Network Working Group
Request for Comments: 5528
Category: Informational

A. Kato
NTT Software Corporation
M. Kanda
NTT
S. Kanno
NTT Software Corporation

April 2009

Camellia Counter Mode and Camellia Counter with CBC-MAC Mode Algorithms

Status of This Memo

Copyright Notice

Abstract

This document describes the algorithms and presents test vectors for the Camellia block cipher algorithm in Counter mode (CTR) and Counter with Cipher Block Chaining MAC mode (CCM). The purpose of this document is to make the Camellia-CTR and Camellia-CCM algorithm conveniently available to the Internet Community.

Table of Contents

1. Introduction

This document describes the use of the Camellia block cipher algorithm in Counter (CTR) mode and Counter with CBC-MAC (CCM) mode.

Camellia is a symmetric cipher with a Feistel structure. Camellia was developed jointly by NTT and Mitsubishi Electric Corporation in 2000. It was designed to withstand all known cryptanalytic attacks, and it has been scrutinized by worldwide cryptographic experts. Camellia is suitable for implementation in software and hardware, offering encryption speed in software and hardware implementations that is comparable to Advanced Encryption Standard (AES) [5].

Camellia supports 128-bit block size and 128-, 192-, and 256-bit key lengths, i.e., the same interface specifications as the AES. Therefore, it is easy to implement Camellia-based algorithms by replacing the AES block of AES-based algorithms with a Camellia block.

Camellia already has been adopted by the IETF and other international standardization organizations; in particular, the IETF has published specifications for the use of Camellia with IPsec [6], TLS [7], Secure/Multipurpose Internet Mail Extensions (S/MIME) [8], and XML Securiy [9]. Camellia is one of the three ISO/IEC international standard [10] 128-bit block ciphers (Camellia, AES, and Super Effective and Efficient Delivery (SEED)). Camellia was selected as a recommended cryptographic primitive by the EU NESSIE (New European Schemes for Signatures, Integrity and Encryption) project [11] and

was included in the list of cryptographic techniques for Japanese e-Government systems that was selected by the Japanese CRYPTREC (Cryptography Research and Evaluation Committees) [12].

Since optimized source code is provided under several open source licenses [13], Camellia has also been adopted by several open source projects (OpenSSL, FreeBSD, Linux, and Firefox).

The algorithm specification and object identifiers are described in [1].

The Camellia web site [14] contains a wealth of information about Camellia, including detailed specification, security analysis, performance figures, reference implementation, optimized implementation, test vectors (TVs), and intellectual property information.

1.1. Terminology

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 RFC 2119 [2].

All multi-octet values in this document are encoded and represented in network byte order, i.e., most significant octet first.

2. The Camellia Cipher Algorithm

All symmetric block cipher algorithms share common characteristics and variables, including mode, key size, weak keys, and block size. The following sections contain descriptions of the relevant characteristics of Camellia.

2.1. Key Size

Camellia supports three key sizes: 128 bits, 192 bits, and 256 bits. The default key size is 128 bits, and all implementations MUST support this key size. Implementations MAY also support key sizes of 192 bits and 256 bits.

2.2. Weak Keys

At the time of writing this document, there are no known weak keys for Camellia.

2.3. Block Size and Padding

Camellia uses a block size of 16 octets (128 bits).

Padding is required by the algorithm to maintain a 16-octet (128-bit) block size. Padding MUST be added such that the data to be encrypted has a length that is a multiple of 16 octets.

Because of the algorithm-specific padding requirement, no additional padding is required to ensure that the ciphertext terminates on a 4-octet boundary (i.e., maintaining a 16-octet block size guarantees that the Encapsulating Security Payload (ESP) Pad Length and Next Header fields will be right aligned within a 4-octet word). Additional padding MAY be included as long as the 16-octet block size is maintained.

2.4. Performance

Performance figures for Camellia are available at [14]. The NESSIE project has reported on the performance of optimized implementations independently [11].

3. Modes of Operation

Camellia Counter (Camellia-CTR) mode and Camellia Counter with CBC-MAC (Camellia-CCM) mode are based on [3][15][4].

CTR mode [3] behaves like a stream cipher, but is based on a block cipher primitive (that is, CTR mode operation of a block cipher results in a stream cipher).

CCM mode [15][4] is a generic authenticate-and-encrypt block cipher mode. In this specification, CCM is used with the Camellia [1] block cipher.

4. Test Vectors

4.1. Camellia-CTR

This section contains nine TVs, which can be used to confirm that an implementation has correctly implemented Camellia-CTR. The first three TVs use Camellia with a 128-bit key; the next three TVs use Camellia with a 192-bit key; and the last three TVs use Camellia with a 256-bit key.

```
   TV #1: Encrypting 16 octets using Camellia-CTR with 128-bit key
   Camellia Key     : AE 68 52 F8 12 10 67 CC 4B F7 A5 76 55 77 F3 9E
   Camellia-CTR IV  : 00 00 00 00 00 00 00 00
   Nonce            : 00 00 00 30
   Plaintext        : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
   Counter Block (1): 00 00 00 30 00 00 00 00 00 00 00 00 00 00 00 01
   Key Stream    (1): 83 F4 AC FD EE 71 41 F8 4C E8 1F 1D FB 72 78 58
   Ciphertext       : D0 9D C2 9A 82 14 61 9A 20 87 7C 76 DB 1F 0B 3F

   TV #2: Encrypting 32 octets using Camellia-CTR with 128-bit key
   Camellia Key     : 7E 24 06 78 17 FA E0 D7 43 D6 CE 1F 32 53 91 63
   Camellia-CTR IV  : C0 54 3B 59 DA 48 D9 0B
   Nonce            : 00 6C B6 DB
   Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                    : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
   Counter Block (1): 00 6C B6 DB C0 54 3B 59 DA 48 D9 0B 00 00 00 01
   Key Stream    (1): DB F2 C5 8E C4 86 90 D3 D2 75 9A 7C 69 B6 C5 4B
   Counter Block (2): 00 6C B6 DB C0 54 3B 59 DA 48 D9 0B 00 00 00 02
   Key Stream    (2): 3B 9F 9C 1C 25 E5 CA B0 34 6D 0D F8 4F 7D FE 57
   Ciphertext       : DB F3 C7 8D C0 83 96 D4 DA 7C 90 77 65 BB CB 44
                    : 2B 8E 8E 0F 31 F0 DC A7 2C 74 17 E3 53 60 E0 48

   TV #3: Encrypting 36 octets using Camellia-CTR with 128-bit key
   Camellia Key     : 76 91 BE 03 5E 50 20 A8 AC 6E 61 85 29 F9 A0 DC
   Camellia-CTR IV  : 27 77 7F 3F 4A 17 86 F0
   Nonce            : 00 E0 01 7B
   Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                    : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
                    : 20 21 22 23
   Counter Block (1): 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 01
   Key Stream    (1): B1 9C 1D CE CF 70 ED 8F 27 8D 96 E9 41 88 C1 7C
   Counter Block (2): 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 02
   Key Stream    (2): 8C F7 59 38 48 88 65 E6 57 34 47 86 D2 85 97 D2
   Counter Block (3): 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 03
   Key Stream    (3): FF 71 A4 B5 D8 86 12 53 6A 9D 10 A1 13 0F 14 F8
   Ciphertext       : B1 9D 1F CD CB 75 EB 88 2F 84 9C E2 4D 85 CF 73
                    : 9C E6 4B 2B 5C 9D 73 F1 4F 2D 5D 9D CE 98 89 CD
                    : DF 50 86 96

   TV #4: Encrypting 16 octets using Camellia-CTR with 192-bit key
   Camellia Key     : 16 AF 5B 14 5F C9 F5 79 C1 75 F9 3E 3B FB 0E ED
                    : 86 3D 06 CC FD B7 85 15
   Camellia-CTR IV  : 36 73 3C 14 7D 6D 93 CB
   Nonce            : 00 00 00 48
   Plaintext        : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
   Counter Block (1): 00 00 00 48 36 73 3C 14 7D 6D 93 CB 00 00 00 01
   Key Stream    (1): 70 10 57 F9 E6 E8 0B 49 7A 1F 4C AC AB F3 E5 F1
   Ciphertext       : 23 79 39 9E 8A 8D 2B 2B 16 70 2F C7 8B 9E 96 96
```

```
   TV #5: Encrypting 32 octets using Camellia-CTR with 192-bit key
   Camellia Key     : 7C 5C B2 40 1B 3D C3 3C 19 E7 34 08 19 E0 F6 9C
                    : 67 8C 3D B8 E6 F6 A9 1A
   Camellia-CTR IV  : 02 0C 6E AD C2 CB 50 0D
   Nonce            : 00 96 B0 3B
   Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                    : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
   Counter Block (1): 00 96 B0 3B 02 0C 6E AD C2 CB 50 0D 00 00 00 01
   Key Stream    (1): 7D EE 36 F4 A1 D5 E2 12 6F 42 75 F7 A2 6A C9 52
   Counter Block (2): 00 96 B0 3B 02 0C 6E AD C2 CB 50 0D 00 00 00 02
   Key Stream    (2): C0 09 AA 7C E6 25 47 F7 4E 20 30 82 EF 47 52 F2
   Ciphertext       : 7D EF 34 F7 A5 D0 E4 15 67 4B 7F FC AE 67 C7 5D
                    : D0 18 B8 6F F2 30 51 E0 56 39 2A 99 F3 5A 4C ED

   TV #6: Encrypting 36 octets using Camellia-CTR with 192-bit key
   Camellia Key     : 02 BF 39 1E E8 EC B1 59 B9 59 61 7B 09 65 27 9B
                    : F5 9B 60 A7 86 D3 E0 FE
   Camellia-CTR IV  : 5C BD 60 27 8D CC 09 12
   Nonce            : 00 07 BD FD
   Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                    : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
                    : 20 21 22 23
   Counter Block (1): 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 01
   Key Stream    (1): 57 11 E7 55 E5 4D 7C 27 BD A5 04 78 FD 93 40 77
   Counter Block (2): 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 02
   Key Stream    (2): 66 E2 6D CF 85 A4 F9 5A 55 B4 F2 FD 7A BB 53 11
   Counter Block (3): 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 03
   Key Stream    (3): F5 76 89 74 63 52 A8 C5 1E 82 DE 66 C3 9F 38 34
   Ciphertext       : 57 10 E5 56 E1 48 7A 20 B5 AC 0E 73 F1 9E 4E 78
                    : 76 F3 7F DC 91 B1 EF 4D 4D AD E8 E6 66 A6 4D 0E
                    : D5 57 AB 57

   TV #7: Encrypting 16 octets using Camellia-CTR with 256-bit key
   Camellia Key     : 77 6B EF F2 85 1D B0 6F 4C 8A 05 42 C8 69 6F 6C
                    : 6A 81 AF 1E EC 96 B4 D3 7F C1 D6 89 E6 C1 C1 04
   Camellia-CTR IV  : DB 56 72 C9 7A A8 F0 B2
   Nonce            : 00 00 00 60
   Plaintext        : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
   Counter Block (1): 00 00 00 60 DB 56 72 C9 7A A8 F0 B2 00 00 00 01
   Key Stream    (1): 67 68 97 AF 48 1B DF AC D1 06 F7 1A 6C 76 C8 76
   Ciphertext       : 34 01 F9 C8 24 7E FF CE BD 69 94 71 4C 1B BB 11
```

```
TV #8: Encrypting 32 octets using Camellia-CTR with 256-bit key
Camellia Key     : F6 D6 6D 6B D5 2D 59 BB 07 96 36 58 79 EF F8 86
                 : C6 6D D5 1A 5B 6A 99 74 4B 50 59 0C 87 A2 38 84
Camellia-CTR IV  : C1 58 5E F1 5A 43 D8 75
Nonce            : 00 FA AC 24
Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                 : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Counter Block (1): 00 FA AC 24 C1 58 5E F1 5A 43 D8 75 00 00 00 01
Key Stream    (1): D6 C2 01 91 20 6A 7E 0F A0 35 21 29 A4 8E 90 4A
Counter Block (2): 00 FA AC 24 C1 58 5E F1 5A 43 D8 75 00 00 00 02
Key Stream    (2): F5 0D C6 99 08 CA 56 79 A4 85 D8 C8 B7 9E 5F 17
Ciphertext       : D6 C3 03 92 24 6F 78 08 A8 3C 2B 22 A8 83 9E 45
                 : E5 1C D4 8A 1C DF 40 6E BC 9C C2 D3 AB 83 41 08

TV #9: Encrypting 36 octets using Camellia-CTR with 256-bit key
Camellia Key     : FF 7A 61 7C E6 91 48 E4 F1 72 6E 2F 43 58 1D E2
                 : AA 62 D9 F8 05 53 2E DF F1 EE D6 87 FB 54 15 3D
Camellia-CTR IV  : 51 A5 1D 70 A1 C1 11 48
Nonce            : 00 1C C5 B7
Plaintext        : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                 : 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
                 : 20 21 22 23
Counter Block (1): 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 01
Key Stream    (1): A4 DB 21 FF E2 A0 F9 AD 65 6D A4 91 0A 5F AA 23
Counter Block (2): 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 02
Key Stream    (2): C1 70 B1 58 71 EC 71 88 6D D9 05 0B 03 6C 39 70
Counter Block (3): 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 03
Key Stream    (3): 35 CE 2F AE 90 78 B3 72 F5 76 12 39 1F 8B AF BF
Ciphertext       : A4 DA 23 FC E6 A5 FF AA 6D 64 AE 9A 06 52 A4 2C
                 : D1 61 A3 4B 65 F9 67 9F 75 C0 1F 10 1F 71 27 6F
                 : 15 EF 0D 8D
```

4.2. Camellia-CCM

This section contains twenty four TVs, which can be used to confirm that an implementation has correctly implemented Camellia-CCM. In each of these TVs, the least significant sixteen bits of the counter block is used for the block counter, and the nonce is 13 octets. Some of the TVs include an eight octet authentication value, and others include a ten octet authentication value.

```
   =============== Packet Vector #1 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 03  02 01 00 A0  A1 A2 A3 A4  A5
   Total packet length =   31. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E
   CBC IV in: 59 00 00 00  03 02 01 00  A0 A1 A2 A3  A4 A5 00 17
   CBC IV out:D4 DB CD 92  A8 96 41 56  1D 0D BB D0  D5 7F 7E 1D
   After xor: D4 D3 CD 93  AA 95 45 53  1B 0A BB D0  D5 7F 7E 1D   [hdr]
   After CAM: BD 84 03 80  73 59 37 B7  CE F5 E4 BA  1B 18 54 DC
   After xor: B5 8D 09 8B  7F 54 39 B8  DE E4 F6 A9  0F 0D 42 CB   [msg]
   After CAM: CE 21 82 9C  F6 F2 4D A2  CB 35 D1 FD  81 27 63 EC
   After xor: D6 38 98 87  EA EF 53 A2  CB 35 D1 FD  81 27 63 EC   [msg]
   After CAM: 20 11 FE E2  53 B1 A7 DB  02 77 FA 37  6D 78 EE 10
   MIC tag  : 20 11 FE E2  53 B1 A7 DB
   CTR Start: 01 00 00 00  03 02 01 00  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: B2 7A 7B 8E  EB 14 3F 0B  82 E2 98 4C  06 44 CC 42
   CTR[0002]: E2 E2 D3 52  98 97 13 45  D1 63 22 90  E7 F8 15 4A
   CTR[MIC ]: DC BF 30 96  38 8C 1E 76
   Total packet length =   39. [Encrypted]
              00 01 02 03  04 05 06 07  BA 73 71 85  E7 19 31 04
              92 F3 8A 5F  12 51 DA 55  FA FB C9 49  84 8A 0D FC
              AE CE 74 6B  3D B9 AD

   =============== Packet Vector #2 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 04  03 02 01 A0  A1 A2 A3 A4  A5
   Total packet length =   32. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
   CBC IV in: 59 00 00 00  04 03 02 01  A0 A1 A2 A3  A4 A5 00 18
   CBC IV out:07 0B 22 50  8A 24 3C DD  5B BA 54 DB  60 52 88 06
   After xor: 07 03 22 51  88 27 38 D8  5D BD 54 DB  60 52 88 06   [hdr]
   After CAM: 10 FD C2 F2  90 4A 9F 96  B0 4F 62 A4  A1 A9 31 1E
   After xor: 18 F4 C8 F9  9C 47 91 99  A0 5E 70 B7  B5 BC 27 09   [msg]
   After CAM: E4 C8 82 02  89 55 5C 15  CE 7F E4 60  B1 B9 5A 08
   After xor: FC D1 98 19  95 48 42 0A  CE 7F E4 60  B1 B9 5A 08   [msg]
   After CAM: D2 96 BA 4F  83 DE B5 DF  A2 19 08 F7  47 4E 3C 40
   MIC tag  : D2 96 BA 4F  83 DE B5 DF
   CTR Start: 01 00 00 00  04 03 02 01  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 55 2C 6E B4  82 A2 EF D6  85 37 FE 12  79 0E E6 55
   CTR[0002]: 54 E2 C8 D6  7E 99 91 2C  F2 8A D7 8E  83 04 10 36
   CTR[MIC ]: B2 24 93 12  71 9C 36 37
   Total packet length =   40. [Encrypted]
              00 01 02 03  04 05 06 07  5D 25 64 BF  8E AF E1 D9
              95 26 EC 01  6D 1B F0 42  4C FB D2 CD  62 84 8F 33
              60 B2 29 5D  F2 42 83 E8
```

```
   =============== Packet Vector #3 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 05  04 03 02 A0  A1 A2 A3 A4  A5
   Total packet length =   33. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
              20
   CBC IV in: 59 00 00 00  05 04 03 02  A0 A1 A2 A3  A4 A5 00 19
   CBC IV out:6F 69 15 DF  A6 A0 DF 24  84 A7 37 88  A3 65 F9 2E
   After xor: 6F 61 15 DE  A4 A3 DB 21  82 A0 37 88  A3 65 F9 2E   [hdr]
   After CAM: 59 5D 99 48  79 04 DA C9  13 93 36 C9  11 A8 09 1D
   After xor: 51 54 93 43  75 09 D4 C6  03 82 24 DA  05 BD 1F 0A   [msg]
   After CAM: 1A 43 D7 19  65 43 97 C1  43 6F 4F 11  A7 6C 6B ED
   After xor: 02 5A CD 02  79 5E 89 DE  63 6F 4F 11  A7 6C 6B ED   [msg]
   After CAM: 30 0B 06 8A  A0 D1 4D C5  9E 44 22 84  82 45 42 0B
   MIC tag  : 30 0B 06 8A  A0 D1 4D C5
   CTR Start: 01 00 00 00  05 04 03 02  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 89 FF 69 DD  CB 75 76 18  E9 31 24 1B  AD 97 BB 02
   CTR[0002]: C4 32 A7 9C  CB 4B E9 8D  24 A8 F0 AB  C6 87 16 11
   CTR[MIC ]: C5 5A D0 E2  8F F2 E7 83
   Total packet length =   41. [Encrypted]
              00 01 02 03  04 05 06 07  81 F6 63 D6  C7 78 78 17
              F9 20 36 08  B9 82 AD 15  DC 2B BD 87  D7 56 F7 92
              04 F5 51 D6  68 2F 23 AA  46

   =============== Packet Vector #4 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 06  05 04 03 A0  A1 A2 A3 A4  A5
   Total packet length =   31. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E
   CBC IV in: 59 00 00 00  06 05 04 03  A0 A1 A2 A3  A4 A5 00 13
   CBC IV out:F5 51 CF 6C  7C F7 D4 0B  2B 76 F1 6B  57 F0 19 FE
   After xor: F5 5D CF 6D  7E F4 D0 0E  2D 71 F9 62  5D FB 19 FE   [hdr]
   After CAM: 02 2B 21 1B  EB 97 02 3B  F8 10 7D CC  62 14 E5 7C
   After xor: 0E 26 2F 14  FB 86 10 28  EC 05 6B DB  7A 0D FF 67   [msg]
   After CAM: 48 14 A4 2D  31 25 1C 37  19 C5 6F DD  5A 37 81 42
   After xor: 54 09 BA 2D  31 25 1C 37  19 C5 6F DD  5A 37 81 42   [msg]
   After CAM: CF 85 25 D2  80 D5 F0 09  53 2C 9D 43  4E F3 04 47
   MIC tag  : CF 85 25 D2  80 D5 F0 09
   CTR Start: 01 00 00 00  06 05 04 03  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: C6 E2 10 8D  62 00 A2 9C  6F CC 19 1F  DF 6B 92 DB
   CTR[0002]: 6C B9 BE EE  1E A2 E9 B3  2D D6 C2 9A  E8 26 D5 C2
   CTR[MIC ]: 44 BF B6 E8  E3 31 67 A9
   Total packet length =   39. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  CA EF 1E 82
              72 11 B0 8F  7B D9 0F 08  C7 72 88 C0  70 A4 A0 8B
              3A 93 3A 63  E4 97 A0
```

```
   =============== Packet Vector #5 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 07  06 05 04 A0  A1 A2 A3 A4  A5
   Total packet length =   32. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
   CBC IV in: 59 00 00 00  07 06 05 04  A0 A1 A2 A3  A4 A5 00 14
   CBC IV out:73 72 9D 76  7A BD B9 82  60 3A 12 7B  EF 26 FB 80
   After xor: 73 7E 9D 77  78 BE BD 87  66 3D 1A 72  E5 2D FB 80   [hdr]
   After CAM: E1 B7 A6 72  E2 5C 87 75  91 21 22 A4  07 13 CD 5B
   After xor: ED BA A8 7D  F2 4D 95 66  85 34 34 B3  1F 0A D7 40   [msg]
   After CAM: 13 2F 58 D9  5D 0F 95 B8  90 BF 6F 1D  31 84 54 C7
   After xor: 0F 32 46 C6  5D 0F 95 B8  90 BF 6F 1D  31 84 54 C7   [msg]
   After CAM: 47 8F 1E B0  71 24 8B 13  AF C8 C8 44  E6 0F 88 B6
   MIC tag  : 47 8F 1E B0  71 24 8B 13
   CTR Start: 01 00 00 00  07 06 05 04  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 26 DE B4 D6  5F D4 3C 81  AA 56 98 95  64 09 39 A2
   CTR[0002]: 76 97 69 3A  21 13 0C 39  2E 4E EB BF  48 7B 24 BE
   CTR[MIC ]: C8 2E 65 17  82 15 50 1A
   Total packet length =   40. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  2A D3 BA D9
              4F C5 2E 92  BE 43 8E 82  7C 10 23 B9  6A 8A 77 25
              8F A1 7B A7  F3 31 DB 09

   =============== Packet Vector #6 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 08  07 06 05 A0  A1 A2 A3 A4  A5
   Total packet length =   33. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
              20
   CBC IV in: 59 00 00 00  08 07 06 05  A0 A1 A2 A3  A4 A5 00 15
   CBC IV out:EB 59 05 CC  3F 52 61 10  26 24 75 93  DD B9 A0 F4
   After xor: EB 55 05 CD  3D 51 65 15  20 23 7D 9A  D7 B2 A0 F4   [hdr]
   After CAM: 18 A9 AE A4  3D D2 A9 11  6C 0A E5 4F  40 D1 4D 9F
   After xor: 14 A4 A0 AB  2D C3 BB 02  78 1F F3 58  58 C8 57 84   [msg]
   After CAM: FA C4 13 18  98 54 1B 54  93 9C 64 B8  CB FD 5B 18
   After xor: E6 D9 0D 07  B8 54 1B 54  93 9C 64 B8  CB FD 5B 18   [msg]
   After CAM: 49 E6 E8 ED  32 FB CA 2F  2E 55 CD AF  D0 F2 B3 05
   MIC tag  : 49 E6 E8 ED  32 FB CA 2F
   CTR Start: 01 00 00 00  08 07 06 05  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: F2 A8 46 04  B5 2E BA C0  D7 51 34 BD  D6 54 FC 64
   CTR[0002]: E6 26 A9 24  8B E6 86 CB  92 D6 FB FC  2E F2 91 98
   CTR[MIC ]: E2 D0 49 03  7D 1B 34 07
   Total packet length =   41. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  FE A5 48 0B
              A5 3F A8 D3  C3 44 22 AA  CE 4D E6 7F  FA 3B B7 3B
              AB AB 36 A1  EE 4F E0 FE  28
```

```
   =============== Packet Vector #7 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 09  08 07 06 A0  A1 A2 A3 A4  A5
   Total packet length =   31. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E
   CBC IV in: 61 00 00 00  09 08 07 06  A0 A1 A2 A3  A4 A5 00 17
   CBC IV out:AC F1 5D 79  99 1A 15 BF  5C DC F6 C4  45 AE 1F CB
   After xor: AC F9 5D 78  9B 19 11 BA  5A DB F6 C4  45 AE 1F CB   [hdr]
   After CAM: E9 C0 AC FD  C7 E8 E7 1D  FA E8 8B 66  95 9E 01 45
   After xor: E1 C9 A6 F6  CB E5 E9 12  EA F9 99 75  81 8B 17 52   [msg]
   After CAM: 9C FF ED 72  09 A6 7D 2A  48 B7 29 BF  D8 BE 39 59
   After xor: 84 E6 F7 69  15 BB 63 2A  48 B7 29 BF  D8 BE 39 59   [msg]
   After CAM: 4F 41 FA DE  B2 58 F3 32  54 0A 55 7A  80 4A A3 F5
   MIC tag  : 4F 41 FA DE  B2 58 F3 32  54 0A
   CTR Start: 01 00 00 00  09 08 07 06  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 5C 5A 2A 2D  E9 41 1F 95  9D 27 CB FF  7A 0B CF 63
   CTR[0002]: 0E D1 6A 97  57 41 32 4F  33 1B 4A 42  B1 4A 54 63
   CTR[MIC ]: E3 EE 59 62  7D 22 BD 8D  C1 79
   Total packet length =   41. [Encrypted]
              00 01 02 03  04 05 06 07  54 53 20 26  E5 4C 11 9A
              8D 36 D9 EC  6E 1E D9 74  16 C8 70 8C  4B 5C 2C AC
              AF A3 BC CF  7A 4E BF 95  73

   =============== Packet Vector #8 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 0A  09 08 07 A0  A1 A2 A3 A4  A5
   Total packet length =   32. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
   CBC IV in: 61 00 00 00  0A 09 08 07  A0 A1 A2 A3  A4 A5 00 18
   CBC IV out:AD CA 1C 1D  45 E7 E2 62  58 D5 DA 46  D8 2F 69 3A
   After xor: AD C2 1C 1C  47 E4 E6 67  5E D2 DA 46  D8 2F 69 3A   [hdr]
   After CAM: FA DE 0E B4  3E CA C1 E9  69 BB 8C A4  7C 0D 80 8F
   After xor: F2 D7 04 BF  32 C7 CF E6  79 AA 9E B7  68 18 96 98   [msg]
   After CAM: D2 87 35 C2  D0 E4 AE 4E  BC C2 99 FF  B3 77 F8 A1
   After xor: CA 9E 2F D9  CC F9 B0 51  BC C2 99 FF  B3 77 F8 A1   [msg]
   After CAM: BD F6 FB 55  9E 90 C0 E7  DF 4B 0C 37  DC 42 32 A2
   MIC tag  : BD F6 FB 55  9E 90 C0 E7  DF 4B
   CTR Start: 01 00 00 00  0A 09 08 07  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 82 D8 91 0B  16 8A DF 47  E4 C8 39 FC  20 47 4A DB
   CTR[0002]: FB BF 26 7E  0E BB EB 6A  07 4E 29 CF  3D 12 E6 DB
   CTR[MIC ]: CE 7E 1F C4  A0 61 87 E6  2B 0A
   Total packet length =   42. [Encrypted]
              00 01 02 03  04 05 06 07  8A D1 9B 00  1A 87 D1 48
              F4 D9 2B EF  34 52 5C CC  E3 A6 3C 65  12 A6 F5 75
              73 88 E4 91  3E F1 47 01  F4 41
```

```
   =============== Packet Vector #9 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 0B  0A 09 08 A0  A1 A2 A3 A4  A5
   Total packet length =   33. [Input (8 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
              20
   CBC IV in: 61 00 00 00  0B 0A 09 08  A0 A1 A2 A3  A4 A5 00 19
   CBC IV out:D0 A9 A5 94  00 63 86 40  11 0D DB 40  CA F8 4A 9C
   After xor: D0 A1 A5 95  02 60 82 45  17 0A DB 40  CA F8 4A 9C   [hdr]
   After CAM: 7B CA 4E 2D  79 82 0D 1E  15 22 DD E8  37 B9 B1 F0
   After xor: 73 C3 44 26  75 8F 03 11  05 33 CF FB  23 AC A7 E7   [msg]
   After CAM: 6B 75 9F 83  C0 8F 56 64  F2 FA D5 7F  67 01 B8 21
   After xor: 73 6C 85 98  DC 92 48 7B  D2 FA D5 7F  67 01 B8 21   [msg]
   After CAM: 7D B7 BE FF  72 F3 26 74  9E 20 07 28  1E 5B 1A 8A
   MIC tag  : 7D B7 BE FF  72 F3 26 74  9E 20
   CTR Start: 01 00 00 00  0B 0A 09 08  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 55 B9 87 69  4C 73 60 3E  C6 1E 8E B1  D2 11 62 36
   CTR[0002]: 82 D9 A4 4B  DC C9 BB 68  A7 FE 15 A5  19 51 57 87
   CTR[MIC ]: E9 61 5C CF  BF D6 EF 8A  21 A7
   Total packet length =   43. [Encrypted]
              00 01 02 03  04 05 06 07  5D B0 8D 62  40 7E 6E 31
              D6 0F 9C A2  C6 04 74 21  9A C0 BE 50  C0 D4 A5 77
              87 94 D6 E2  30 CD 25 C9  FE BF 87

   =============== Packet Vector #10 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 0C  0B 0A 09 A0  A1 A2 A3 A4  A5
   Total packet length =   31. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E
   CBC IV in: 61 00 00 00  0C 0B 0A 09  A0 A1 A2 A3  A4 A5 00 13
   CBC IV out:B1 85 73 A3  1C 6F EC 01  90 E3 CE 94  27 11 04 B9
   After xor: B1 89 73 A2  1E 6C E8 04  96 E4 C6 9D  2D 1A 04 B9   [hdr]
   After CAM: A6 AD EA 9C  FA 3F 76 78  4C 17 8A F3  DC 69 F0 82
   After xor: AA A0 E4 93  EA 2E 64 6B  58 02 9C E4  C4 70 EA 99   [msg]
   After CAM: 35 50 B7 27  78 F8 C6 BF  02 4B 65 60  05 C0 E1 ED
   After xor: 29 4D A9 27  78 F8 C6 BF  02 4B 65 60  05 C0 E1 ED   [msg]
   After CAM: 3D B5 A6 E6  85 AF 1C 58  80 B0 32 2E  01 74 91 FC
   MIC tag  : 3D B5 A6 E6  85 AF 1C 58  80 B0
   CTR Start: 01 00 00 00  0C 0B 0A 09  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: D7 1C 82 C1  D1 A9 64 0F  93 69 CE 81  22 7E CC E8
   CTR[0002]: A7 A1 42 44  32 4E 69 FE  4C D0 36 65  A5 31 0B AB
   CTR[MIC ]: ED 27 3F 0D  94 5C 0E AA  B2 87
   Total packet length =   41. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  DB 11 8C CE
              C1 B8 76 1C  87 7C D8 96  3A 67 D6 F3  BB BC 5C D0
              92 99 EB 11  F3 12 F2 32  37
```

```
   =============== Packet Vector #11 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 0D  0C 0B 0A A0  A1 A2 A3 A4  A5
   Total packet length =   32. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
   CBC IV in: 61 00 00 00  0D 0C 0B 0A  A0 A1 A2 A3  A4 A5 00 14
   CBC IV out:45 DF B5 07  6F BB 10 EA  F1 15 15 AD  21 4F B0 0E
   After xor: 45 D3 B5 06  6D B8 14 EF  F7 12 1D A4  2B 44 B0 0E   [hdr]
   After CAM: 17 52 F9 6D  DD BC 5B 1C  1E EB 80 FC  F6 10 AC 03
   After xor: 1B 5F F7 62  CD AD 49 0F  0A FE 96 EB  EE 09 B6 18   [msg]
   After CAM: BE F0 A0 B9  EC 94 B6 B3  E8 EC 1B 82  14 14 09 87
   After xor: A2 ED BE A6  EC 94 B6 B3  E8 EC 1B 82  14 14 09 87   [msg]
   After CAM: 70 16 E4 F9  C4 2C 30 10  84 BF EC 69  34 89 91 FD
   MIC tag  : 70 16 E4 F9  C4 2C 30 10  84 BF
   CTR Start: 01 00 00 00  0D 0C 0B 0A  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 70 C5 33 82  D4 80 11 41  4F 5D 2B D2  D2 67 B3 B0
   CTR[0002]: 9D 36 6E 49  39 C5 16 76  5C 1C 25 12  81 79 94 70
   CTR[MIC ]: 77 8B 4B 03  1E 3A FC DF  A8 F1
   Total packet length =   42. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  7C C8 3D 8D
              C4 91 03 52  5B 48 3D C5  CA 7E A9 AB  81 2B 70 56
              07 9D AF FA  DA 16 CC CF  2C 4E

   =============== Packet Vector #12 ==================
   CAM Key:   C0 C1 C2 C3  C4 C5 C6 C7  C8 C9 CA CB  CC CD CE CF
   Nonce =    00 00 00 0E  0D 0C 0B A0  A1 A2 A3 A4  A5
   Total packet length =   33. [Input (12 cleartext header octets)]
              00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
              10 11 12 13  14 15 16 17  18 19 1A 1B  1C 1D 1E 1F
              20
   CBC IV in: 61 00 00 00  0E 0D 0C 0B  A0 A1 A2 A3  A4 A5 00 15
   CBC IV out:81 E4 EB 1E  50 A9 70 CE  18 CA 1A 4B  68 39 80 2E
   After xor: 81 E8 EB 1F  52 AA 74 CB  1E CD 12 42  62 32 80 2E   [hdr]
   After CAM: 04 AB D9 62  34 B9 8F 32  8C 0F 08 3F  3D 87 9D 57
   After xor: 08 A6 D7 6D  24 A8 9D 21  98 1A 1E 28  25 9E 87 4C   [msg]
   After CAM: BD A2 EA CB  3A DA 6A E7  9F BB C2 2C  E6 4C 98 89
   After xor: A1 BF F4 D4  1A DA 6A E7  9F BB C2 2C  E6 4C 98 89   [msg]
   After CAM: B6 FC E1 46  D3 EA DC 91  E0 AB 10 AD  D8 55 E7 03
   MIC tag  : B6 FC E1 46  D3 EA DC 91  E0 AB
   CTR Start: 01 00 00 00  0E 0D 0C 0B  A0 A1 A2 A3  A4 A5 00 01
   CTR[0001]: 20 DE 55 87  30 C3 2C 69  B7 44 A6 FE  37 DE 89 7C
   CTR[0002]: 3F 96 32 D8  68 6D C2 B5  22 97 42 27  EB F9 26 5E
   CTR[MIC ]: 7D 45 AD 6F  94 93 E1 F5  4F DE
   Total packet length =   43. [Encrypted]
              00 01 02 03  04 05 06 07  08 09 0A 0B  2C D3 5B 88
              20 D2 3E 7A  A3 51 B0 E9  2F C7 93 67  23 8B 2C C7
              48 CB B9 4C  29 47 79 3D  64 AF 75
```

```
   =============== Packet Vector #13 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 A9 70 11  0E 19 27 B1  60 B6 A3 1C  1C
   Total packet length =   31. [Input (8 cleartext header octets)]
              6B 7F 46 45  07 FA E4 96  C6 B5 F3 E6  CA 23 11 AE
              F7 47 2B 20  3E 73 5E A5  61 AD B1 7D  56 C5 A3
   CBC IV in: 59 00 A9 70  11 0E 19 27  B1 60 B6 A3  1C 1C 00 17
   CBC IV out:D7 24 B0 0F  B1 87 04 C6  C1 4E 90 37  AA F2 F1 F9
   After xor: D7 2C DB 70  F7 C2 03 3C  25 D8 90 37  AA F2 F1 F9   [hdr]
   After CAM: 9B 13 6D E3  D9 9F C3 6D  7D 0D B7 D8  A1 BF E9 BD
   After xor: 5D A6 9E 05  13 BC D2 C3  8A 4A 9C F8  9F CC B7 18   [msg]
   After CAM: F8 BF 25 7D  23 F8 D9 B5  82 E6 C9 3E  C8 9B 85 73
   After xor: 99 12 94 00  75 3D 7A B5  82 E6 C9 3E  C8 9B 85 73   [msg]
   After CAM: D9 D6 62 21  6D B2 CA FD  1F C6 FE 9D  2C AF 5B 69
   MIC tag  : D9 D6 62 21  6D B2 CA FD
   CTR Start: 01 00 A9 70  11 0E 19 27  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 62 80 24 C1  FE AE CC 8C  67 38 55 98  CB 8E E5 E8
   CTR[0002]: F2 30 17 2F  1B 71 55 9F  8B CE 79 E5  13 01 FC 6A
   CTR[MIC ]: 9C 8E A2 0C  48 03 ED 13
   Total packet length =   39. [Encrypted]
              6B 7F 46 45  07 FA E4 96  A4 35 D7 27  34 8D DD 22
              90 7F 7E B8  F5 FD BB 4D  93 9D A6 52  4D B4 F6 45
              58 C0 2D 25  B1 27 EE

   =============== Packet Vector #14 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 83 CD 8C  E0 CB 42 B1  60 B6 A3 1C  1C
   Total packet length =   32. [Input (8 cleartext header octets)]
              98 66 05 B4  3D F1 5D E7  01 F6 CE 67  64 C5 74 48
              3B B0 2E 6B  BF 1E 0A BD  26 A2 25 72  B4 D8 0E E7
   CBC IV in: 59 00 83 CD  8C E0 CB 42  B1 60 B6 A3  1C 1C 00 18
   CBC IV out:A0 8A 29 78  36 23 1D 84  96 76 93 FF  0A 4C 92 7A
   After xor: A0 82 B1 1E  33 97 20 75  CB 91 93 FF  0A 4C 92 7A   [hdr]
   After CAM: 8C F5 F4 23  BF 09 1C 74  CD 47 00 C1  32 5D 5C 92
   After xor: 8D 03 3A 44  DB CC 68 3C  F6 F7 2E AA  8D 43 56 2F   [msg]
   After CAM: 69 DA 48 24  41 1E AC 8E  A9 0A CD 8B  DD 00 2B 9A
   After xor: 4F 78 6D 56  F5 C6 A2 69  A9 0A CD 8B  DD 00 2B 9A   [msg]
   After CAM: C2 03 3B 08  6D B3 CB 3B  2C C8 5D E7  76 A1 C0 44
   MIC tag  : C2 03 3B 08  6D B3 CB 3B
   CTR Start: 01 00 83 CD  8C E0 CB 42  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 8B 16 9C 37  EB 7B BE DB  15 84 41 6E  5F C2 07 46
   CTR[0002]: E9 31 BB DD  4E E6 56 9B  68 95 13 5F  AB A4 DF EF
   CTR[MIC ]: 44 7E 55 14  25 C3 F3 3D
   Total packet length =   40. [Encrypted]
              98 66 05 B4  3D F1 5D E7  8A E0 52 50  8F BE CA 93
              2E 34 6F 05  E0 DC 0D FB  CF 93 9E AF  FA 3E 58 7C
              86 7D 6E 1C  48 70 38 06
```

```
   =============== Packet Vector #15 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 5F 54 95  0B 18 F2 B1  60 B6 A3 1C  1C
   Total packet length =   33. [Input (8 cleartext header octets)]
              48 F2 E7 E1  A7 67 1A 51  CD F1 D8 40  6F C2 E9 01
              49 53 89 70  05 FB FB 8B  A5 72 76 F9  24 04 60 8E
              08
   CBC IV in: 59 00 5F 54  95 0B 18 F2  B1 60 B6 A3  1C 1C 00 19
   CBC IV out:76 74 53 37  95 23 3C F0  EB 77 CE 93  73 06 99 A8
   After xor: 76 7C 1B C5  72 C2 9B 97  F1 26 CE 93  73 06 99 A8   [hdr]
   After CAM: EF 79 8B 70  34 E4 D5 6B  57 3A F9 44  F0 AF D6 9A
   After xor: 22 88 53 30  5B 26 3C 6A  1E 69 70 34  F5 54 2D 11   [msg]
   After CAM: 63 BF 4E 10  01 79 38 0B  E4 EC C1 39  B2 B4 3B 8C
   After xor: C6 CD 38 E9  25 7D 58 85  EC EC C1 39  B2 B4 3B 8C   [msg]
   After CAM: 39 E1 0E FA  BD 2F 43 00  50 9E E7 EB  A4 FF 6B 8F
   MIC tag  : 39 E1 0E FA  BD 2F 43 00
   CTR Start: 01 00 5F 54  95 0B 18 F2  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: C5 47 A6 A2  73 49 1B 6F  0E 6D C9 F5  9C 12 3B 08
   CTR[0002]: C8 18 86 42  3C DB 35 C8  64 4D 8C 4C  58 01 47 27
   CTR[MIC ]: 91 E9 76 5D  2D 68 2E E5
   Total packet length =   41. [Encrypted]
              48 F2 E7 E1  A7 67 1A 51  08 B6 7E E2  1C 8B F2 6E
              47 3E 40 85  99 E9 C0 83  6D 6A F0 BB  18 DF 55 46
              6C A8 08 78  A7 90 47 6D  E5

   =============== Packet Vector #16 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 EC 60 08  63 31 9A B1  60 B6 A3 1C  1C
   Total packet length =   31. [Input (12 cleartext header octets)]
              DE 97 DF 3B  8C BD 6D 8E  50 30 DA 4C  B0 05 DC FA
              0B 59 18 14  26 A9 61 68  5A 99 3D 8C  43 18 5B
   CBC IV in: 59 00 EC 60  08 63 31 9A  B1 60 B6 A3  1C 1C 00 13
   CBC IV out:78 EE 05 5A  88 48 E3 5B  8A 45 46 8F  35 4F 0C A2
   After xor: 78 E2 DB CD  57 73 6F E6  E7 CB 16 BF  EF 03 0C A2   [hdr]
   After CAM: A9 C6 7F 15  00 1A C6 92  81 67 BD EC  DF D2 35 C9
   After xor: 19 C3 A3 EF  0B 43 DE 86  A7 CE DC 84  85 4B 08 45   [msg]
   After CAM: 7C A8 9C 90  46 42 4B E2  4D 96 DF CF  BA 12 FD 18
   After xor: 3F B0 C7 90  46 42 4B E2  4D 96 DF CF  BA 12 FD 18   [msg]
   After CAM: 89 C7 B4 E8  A4 24 8C 6C  52 ED 34 50  E3 53 AD F5
   MIC tag  : 89 C7 B4 E8  A4 24 8C 6C
   CTR Start: 01 00 EC 60  08 63 31 9A  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: D3 B2 57 B3  6C E8 86 CF  91 9A AC 79  4E 6F 73 3E
   CTR[0002]: 65 10 C8 72  39 AF 0F 52  9F D0 A4 DF  54 BF D6 EB
   CTR[MIC ]: E1 04 E0 6A  29 B1 80 A9
   Total packet length =   39. [Encrypted]
              DE 97 DF 3B  8C BD 6D 8E  50 30 DA 4C  63 B7 8B 49
              67 B1 9E DB  B7 33 CD 11  14 F6 4E B2  26 08 93 68
              C3 54 82 8D  95 0C C5
```

```
   =============== Packet Vector #17 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 60 CF F1  A3 1E A1 B1  60 B6 A3 1C  1C
   Total packet length =   32. [Input (12 cleartext header octets)]
              A5 EE 93 E4  57 DF 05 46  6E 78 2D CF  2E 20 21 12
              98 10 5F 12  9D 5E D9 5B  93 F7 2D 30  B2 FA CC D7
   CBC IV in: 59 00 60 CF  F1 A3 1E A1  B1 60 B6 A3  1C 1C 00 14
   CBC IV out:C3 34 69 7D  11 38 73 06  BD 34 E2 10  1F 66 17 E8
   After xor: C3 38 CC 93  82 DC 24 D9  B8 72 8C 68  32 A9 17 E8   [hdr]
   After CAM: 43 6F 37 74  AB 94 3B 41  EA AD 00 CA  C3 99 13 7B
   After xor: 6D 4F 16 66  33 84 64 53  77 F3 D9 91  50 6E 3E 4B   [msg]
   After CAM: 2D 28 FB 62  DA 06 97 A7  4C D4 31 B8  B5 AE AE EE
   After xor: 9F D2 37 B5  DA 06 97 A7  4C D4 31 B8  B5 AE AE EE   [msg]
   After CAM: F3 DE 10 CD  91 4D B1 B6  CC 37 F0 A2  4A 5A B7 A1
   MIC tag  : F3 DE 10 CD  91 4D B1 B6
   CTR Start: 01 00 60 CF  F1 A3 1E A1  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 25 E6 9A F0  30 A9 56 E6  FF C0 3F 87  87 7A 89 74
   CTR[0002]: A2 1B 46 23  76 A2 1E DD  F2 AC 4B EC  42 95 3D D3
   CTR[MIC ]: C2 99 28 FF  E7 BB DB 29
   Total packet length =   40. [Encrypted]
              A5 EE 93 E4  57 DF 05 46  6E 78 2D CF  0B C6 BB E2
              A8 B9 09 F4  62 9E E6 DC  14 8D A4 44  10 E1 8A F4
              31 47 38 32  76 F6 6A 9F

   =============== Packet Vector #18 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 0F 85 CD  99 5C 97 B1  60 B6 A3 1C  1C
   Total packet length =   33. [Input (12 cleartext header octets)]
              24 AA 1B F9  A5 CD 87 61  82 A2 50 74  26 45 94 1E
              75 63 2D 34  91 AF 0F C0  C9 87 6C 3B  E4 AA 74 68
              C9
   CBC IV in: 59 00 0F 85  CD 99 5C 97  B1 60 B6 A3  1C 1C 00 15
   CBC IV out:72 0A 46 75  0F 40 59 53  F2 3B D2 1F  6A 11 60 F6
   After xor: 72 06 62 DF  14 B9 FC 9E  75 5A 50 BD  3A 65 60 F6   [hdr]
   After CAM: 67 73 A0 FD  D5 7E D3 5E  E8 24 06 D0  A1 8B 0E 18
   After xor: 41 36 34 E3  A0 1D FE 6A  79 8B 09 10  68 0C 62 23   [msg]
   After CAM: BB 1E D8 9F  60 29 D0 99  09 14 06 A5  E3 8B 72 7B
   After xor: 5F B4 AC F7  A9 29 D0 99  09 14 06 A5  E3 8B 72 7B   [msg]
   After CAM: 3E 4F 40 73  D1 31 E9 B8  02 C8 99 BC  FD AC 19 4B
   MIC tag  : 3E 4F 40 73  D1 31 E9 B8
   CTR Start: 01 00 0F 85  CD 99 5C 97  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 04 6F 42 2C  8F 52 FB 9B  06 A3 3B 9F  B7 F0 A6 00
   CTR[0002]: 34 76 51 DB  89 10 FB E6  73 E8 56 6E  DB 66 47 5D
   CTR[MIC ]: 9F EC 93 6C  5C 7A AD 0F
   Total packet length =   41. [Encrypted]
              24 AA 1B F9  A5 CD 87 61  82 A2 50 74  22 2A D6 32
              FA 31 D6 AF  97 0C 34 5F  7E 77 CA 3B  D0 DC 25 B3
              40 A1 A3 D3  1F 8D 4B 44  B7
```

```
   =============== Packet Vector #19 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 C2 9B 2C  AA C4 CD B1  60 B6 A3 1C  1C
   Total packet length =   31. [Input (8 cleartext header octets)]
              69 19 46 B9  CA 07 BE 87  07 01 35 A6  43 7C 9D B1
              20 CD 61 D8  F6 C3 9C 3E  A1 25 FD 95  A0 D2 3D
   CBC IV in: 61 00 C2 9B  2C AA C4 CD  B1 60 B6 A3  1C 1C 00 17
   CBC IV out:74 AD F8 04  05 2A 48 E7  46 97 38 D5  BA A1 27 79
   After xor: 74 A5 91 1D  43 93 82 E0  F8 10 38 D5  BA A1 27 79   [hdr]
   After CAM: BD C3 B1 41  1C 64 C8 B3  A9 DC 6A 94  78 97 88 E2
   After xor: BA C2 84 E7  5F 18 55 02  89 11 0B 4C  8E 54 14 DC   [msg]
   After CAM: 7D 6C 8A BF  AD 68 48 D8  C5 FB CD 1E  AF F2 44 99
   After xor: DC 49 77 2A  0D BA 75 D8  C5 FB CD 1E  AF F2 44 99   [msg]
   After CAM: 19 99 AB 92  5E 30 46 96  3D EF FB 1B  4C 87 F7 76
   MIC tag  : 19 99 AB 92  5E 30 46 96  3D EF
   CTR Start: 01 00 C2 9B  2C AA C4 CD  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 02 B9 D4 1F  87 E0 60 E7  EF DE 6B 7E  D3 DE 5E D2
   CTR[0002]: 61 49 31 C5  2F 34 AA 47  A3 E4 D3 2C  0B 36 41 C6
   CTR[MIC ]: B9 9F C6 C5  96 7B AA 8E  1A 87
   Total packet length =   41. [Encrypted]
              69 19 46 B9  CA 07 BE 87  05 B8 E1 B9  C4 9C FD 56
              CF 13 0A A6  25 1D C2 EC  C0 6C CC 50  8F E6 97 A0
              06 6D 57 C8  4B EC 18 27  68

   =============== Packet Vector #20 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 2C 6B 75  95 EE 62 B1  60 B6 A3 1C  1C
   Total packet length =   32. [Input (8 cleartext header octets)]
              D0 C5 4E CB  84 62 7D C4  C8 C0 88 0E  6C 63 6E 20
              09 3D D6 59  42 17 D2 E1  88 77 DB 26  4E 71 A5 CC
   CBC IV in: 61 00 2C 6B  75 95 EE 62  B1 60 B6 A3  1C 1C 00 18
   CBC IV out:35 A9 48 70  F9 B0 C7 85  FB 32 1A D1  3C 8C A4 9A
   After xor: 35 A1 98 B5  B7 7B 43 E7  86 F6 1A D1  3C 8C A4 9A   [hdr]
   After CAM: 0A 3C E3 0F  AC 09 DC 5C  00 10 5C 69  AC 19 F7 19
   After xor: C2 FC 6B 01  C0 6A B2 7C  09 2D 8A 30  EE 0E 25 F8   [msg]
   After CAM: 61 CD 80 D0  72 E6 84 E1  BF E1 4A 00  27 2A 4D 96
   After xor: E9 BA 5B F6  3C 97 21 2D  BF E1 4A 00  27 2A 4D 96   [msg]
   After CAM: E5 F9 F2 AB  47 FD 7B 8D  6F 72 F4 72  74 D7 69 BB
   MIC tag  : E5 F9 F2 AB  47 FD 7B 8D  6F 72
   CTR Start: 01 00 2C 6B  75 95 EE 62  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 9C 0E 31 66  B2 81 58 31  5E 63 16 5A  9D BD CE 35
   CTR[0002]: 00 3E 66 D3  E0 5F 7E A7  EF C8 9A 5F  DD 39 E3 54
   CTR[MIC ]: 9A 5E 87 1A  17 10 38 0E  AA DB
   Total packet length =   42. [Encrypted]
              D0 C5 4E CB  84 62 7D C4  54 CE B9 68  DE E2 36 11
              57 5E C0 03  DF AA 1C D4  88 49 BD F5  AE 2E DB 6B
              7F A7 75 B1  50 ED 43 83  C5 A9
```

```
   =============== Packet Vector #21 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 C5 3C D4  C2 AA 24 B1  60 B6 A3 1C  1C
   Total packet length =   33. [Input (8 cleartext header octets)]
              E2 85 E0 E4  80 8C DA 3D  F7 5D AA 07  10 C4 E6 42
              97 79 4D C2  B7 D2 A2 07  57 B1 AA 4E  44 80 02 FF
              AB
   CBC IV in: 61 00 C5 3C  D4 C2 AA 24  B1 60 B6 A3  1C 1C 00 19
   CBC IV out:2A 3C 23 B2  43 F5 1C 35  F7 79 5A CB  3B 20 21 2F
   After xor: 2A 34 C1 37  A3 11 9C B9  2D 44 5A CB  3B 20 21 2F   [hdr]
   After CAM: A1 7E AD 4C  EE AB 51 21  1D 2A 32 F2  D4 45 A6 D6
   After xor: 56 23 07 4B  FE 6F B7 63  8A 53 7F 30  63 97 04 D1   [msg]
   After CAM: A9 A1 32 55  8F C6 9B 98  A9 CC 23 96  FE CA 84 EB
   After xor: FE 10 98 1B  CB 46 99 67  02 CC 23 96  FE CA 84 EB   [msg]
   After CAM: 6A 5E 04 42  D1 A5 7E 17  9A 6C 8B 56  F7 19 80 C5
   MIC tag  : 6A 5E 04 42  D1 A5 7E 17  9A 6C
   CTR Start: 01 00 C5 3C  D4 C2 AA 24  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 46 1D EF 41  AF A2 94 52  5D 51 AE CB  04 49 74 CD
   CTR[0002]: 29 2E 62 66  1B 66 9A 2B  97 72 6B 77  32 A8 DC 35
   CTR[MIC ]: B8 54 06 A2  6C 6F 93 37  8A BF
   Total packet length =   43. [Encrypted]
              E2 85 E0 E4  80 8C DA 3D  B1 40 45 46  BF 66 72 10
              CA 28 E3 09  B3 9B D6 CA  7E 9F C8 28  5F E6 98 D4
              3C D2 0A 02  E0 BD CA ED  20 10 D3

   =============== Packet Vector #22 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 BE E9 26  7F BA DC B1  60 B6 A3 1C  1C
   Total packet length =   31. [Input (12 cleartext header octets)]
              6C AE F9 94  11 41 57 0D  7C 81 34 05  C2 38 82 2F
              AC 5F 98 FF  92 94 05 B0  AD 12 7A 4E  41 85 4E
   CBC IV in: 61 00 BE E9  26 7F BA DC  B1 60 B6 A3  1C 1C 00 13
   CBC IV out:20 60 6A D1  E1 A0 84 52  2F A3 8B F4  88 1D D6 8B
   After xor: 20 6C 06 7F  18 34 95 13  78 AE F7 75  BC 18 D6 8B   [hdr]
   After CAM: 71 FD FF E7  D9 C8 95 75  D3 EC 0B 7E  7B 8B BE E7
   After xor: B3 C5 7D C8  75 97 0D 8A  41 78 0E CE  D6 99 C4 A9   [msg]
   After CAM: CA AD 93 9C  59 BA 40 AA  1A 0B 88 1B  EE 3D 3C 65
   After xor: 8B 28 DD 9C  59 BA 40 AA  1A 0B 88 1B  EE 3D 3C 65   [msg]
   After CAM: DC 48 8F AA  9C 75 E7 03  17 56 C2 C7  48 48 8D 1B
   MIC tag  : DC 48 8F AA  9C 75 E7 03  17 56
   CTR Start: 01 00 BE E9  26 7F BA DC  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 56 F0 17 B3  BD 09 02 D6  EA A5 A2 91  AD 4A 2D E5
   CTR[0002]: 20 3D 34 21  EF 5B F8 FC  7B 21 5C 76  7B A5 21 A6
   CTR[MIC ]: F1 A2 86 9C  2A 9E B8 61  48 0B
   Total packet length =   41. [Encrypted]
              6C AE F9 94  11 41 57 0D  7C 81 34 05  94 C8 95 9C
              11 56 9A 29  78 31 A7 21  00 58 57 AB  61 B8 7A 2D
              EA 09 36 B6  EB 5F 62 5F  5D
```

```
   =============== Packet Vector #23 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 DF A8 B1  24 50 07 B1  60 B6 A3 1C  1C
   Total packet length =   32. [Input (12 cleartext header octets)]
              36 A5 2C F1  6B 19 A2 03  7A B7 01 1E  4D BF 3E 77
              4A D2 45 E5  D5 89 1F 9D  1C 32 A0 AE  02 2C 85 D7
   CBC IV in: 61 00 DF A8  B1 24 50 07  B1 60 B6 A3  1C 1C 00 14
   CBC IV out:78 FD B6 AF  61 9E 1C 8D  82 41 17 A8  73 60 1B 70
   After xor: 78 F1 80 0A  4D 6F 77 94  20 42 6D 1F  72 7E 1B 70   [hdr]
   After CAM: 62 2E 28 65  92 43 DB 82  88 79 09 1E  A7 24 54 67
   After xor: 2F 91 16 12  D8 91 9E 67  5D F0 16 83  BB 16 F4 C9   [msg]
   After CAM: 95 0E 52 08  FF 16 70 8C  1E D9 BB 06  3E 1E 41 CF
   After xor: 97 22 D7 DF  FF 16 70 8C  1E D9 BB 06  3E 1E 41 CF   [msg]
   After CAM: BA CD 51 FC  77 F4 02 8D  47 D5 7D 54  7D 46 33 4B
   MIC tag  : BA CD 51 FC  77 F4 02 8D  47 D5
   CTR Start: 01 00 DF A8  B1 24 50 07  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: 15 D6 DD DD  98 96 39 91  35 75 1A 64  B8 D8 D4 F9
   CTR[0002]: 7D 61 6D 1D  EB 92 00 2B  6F FA AB 53  BC AF 69 89
   CTR[MIC ]: 33 E9 27 BE  E1 59 06 9C  DB 32
   Total packet length =   42. [Encrypted]
              36 A5 2C F1  6B 19 A2 03  7A B7 01 1E  58 69 E3 AA
              D2 44 7C 74  E0 FC 05 F9  A4 EA 74 57  7F 4D E8 CA
              89 24 76 42  96 AD 04 11  9C E7

   =============== Packet Vector #24 ==================
   CAM Key:   D7 5C 27 78  07 8C A9 3D  97 1F 96 FD  E7 20 F4 CD
   Nonce =    00 3B 8F D8  D3 A9 37 B1  60 B6 A3 1C  1C
   Total packet length =   33. [Input (12 cleartext header octets)]
              A4 D4 99 F7  84 19 72 8C  19 17 8B 0C  9D C9 ED AE
              2F F5 DF 86  36 E8 C6 DE  0E ED 55 F7  86 7E 33 33
              7D
   CBC IV in: 61 00 3B 8F  D8 D3 A9 37  B1 60 B6 A3  1C 1C 00 15
   CBC IV out:84 E6 CF DD  6A 37 68 5D  E6 71 AD 54  B3 BE FE B9
   After xor: 84 EA 6B 09  F3 C0 EC 44  94 FD B4 43  38 B2 FE B9   [hdr]
   After CAM: C5 0F A0 62  20 18 F1 21  0E BC 3D 2E  47 B7 B8 C3
   After xor: 58 C6 4D CC  0F ED 2E A7  38 54 FB F0  49 5A ED 34   [msg]
   After CAM: C4 6F 6D C3  17 3C 2A 7A  81 FC 2D DA  7F B7 C6 60
   After xor: 42 11 5E F0  6A 3C 2A 7A  81 FC 2D DA  7F B7 C6 60   [msg]
   After CAM: DF AB 2E 76  B0 67 50 B3  7C DD 9A AC  F3 79 17 71
   MIC tag  : DF AB 2E 76  B0 67 50 B3  7C DD
   CTR Start: 01 00 3B 8F  D8 D3 A9 37  B1 60 B6 A3  1C 1C 00 01
   CTR[0001]: D6 D0 6C F8  16 CE D0 F1  A0 E0 AC 71  BA B9 AD 34
   CTR[0002]: 76 4A FF 9A  1B F8 55 1F  68 54 39 0A  EE 37 24 28
   CTR[MIC ]: 4B F4 31 B8  17 86 4B 5D  16 F2
   Total packet length =   43. [Encrypted]
              A4 D4 99 F7  84 19 72 8C  19 17 8B 0C  4B 19 81 56
              39 3B 0F 77  96 08 6A AF  B4 54 F8 C3  F0 34 CC A9
              66 94 5F 1F  CE A7 E1 1B  EE 6A 2F
```

5. Security Considerations

Camellia-CTR and Camellia-CCM employ CTR mode for confidentiality. For the security of CTR mode, refer to the Security Considerations of [16].

6. Acknowledgments

Thanks to Rui Hodai for comments and suggestions. Special thanks to Alfred Hoenes for several very detailed reviews and suggestions.

7. References

7.1. Normative References

[1] Matsui, M., Nakajima, J., and S. Moriai, "A Description of the Camellia Encryption Algorithm", RFC 3713, April 2004.

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

[3] Dworkin, M., "Recommendation for Block Cipher Modes of Operation - Methods and Techniques", NIST Special Publication 800-38A, December 2001, <http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf>.

[4] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes Operation : The CCM Mode for Authentication and Confidentiality", May 2004, <http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C.pdf>.

7.2. Informative References

[5] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[6] Kato, A., Moriai, S., and M. Kanda, "The Camellia Cipher Algorithm and Its Use With IPsec", RFC 4312, December 2005.

[7] Moriai, S., Kato, A., and M. Kanda, "Addition of Camellia Cipher Suites to Transport Layer Security (TLS)", RFC 4132, July 2005.

[8] Moriai, S. and A. Kato, "Use of the Camellia Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3657, January 2004.

[9] Eastlake, D., "Additional XML Security Uniform Resource Identifiers (URIs)", RFC 4051, April 2005.

[10] International Organization for Standardization, "Information technology - Security techniques - Encryption algorithms - Part 3: Block ciphers", ISO/IEC 18033-3, July 2005.

[11] "The NESSIE project (New European Schemes for Signatures, Integrity and Encryption)", <http://www.cosic.esat.kuleuven.be/nessie/>.

[12] Information-technology Promotion Agency (IPA), "Cryptography Research and Evaluation Committees", <http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html>.

[13] "Camellia open source software", <http://info.isl.ntt.co.jp/crypt/eng/camellia/source.html>.

[14] "Camellia web site", <http://info.isl.ntt.co.jp/camellia/>.

[15] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[16] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

Authors' Addresses

Akihiro Kato
NTT Software Corporation

Phone: +81-45-212-7577
Fax: +81-45-212-9800
EMail: akato@po.ntts.co.jp

Masayuki Kanda
NTT

Phone: +81-422-59-3456
Fax: +81-422-59-4015
EMail: kanda.masayuki@lab.ntt.co.jp

Satoru Kanno
NTT Software Corporation

Phone: +81-45-212-7577
Fax: +81-45-212-9800
EMail: kanno-s@po.ntts.co.jp