6188] but with each invocation of AES replaced with an invocation of ARIA.

A.3.1. ARIA_128_CTR_PRF

The inputs to the key derivation function are the 16-octet master key and the 14-octet master salt:

```
master key:  e1f97a0d3e018be0d64fa32c06de4139
master salt: 0ec675ad498afeebb6960b3aabe6

index DIV kdr:                 000000000000
label:                       00
master salt:   0ec675ad498afeebb6960b3aabe6
-----------------------------------------------
xor:           0ec675ad498afeebb6960b3aabe6     (x, PRF input)

x*2^16:        0ec675ad498afeebb6960b3aabe60000 (ARIA-CTR input)

cipher key:    dbd85a3c4d9219b3e81f7d942e299de4 (ARIA-CTR output)
```

ARIA-CTR protection profile requires a 14-octet cipher salt while ARIA-GCM protection profile requires a 12-octet cipher salt.

```
index DIV kdr:                 000000000000
label:                       02
master salt:   0ec675ad498afeebb6960b3aabe6
-----------------------------------------------
xor:           0ec675ad498afee9b6960b3aabe6     (x, PRF input)

x*2^16:        0ec675ad498afee9b6960b3aabe60000 (ARIA-CTR input)

               9700657f5f34161830d7d85f5dc8be7f (ARIA-CTR output)

cipher salt:   9700657f5f34161830d7d85f5dc8     (ARIA-CTR profile)
               9700657f5f34161830d7d85f         (ARIA-GCM profile)
index DIV kdr:                 000000000000
label:                       01
master salt:   0ec675ad498afeebb6960b3aabe6
-----------------------------------------------
xor:           0ec675ad498afeeab6960b3aabe6     (x, PRF input)

x*2^16:        0ec675ad498afeeab6960b3aabe60000 (ARIA-CTR input)
```

Below, the auth key is shown on the left, while the corresponding ARIA input blocks are shown on the right.

```
auth key                          ARIA input blocks

d021877bd3eaf92d581ed70ddc050e03  0ec675ad498afeeab6960b3aabe60000
f11257032676f2a29f57b21abd3a1423  0ec675ad498afeeab6960b3aabe60001
769749bdc5dd9ca5b43ca6b6c1f3a7de  0ec675ad498afeeab6960b3aabe60002
4047904bcf811f601cc03eaa5d7af6db  0ec675ad498afeeab6960b3aabe60003
9f88efa2e51ca832fc2a15b126fa7be2  0ec675ad498afeeab6960b3aabe60004
469af896acb1852c31d822c45799      0ec675ad498afeeab6960b3aabe60005
```

A.3.2. ARIA_256_CTR_PRF

The inputs to the key derivation function are the 32-octet master key and the 14-octet master salt:

```
  master key:  0c5ffd37a11edc42c325287fc0604f2e
               3e8cd5671a00fe3216aa5eb105783b54
  master salt: 0ec675ad498afeebb6960b3aabe6

  index DIV kdr:                000000000000
  label:                      00
  master salt: 0ec675ad498afeebb6960b3aabe6
  ----------------------------------------------
  xor:         0ec675ad498afeebb6960b3aabe6     (x, PRF input)

  x*2^16:      0ec675ad498afeebb6960b3aabe60000 (ARIA-CTR input)

  cipher key:  0649a09d93755fe9c2b2efba1cce930a (ARIA-CTR 1st output)
               f2e76ce8b77e4b175950321aa94b0cf4 (ARIA-CTR 2nd output)
```

ARIA-CTR protection profile requires a 14-octet cipher salt while ARIA-GCM protection profile requires a 12-octet cipher salt.

```
  index DIV kdr:                 000000000000
  label:                       02
  master salt:  0ec675ad498afeebb6960b3aabe6
  ---------------------------------------------
  xor:          0ec675ad498afee9b6960b3aabe6     (x, PRF input)

  x*2^16:       0ec675ad498afee9b6960b3aabe60000 (ARIA-CTR input)

                194abaa8553a8eba8a413a340fc80a3d (ARIA-CTR output)

  cipher salt:  194abaa8553a8eba8a413a340fc8     (ARIA-CTR profile)
                194abaa8553a8eba8a413a34         (ARIA-GCM profile)

  index DIV kdr:                 000000000000
  label:                       01
  master salt:  0ec675ad498afeebb6960b3aabe6
  ----------------------------------------------
  xor:          0ec675ad498afeeab6960b3aabe6     (x, PRF input)

  x*2^16:       0ec675ad498afeeab6960b3aabe60000 (ARIA-CTR input)
```

Below, the auth key is shown on the left, while the corresponding ARIA input blocks are shown on the right.

```
auth key                            ARIA input blocks

e58d42915873b71899234807334658f2    0ec675ad498afeeab6960b3aabe60000
0bc460181d06e02b7a9e60f02ff10bfc    0ec675ad498afeeab6960b3aabe60001
9ade3795cf78f3e0f2556d9d913470c4    0ec675ad498afeeab6960b3aabe60002
e82e45d254bfb8e2933851a3930ffe7d    0ec675ad498afeeab6960b3aabe60003
fca751c03ec1e77e35e28dac4f17d1a5    0ec675ad498afeeab6960b3aabe60004
80bdac028766d3b1e8f5a41faa3c        0ec675ad498afeeab6960b3aabe60005
```

Authors' Addresses

Woo-Hwan Kim
National Security Research Institute
P.O. Box 1, Yuseong
Daejeon 34188
Korea

Email: whkim5@nsr.re.kr

Jungkeun Lee
National Security Research Institute
P.O. Box 1, Yuseong
Daejeon 34188
Korea

Email: jklee@nsr.re.kr

Je-Hong Park
National Security Research Institute
P.O. Box 1, Yuseong
Daejeon 34188
Korea

Email: jhpark@nsr.re.kr

Daesung Kwon
National Security Research Institute
P.O. Box 1, Yuseong
Daejeon 34188
Korea

Email: ds_kwon@nsr.re.kr

Dong-Chan Kim
Kookmin University
77 Jeongneung-ro, Seongbuk-gu
Seoul 02707
Korea

Email: dckim@kookmin.ac.kr