

TLS Working Group
Internet-Draft
Intended status: Informational
Expires: August 16, 2014

D. McGrew
Cisco Systems
D. Bailey
RSA/EMC
M. Campagna
R. Dugal
Certicom Corp.
February 12, 2014

AES-CCM ECC Cipher Suites for TLS
draft-mcgrew-tls-aes-ccm-ecc-08

Abstract

This memo describes the use of the Advanced Encryption Standard (AES) in the Counter and CBC-MAC Mode (CCM) of operation within Transport Layer Security (TLS) to provide confidentiality and data origin authentication. The AES-CCM algorithm is amenable to compact implementations, making it suitable for constrained environments, while at the same time providing a high level of security. The ciphersuites defined in this document use Elliptic Curve Cryptography (ECC), and are advantageous in networks with limited bandwidth.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on August 16, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

Internet-Draft

AES-CCM ECC TLS

February 2014

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Conventions Used In This Document	3
2.	ECC based AES-CCM Cipher Suites	3
2.1.	AEAD algorithms	5
2.2.	Requirements on Curves and Hashes	5
3.	TLS Versions	5
4.	History	6
5.	IANA Considerations	6
6.	Security Considerations	7
6.1.	Perfect Forward Secrecy	7
6.2.	Counter Reuse	7
6.3.	Hardware Security Modules	7
7.	Acknowledgements	7
8.	References	8
8.1.	Normative References	8
8.2.	Informative References	9
Appendix A.	Recommended Curves and Algorithms	9
	Authors' Addresses	10

1. Introduction

This document describes the use of Advanced Encryption Standard (AES) [[AES](#)] in Counter with CBC-MAC Mode (CCM) [[CCM](#)] in several TLS ciphersuites. AES-CCM provides both authentication and confidentiality and uses as its only primitive the AES encrypt operation (the AES decrypt operation is not needed). This makes it amenable to compact implementations, which is advantageous in constrained environments. Of course, adoption outside of constrained environments is necessary to enable interoperability, such as that between web clients and embedded servers, or between embedded clients and web servers. The use of AES-CCM has been specified for IPsec ESP [[RFC4309](#)] and 802.15.4 wireless networks [[IEEE802154](#)].

Authenticated encryption, in addition to providing confidentiality for the plaintext that is encrypted, provides a way to check its integrity and authenticity. Authenticated Encryption with Associated Data, or AEAD [[RFC5116](#)], adds the ability to check the integrity and authenticity of some associated data that is not encrypted. This memo utilizes the AEAD facility within TLS 1.2 [[RFC5246](#)] and the AES-CCM-based AEAD algorithms defined in [[RFC5116](#)] and [[RFC6655](#)]. All of these algorithms use AES-CCM; some have shorter authentication tags, and are therefore more suitable for use across networks in which bandwidth is constrained and message sizes may be small.

The ciphersuites defined in this document use Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) as their key establishment mechanism; these ciphersuites can be used with DTLS [[RFC6347](#)].

1.1. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

2. ECC based AES-CCM Cipher Suites

The ciphersuites defined in this document are based on the AES-CCM authenticated encryption with associated data (AEAD) algorithms AEAD_AES_128_CCM and AEAD_AES_256_CCM described in [[RFC5116](#)]. The following ciphersuites are defined:

```
CipherSuite TLS_ECDHE_ECDSA_WITH_AES_128_CCM = {TBD1,TBD1}
CipherSuite TLS_ECDHE_ECDSA_WITH_AES_256_CCM = {TBD2,TBD2}
CipherSuite TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 = {TBD3,TBD3}
CipherSuite TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 = {TBD4,TBD4}
```

These ciphersuites make use of the AEAD capability in TLS 1.2 [[RFC5246](#)]. Note that each of these AEAD algorithms uses AES-CCM. Ciphersuites ending with "8" use eight-octet authentication tags; the other ciphersuites have 16 octet authentication tags.

The HMAC truncation option described in [Section 7 of \[RFC6066\]](#) (which negotiates the "truncated_hmac" TLS extension) does not have an effect on the cipher suites defined in this note, because they do not use HMAC to protect TLS records.

The "nonce" input to the AEAD algorithm is as defined in [[RFC6655](#)].

In DTLS, the 64-bit seq_num field is the 16-bit DTLS epoch field concatenated with the 48-bit sequence_number field. The epoch and sequence_number appear in the DTLS record layer.

This construction allows the internal counter to be 32-bits long, which is a convenient size for use with CCM.

These ciphersuites make use of the default TLS 1.2 Pseudorandom Function (PRF), which uses HMAC with the SHA-256 hash function.

The ECDHE_ECDSA key exchange is performed as defined in [[RFC4492](#)], with the following additional stipulations:

- o Curves with a cofactor equal to one SHOULD be used; this simplifies their use.
- o The uncompressed point format MUST be supported. Other point formats MAY be used.

- o The client SHOULD offer the `elliptic_curves` extension and the server SHOULD expect to receive it.
- o The client MAY offer the `ec_point_formats` extension, but the server need not expect to receive it.
- o [\[RFC6090\]](#) MAY be used as an implementation method.
- o Following [\[RFC4492\]](#), the server's certificate MUST contain a suitable ECC public key, and MUST be signed with a suitable ECC public key. The elliptic curve and hash function SHOULD be selected to ensure a uniform security level; guidance on acceptable choices of hashes and curves that can be used with each ciphersuite is detailed in [Section 2.2](#). The Signature Algorithms extension ([Section 7.4.1.4.1 of \[RFC5246\]](#)) SHOULD be used to indicate support of those signature and hash algorithms. If a client certificate is used, the same criteria SHOULD apply to it.

Implementations of these ciphersuites will interoperate with [\[RFC4492\]](#), but can be more compact than a full implementation of that RFC.

[2.1.](#) AEAD algorithms

The following AEAD algorithms are used:

AEAD_AES_128_CCM is used in the TLS_ECDHE_ECDSA_WITH_AES_128_CCM ciphersuite,

AEAD_AES_256_CCM is used in the TLS_ECDHE_ECDSA_WITH_AES_256_CCM ciphersuite,

AEAD_AES_128_CCM_8 is used in the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 ciphersuite, and

AEAD_AES_256_CCM_8 is used in the TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 ciphersuite.

[2.2.](#) Requirements on Curves and Hashes

Implementations SHOULD select elliptic curves and hash functions so that AES-128 is used with a curve and a hash function supporting a 128-bit security level, and AES-256 is used with a curve and a hash function supporting a 192-bit or 256-bit security level. More

detailed guidance on cryptographic parameter selection is given in [\[SP800-57\]](#) (see especially Tables 2 and 3).

[Appendix A](#) describes suitable curves and hash functions that are widely available.

[3.](#) TLS Versions

These ciphersuites make use of the authenticated encryption with additional data defined in TLS 1.2 [\[RFC5288\]](#). They MUST NOT be negotiated in older versions of TLS. Clients MUST NOT offer these cipher suites if they do not offer TLS 1.2 or later. Servers which select an earlier version of TLS MUST NOT select one of these cipher suites. Earlier versions do not have support for AEAD; for instance, the TLSCiphertext structure does not have the "aead" option in TLS 1.1. Because TLS has no way for the client to indicate that it supports TLS 1.2 but not earlier, a non-compliant server might potentially negotiate TLS 1.1 or earlier and select one of the cipher suites in this document. Clients MUST check the TLS version and generate a fatal "illegal_parameter" alert if they detect an incorrect version.

[4.](#) History

The 08 version changed a MUST to a SHOULD to align with [\[RFC4492\]](#) and tweaked text identified during IESG review. It also adds text describing the unfortunate interaction between PKCS 11 and the TLS AEAD ciphersuites that Mike StJohns identified.

The 07 version removed the mandatory-to-implement elliptic curves and hash functions, and replaced them with non-normative guidance, which is in [Appendix A](#).

The 06 version replaced obsoleted references with updated ones to [RFC6066](#), [RFC6655](#), [RFC5246](#), fixes a boilerplate error, and corrects the section reference for the truncated HMAC RFC. It also changes the mandatory-to-implement curves and hash algorithms to be less

restrictive, so that the specification can potentially be used with curves other than secp256r1, secp384r1, and secp521r1. A reference to SP 800-57 was added to provide guidance on parameter selection.

The 05 version updated the IANA considerations.

The 04 version changed the intended status to "Informational", and removed the redundant definition of the AEAD nonce and replaced it with a reference to [draft-mcgrew-tls-aes-ccm](#), to avoid incompatible descriptions.

The 03 version removed materials that are redundant with [draft-mcgrew-tls-aes-ccm](#), and replaced them with references to that draft. That draft has been approved for RFC and will be a suitable stable normative reference.

The 02 version removed the AEAD_AES_128_CCM_12 and AEAD_AES_256_CCM_12 AEAD algorithms, because they were not needed in any ciphersuites. The AES-256 ciphersuites were retained, however, to provide a secure cipher for use with the higher security curves secp384r1 and secp521r1.

This section is to be removed by the RFC editor upon publication.

[5.](#) IANA Considerations

IANA is requested to assign the values for the ciphersuites defined in Section [Section 2](#) from the TLS and DTLS CipherSuite registries. IANA, please note that the DTLS-OK column should be marked as "Y" for each of these algorithms.

[6.](#) Security Considerations

[6.1.](#) Perfect Forward Secrecy

The perfect forward secrecy properties of ephemeral Diffie-Hellman ciphersuites are discussed in the security analysis of [\[RFC5246\]](#). This analysis applies to the ECDHE ciphersuites.

[6.2.](#) Counter Reuse

AES-CCM security requires that the counter is never reused. The nonce construction in [Section 2](#) is designed to prevent counter reuse.

[6.3.](#) Hardware Security Modules

A ciphersuite can be implemented in such a way that the secret keys and private keys are stored inside a Hardware Security Module (HSM), and the cryptographic operations involving those keys are performed by the HSM on data provided by an application interacting with the HSM through an interface such as that defined by the Cryptographic Token Interface Standard [[PKCS11](#)]. When an AEAD ciphersuite, such as those in this note, are implemented in this way, special handling of the nonce is required. This is because the "salt" part of the nonce is set to the client_write_IV or server_write_IV, which is a function of the TLS master secret.

Another potential issue with the Cryptographic Token Interface Standard is that the use of the DecryptUpdate function is not possible with the CCM decrypt operation, or the decrypt operation any other authenticated encryption method. This is because the DecryptUpdate requires that post-decryption plaintext be returned before the authentication check is completed.

[7.](#) Acknowledgements

This draft borrows heavily from [[RFC5288](#)]. Thanks are due to Robert Cragie for his great help in making this work complete, correct, and useful, and to Peter Dettman for his review. Thanks also to Mike StJohns for pointing out the HSM issues.

This draft is motivated by the considerations raised in the Zigbee Smart Energy 2.0 working group.

[8.](#) References

[8.1.](#) Normative References

- [AES] National Institute of Standards and Technology, "Specification for the Advanced Encryption Standard (AES)", FIPS 197, November 2001.
- [CCM] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality", SP 800-38C, May 2004.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", [RFC 4492](#), May 2006.
- [RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", [RFC 5116](#), January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", [RFC 5288](#), August 2008.
- [RFC5639] Lochter, M. and J. Merkle, "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation", [RFC 5639](#), March 2010.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), January 2011.
- [RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", [RFC 6090](#), February 2011.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", [RFC 6347](#), January 2012.
- [RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", [RFC 6655](#), July 2012.
- [SP800-57] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1: General (Revision 3)", SP 800-57 Part 1, July 2012.

[8.2.](#) Informative References

- [IEEE802154]
Institute of Electrical and Electronics Engineers,
"Wireless Personal Area Networks", IEEE Standard 802.15.4-
2006, 2006.
- [PKCS11] RSA Laboratories, "PKCS #11: Cryptographic Token Interface
Standard version 2.20", Public Key Cryptography
Standards PKCS#11-v2.20, 2004.
- [RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM
Mode with IPsec Encapsulating Security Payload (ESP)",
[RFC 4309](#), December 2005.

[Appendix A.](#) Recommended Curves and Algorithms

This memo does not mandate any particular elliptic curves or cryptographic algorithms, for the sake of flexibility. However, since the main motivation for the AES-CCM-ECC ciphersuites is their suitability for constrained environments, it is valuable to identify a particular suitable set of curves and algorithms.

This appendix identifies a set of elliptic curves and cryptographic algorithms that meet the requirements of this note, which are widely supported and believed to be secure.

The curves and hash algorithms recommended for each ciphersuite are:

An implementation that includes either
TLS_ECDHE_ECDSA_WITH_AES_128_CCM or
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 SHOULD support the secp256r1
curve and the SHA-256 hash function.
An implementation that includes either
TLS_ECDHE_ECDSA_WITH_AES_256_CCM or
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 SHOULD support the secp384r1
curve and the SHA-384 hash function, and MAY support the secp521r1
curve and the SHA-512 hash function.

More information about the secp256r1, secp384r1, and secp521r1 curves is available in [Appendix A of \[RFC4492\]](#).

It is not necessary to implement the above curves and hash functions in order to conform to this specification. Other elliptic curves, such as the Brainpool curves [[RFC5639](#)] for example, meet the criteria

laid out in this memo.

McGrew, et al.

Expires August 16, 2014

[Page 9]

Internet-Draft

AES-CCM ECC TLS

February 2014

Authors' Addresses

David McGrew
Cisco Systems
13600 Dulles Technology Drive
Herndon, VA 20171
USA

Email: mcgrew@cisco.com

Daniel V. Bailey
RSA/EMC
174 Middlesex Tpke.
Bedford, MA 01463
USA

Email: dbailey@rsa.com

Matthew Campagna
Certicom Corp.
5520 Explorer Drive #400
Mississauga, Ontario L4W 5L1
Canada

Email: mcampagna@certicom.com

Robert Dugal
Certicom Corp.
5520 Explorer Drive #400
Mississauga, Ontario L4W 5L1
Canada

Email: rdugal@certicom.com

