Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: 16 May 2026

F. Denis
Fastly Inc.
S. Lucas
Individual Contributor

12 November 2025

AEGIS-based Cipher Suites for TLS 1.3, DTLS 1.3 and QUIC
draft-denis-tls-aegis-05

Abstract

This document proposes new cipher suites based on the AEGIS family of authenticated encryption algorithms for integration into the TLS 1.3, DTLS 1.3, and QUIC protocols.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-denis-tls-aegis/.

Source for this draft and an issue tracker can be found at https://github.com/jedisct1/draft-denis-tls-aegis.

Table of Contents

1. Introduction and Rationale

AEGIS [I-D.irtf-cfrg-aegis-aead] is a family of authenticated encryption algorithms designed for high-performance applications. AEGIS targets the same hardware class as AES-GCM, distinguishing itself through the following key attributes:

1. Reduced memory requirements: AEGIS eliminates the need for a key schedule and precomputation tables, resulting in lower memory demands. This characteristic is particularly advantageous for servers managing a large number of connections.

2. Extended usage limits: AEGIS features higher usage limits, reducing the need for frequent rekeying compared to other available options.

3. Enhanced overall performance: AEGIS is highly efficient on CPUs supporting AES-specific instructions.

AEGIS ciphers integrate seamlessly into established protocols like TLS 1.3 by adhering to the same interface standards as existing algorithms.

This document introduces new cipher suites based on the AEGIS algorithms and outlines the procedures for their incorporation into the TLS 1.3 [RFC8446], DTLS 1.3 [RFC9147], and QUIC [RFC9000] protocols.

2. Conventions and Definitions

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.

3. New Cipher Suites and Preservation of TLS 1.3 Mechanisms

The TLS 1.3 protocol includes a set of mandatory cipher suites listed in [RFC8446], Section 9.1.

Each cipher suite specifies the Authenticated Encryption with Associated Data (AEAD) algorithm for record protection, along with the hash algorithm for use with the HMAC-based Key Derivation Function (HKDF).

The cipher suites and cryptographic negotiation mechanisms established in TLS 1.3 are reused by the DTLS 1.3 and QUIC protocols.

This document introduces additional cipher suites to accommodate AEGIS-based encryption algorithms:

Cipher Suite Name	AEAD Algorithm	Hash Algorithm	Confidentiality Level
TLS_AEGIS_128L_SHA256	AEGIS-128L	SHA256	128 bits
TLS_AEGIS_128X2_SHA256	AEGIS-128X2	SHA256	128 bits
TLS_AEGIS_128X4_SHA256	AEGIS-128X4	SHA256	128 bits
TLS_AEGIS_256_SHA512	AEGIS-256	SHA512	256 bits
TLS_AEGIS_256X2_SHA512	AEGIS-256X2	SHA512	256 bits
TLS_AEGIS_256X4_SHA512	AEGIS-256X4	SHA512	256 bits

Table 1: Proposed AEGIS-based cipher suites

The rationale for recommending the SHA512 hash function for variants employing a 256-bit key is based on the findings presented in [M23].

AEGIS algorithms support both 128-bit and 256-bit authentication tags. For all the cipher suites specified herein, these algorithms MUST be used with a 128-bit authentication tag.

With the inclusion of these new cipher suites, the cryptographic negotiation mechanism in TLS 1.3, as outlined in [RFC8446], Section 4.1.1, remains unchanged, as does the record payload protection mechanism specified in [RFC8446], Section 5.2.

4. DTLS 1.3 Record Number Encryption

In DTLS 1.3, encryption of record sequence numbers follows the specification detailed in [RFC9147], Section 4.2.3.

For AEGIS-based cipher suites, the mask is generated using the AEGIS Stream and ZeroPad functions defined in [I-D.irtf-cfrg-aegis-aead] with:

* a 128-bit tag length

* sn_key, as defined in [RFC9147], Section 4.2.3

* ciphertext[0..16]: the first 16 bytes of the DTLS ciphertext

* nonce_len: the AEGIS nonce length, either 128 or 256 bits, depending on the selected AEAD algorithm.

A 48-bit mask is computed as follows:

```
mask = Stream(48, sn_key, ZeroPad(ciphertext[0..16], nonce_len))
```

5. QUIC Header Protection

In QUIC, specific segments of the QUIC packet headers undergo encryption in accordance with the specification outlined in [RFC9001], Section 5.4.

For AEGIS-based cipher suites, the mask is generated following the same procedure as in DTLS 1.3, utilizing:

* a 128-bit tag length

* hp_key, as defined in [RFC9001], Section 5.4

* ciphertext[0..16]: the first 16 bytes of the ciphertext

* nonce_len: the AEGIS nonce length, either 128 or 256 bits, depending on the selected AEAD algorithm.

A 48-bit mask is computed as follows:

```
mask = Stream(48, hp_key, ZeroPad(ciphertext[0..16], nonce_len))
```

6. Operational Considerations

On devices lacking hardware AES acceleration or protection against side-channel attacks, cipher suites dependent on the AES round function SHOULD NOT be prioritized. This recommendation includes the cipher suites outlined in this document.

On devices equipped with secure hardware AES acceleration, implementations SHOULD prioritize AEGIS-based cipher suites over AES-GCM cipher suites of equivalent security levels.

7. Implementation Status

This note is to be removed before publishing as an RFC.

A list of early implementations can be found at https://github.com/jedisct1/draft-denis-tls-aegis.

8. Security Considerations

A key update MUST be performed before encrypting 2^48 records with the same key. The prescribed mechanism is documented in [RFC8446], Section 4.6.3.

9. IANA Considerations

IANA has registered the following identifiers in the TLS Cipher Suite Registry:

Value	Description	DTLS-OK	Recommended
0x13,0x06	TLS_AEGIS_128L_SHA256	Y	N
0x13,0x07	TLS_AEGIS_256_SHA512	Y	N

Table 2: Assigned IANA identifiers

Implementations MAY use the following identifiers reserved for local testing:

Test Value	Description	DTLS-OK	Recommended
0xff01	TLS_AEGIS_128X2_SHA256	Y	N
0xff02	TLS_AEGIS_256X2_SHA512	Y	N
0xff03	TLS_AEGIS_128X4_SHA256	Y	N
0xff04	TLS_AEGIS_256X4_SHA512	Y	N

Table 3: Additional identifiers

IANA is requested to assign the final identifiers.

10. References

10.1. Normative References

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/rfc/rfc9147>.

10.2. Informative References

[I-D.irtf-cfrg-aegis-aead] Denis, F. and S. Lucas, "The AEGIS Family of Authenticated Encryption Algorithms", Work in Progress, Internet-Draft, draft-irtf-cfrg-aegis-aead-18, 5 October 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-aegis-aead-18>.

[M23] Mattsson, J. P., "Hidden Stream Ciphers and TMTO Attacks on TLS 1.3, DTLS 1.3, QUIC, and Signal", Cryptology ePrint Archive, Paper 2023/913, DOI 10.1007/978-981-99-7563-1_12, 2023, <https://eprint.iacr.org/2023/913.pdf>.

Appendix A. Examples

A.1. TLS 1.3 Handshake

A.1.1. With TLS_AEGIS_128L_SHA256

```
   shared_key:            cbb2b72da2bc70eb85fae05a8f6bc929
                          6f3e2f9693e5972a7b2a3da608e5eda2

   hello_hash:            b77594edb8abd3acc4db7f5ead5869e1
                          96fff7d0fb1beb2bffbaac850bf479d8

   early_secret:          33ad0a1c607ec03b09e6cd9893680ce2
                          10adf300aa1f2660e1b22e10f170f92a

   handshake_secret:      15614a4e6a6c590f16e9760dc20002a1
                          2af27d6ceda73c66a9477de4b690639f

   client_secret:         6e60b228fdd7c8b08ac50e5018fa79ec
                          3f8cd2ee023386111b0d7a2027e5c1b8

   client_handshake_key: 2474bdcd8e8c8dff18af9e169e4470ea

   client_handshake_iv:  42fe48bd086cc5ddaf43be4500d0c7f2

   server_handshake_key: e0d7ea14104a89cfdf253e1f0e0302b0

   server_handshake_iv:  cc421814028367299508e120a7cb3ad2
```

A.1.2. With TLS_AEGIS_256_SHA512

```
   shared_key:           724d41a7ccadc6435d4305dd6756bd01
                         5e26dd0544a19733a2c08430f128b218

   hello_hash:           1a8fd72e2630e12817d768bae1248367
                         30c07141c4ab4cc3423d7f16c3c1a84b
                         91d4c4194453dbc85fca8738b4e9ea3c
                         783bb6d99f579fd6c2f599c69c1c79e1

   early_secret:         fd4a40cb6252b3c08d9b88d5bde85339
                         03caa51a1dba1c79ce18eea0365d35d0
                         71e597a2b95214821100e812f7b79828
                         498f164707cd63c6f7464973cfa22046

   handshake_secret:     55ef8c23352da78bf1daa4626445c883
                         b842bec578769fe9ae6fbf6de5c28953
                         02ec3cbb22b3a94ea1d047ab08cce64e
                         1079f3dbc9bf08152dc3b0bcd74ac977

   client_secret:        728f1edab4426f4dac3f03180b0bc537
                         a0d555514b439ea4f4cccb5910834807
                         408d29b9c79dcbff8e3a3fb8bf220907
                         d96ce595eee7ffaf9f9735e4f6da1e60

   client_handshake_key: 08a37693b14937177d75149422944c34
                         9019de948f6922c2c516d941c0bdafe4

   client_handshake_iv:  e0a2155fedcb592a29588bdcf06334f0
                         4dc6b5c40e659051e62071cb87f8be2c

   server_handshake_key: 366e1ebfb124508aa69137ccef542756
                         c0a748525c5bdc16acd79c66856e7c82

   server_handshake_iv:  8f883c1bb0eae38960efdb717f6b19cf
                         c929d565ad596f1f4b3daab498a7fc29
```

A.2. DTLS 1.3 and QUIC Header Protection Mask

A.2.1. With TLS_AEGIS_128L_SHA256

```
   key:                  000102030405060708090a0b0c0d0e0f

   ciphertext[0..16]:    101112131415161718191a1b1c1d1e1f

   nonce_len:            128 bits

   mask:                 60ede1c811
```

A.2.2. With TLS_AEGIS_128X2_SHA256

```
key:                  000102030405060708090a0b0c0d0e0f

ciphertext[0..16]:    101112131415161718191a1b1c1d1e1f

nonce_len:            128 bits

mask:                 6bf2292472
```

A.2.3. With TLS_AEGIS_256_SHA512

```
key:                  000102030405060708090a0b0c0d0e0f
                      101112131415161718191a1b1c1d1e1f

ciphertext[0..16]:    202122232425262728292a2b2c2d2e2f

nonce_len:            256 bits

mask:                 6e3a2ce297
```

A.2.4. With TLS_AEGIS_256X2_SHA512

```
key:                  000102030405060708090a0b0c0d0e0f
                      101112131415161718191a1b1c1d1e1f

ciphertext[0..16]:    202122232425262728292a2b2c2d2e2f

nonce_len:            256 bits

mask:                 7a515cfb0c
```

Acknowledgments

We would like to thank John Preu Mattsson for suggesting how AEGIS should be used in the context of DTLS and QUIC.

Authors' Addresses

Frank Denis
Fastly Inc.
Email: fde@00f.net

Samuel Lucas
Individual Contributor
Email: samuel-lucas6@pm.me