Internet Engineering Task Force (IETF)
Request for Comments: 6367
Category: Informational
ISSN: 2070-1721

S. Kanno
NTT Software Corporation
M. Kanda
NTT

September 2011

# Addition of the Camellia Cipher Suites to Transport Layer Security (TLS)

## Abstract

This document specifies forty-two cipher suites for the Transport Security Layer (TLS) protocol to support the Camellia encryption algorithm as a block cipher.

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6367.

Table of Contents



## 1. Introduction

The Camellia cipher suites are already specified in RFC 5932 [15] with SHA-256-based Hashed Message Authentication Code (HMAC) using asymmetric key encryption. This document proposes the addition of new cipher suites to the Transport Layer Security (TLS) [8] protocol to support the Camellia [4] cipher algorithm as a block cipher algorithm. The proposed cipher suites include variants using the SHA-2 family of cryptographic hash functions [13] and Galois Counter Mode (GCM) [14]. Elliptic curve cipher suites and pre-shared key (PSK) [5] cipher suites are also included.

### 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 [3].

## 2. Proposed Cipher Suites

### 2.1. HMAC-Based Cipher Suites

The eight cipher suites use Camellia [4] in Cipher Block Chaining (CBC) [4] mode with a SHA-2 family HMAC using the elliptic curve cryptosystem:

```
 CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256 = {0xC0,0x72};
 CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384 = {0xC0,0x73};
 CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256  = {0xC0,0x74};
 CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384  = {0xC0,0x75};
 CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256   = {0xC0,0x76};
 CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384   = {0xC0,0x77};
 CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256    = {0xC0,0x78};
 CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384    = {0xC0,0x79};
```

### 2.2. GCM-Based Cipher Suites

The twenty cipher suites use the same asymmetric key algorithms as those in the previous section but use the authenticated encryption modes defined in TLS 1.2 [8] with Camellia in GCM [14].

```
CipherSuite TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256          = {0xC0,0x7A};
CipherSuite TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384          = {0xC0,0x7B};
CipherSuite TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x7C};
CipherSuite TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x7D};
CipherSuite TLS_DH_RSA_WITH_CAMELLIA_128_GCM_SHA256       = {0xC0,0x7E};
CipherSuite TLS_DH_RSA_WITH_CAMELLIA_256_GCM_SHA384       = {0xC0,0x7F};
CipherSuite TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x80};
CipherSuite TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x81};
CipherSuite TLS_DH_DSS_WITH_CAMELLIA_128_GCM_SHA256       = {0xC0,0x82};
CipherSuite TLS_DH_DSS_WITH_CAMELLIA_256_GCM_SHA384       = {0xC0,0x83};
CipherSuite TLS_DH_anon_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x84};
CipherSuite TLS_DH_anon_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x85};
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256  = {0xC0,0x86};
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384  = {0xC0,0x87};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256   = {0xC0,0x88};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384   = {0xC0,0x89};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256    = {0xC0,0x8A};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384    = {0xC0,0x8B};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256     = {0xC0,0x8C};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384     = {0xC0,0x8D};
```

### 2.3. PSK-Based Cipher Suites

The fourteen cipher suites describe PSK cipher suites. The first six cipher suites use Camellia with GCM, and the next eight cipher suites use Camellia with SHA-2 family HMAC using asymmetric key encryption or the elliptic curve cryptosystem.

```
CipherSuite TLS_PSK_WITH_CAMELLIA_128_GCM_SHA256        = {0xC0,0x8D};
CipherSuite TLS_PSK_WITH_CAMELLIA_256_GCM_SHA384        = {0xC0,0x8F};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256    = {0xC0,0x90};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384    = {0xC0,0x91};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_128_GCM_SHA256    = {0xC0,0x92};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_256_GCM_SHA384    = {0xC0,0x93};
CipherSuite TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256        = {0xC0,0x94};
CipherSuite TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384        = {0xC0,0x95};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256    = {0xC0,0x96};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384    = {0xC0,0x97};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256    = {0xC0,0x98};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384    = {0xC0,0x99};
CipherSuite TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256  = {0xC0,0x9A};
CipherSuite TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384  = {0xC0,0x9B};
```

## 3. Cipher Suite Definitions

### 3.1. Key Exchange

The RSA, DHE_RSA, DH_RSA, DHE_DSS, DH_DSS, ECDH, DH_anon, and ECDHE key exchanges are performed as defined in RFC 5246 [8].

### 3.2. Cipher

This document describes cipher suites based on Camellia cipher using CBC mode and GCM. The details are as follows.

The CAMELLIA_128_CBC cipher suites use Camellia [4] in CBC mode with a 128-bit key and 128-bit Initialization Vector (IV); the CAMELLIA_256_CBC cipher suites use a 256-bit key and 128-bit IV.

Advanced Encryption Standard (AES) [19] authenticated encryption with additional data algorithms, AEAD_AES_128_GCM and AEAD_AES_256_GCM, are described in RFC 5116 [7]. AES GCM cipher suites for TLS are described in RFC 5288 [9]. AES and Camellia share common characteristics including key sizes and block length. CAMELLIA_128_GCM and CAMELLIA_256_GCM are defined according to those of AES.

### 3.3. PRFs

The hash algorithms and pseudorandom function (PRF) algorithms for TLS 1.2 [8] SHALL be as follows:

a. The cipher suites ending with _SHA256 use HMAC-SHA-256 [1] as the MAC algorithm. The PRF is the TLS PRF [8] with SHA-256 [13] as the hash function.

b. The cipher suites ending with _SHA384 use HMAC-SHA-384 [1] as the MAC algorithm. The PRF is the TLS PRF [8] with SHA-384 [13] as the hash function.

When used with TLS versions prior to 1.2 (TLS 1.0 [2] and TLS 1.1 [6]), the PRF is calculated as specified in the appropriate version of the TLS specification.

### 3.4. PSK Cipher Suites

PSK cipher suites for TLS are described in RFC 5487 [11] as to SHA-256/384 and RFC 5489 [12] as to ECDHE_PSK.

## 4. Security Considerations

At the time of writing this document, there are no known weak keys for Camellia. Additionally, no security problems with Camellia have been found (see NESSIE [16], CRYPTREC [17], and LNCS 5867[18]).

The security considerations in previous RFCs (RFC 5116 [7], RFC 5289 [10], and RFC 5487 [11]) apply to this document as well.

## 5. IANA Considerations

IANA allocated the following numbers in the TLS Cipher Suite Registry:

```
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256  = {0xC0,0x72};
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384  = {0xC0,0x73};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256   = {0xC0,0x74};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384   = {0xC0,0x75};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256    = {0xC0,0x76};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384    = {0xC0,0x77};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256     = {0xC0,0x78};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384     = {0xC0,0x79};
CipherSuite TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256          = {0xC0,0x7A};
CipherSuite TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384          = {0xC0,0x7B};
CipherSuite TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x7C};
CipherSuite TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x7D};
```

```
CipherSuite TLS_DH_RSA_WITH_CAMELLIA_128_GCM_SHA256       = {0xC0,0x7E};
CipherSuite TLS_DH_RSA_WITH_CAMELLIA_256_GCM_SHA384       = {0xC0,0x7F};
CipherSuite TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x80};
CipherSuite TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x81};
CipherSuite TLS_DH_DSS_WITH_CAMELLIA_128_GCM_SHA256       = {0xC0,0x82};
CipherSuite TLS_DH_DSS_WITH_CAMELLIA_256_GCM_SHA384       = {0xC0,0x83};
CipherSuite TLS_DH_anon_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x84};
CipherSuite TLS_DH_anon_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x85};
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256  = {0xC0,0x86};
CipherSuite TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384  = {0xC0,0x87};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256   = {0xC0,0x88};
CipherSuite TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384   = {0xC0,0x89};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256    = {0xC0,0x8A};
CipherSuite TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384    = {0xC0,0x8B};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256     = {0xC0,0x8C};
CipherSuite TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384     = {0xC0,0x8D};
CipherSuite TLS_PSK_WITH_CAMELLIA_128_GCM_SHA256          = {0xC0,0x8E};
CipherSuite TLS_PSK_WITH_CAMELLIA_256_GCM_SHA384          = {0xC0,0x8F};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x90};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x91};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_128_GCM_SHA256      = {0xC0,0x92};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_256_GCM_SHA384      = {0xC0,0x93};
CipherSuite TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256          = {0xC0,0x94};
CipherSuite TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384          = {0xC0,0x95};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256      = {0xC0,0x96};
CipherSuite TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384      = {0xC0,0x97};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256      = {0xC0,0x98};
CipherSuite TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384      = {0xC0,0x99};
CipherSuite TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256    = {0xC0,0x9A};
CipherSuite TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384    = {0xC0,0x9B};
```

## 6. References

### 6.1. Normative References

[1] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[2] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

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

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

[5] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[6] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[7] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

[8] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[9] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, August 2008.

[10] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)", RFC 5289, August 2008.

[11] Badra, M., "Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode", RFC 5487, March 2009.

[12] Badra, M. and I. Hajjeh, "ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)", RFC 5489, March 2009.

[13] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS PUB 180, October 2008, <http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf>.

[14] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) for Confidentiality and Authentication", Special Publication 800-38D, April 2006, <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>.

## 6.2. Informative References

[15] Kato, A., Kanda, M., and S. Kanno, "Camellia Cipher Suites for TLS", RFC 5932, June 2010.

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

[17] "CRYPTREC (Cryptography Research and Evaluation Committees)", <http://www.cryptrec.go.jp/english/estimation.html>.

[18] Mala, H., Shakiba, M., Dakhilalian, M., and G. Bagherikaram, "New Results on Impossible Differential Cryptanalysis of Reduced Round Camellia-128", LNCS 5867, November 2009, <http://www.springerlink.com/content/e55783u422436g77/>.

[19] 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>.

## Authors' Addresses

Satoru Kanno
NTT Software Corporation

Phone: +81-45-212-9803
Fax: +81-45-212-9800
EMail: kanno.satoru@po.ntts.co.jp

Masayuki Kanda
NTT

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