

TCP Maintenance (TCPM)  
Internet-Draft  
Intended status: Standards Track  
Expires: June 14, 2019

A. Sujeet Nayak  
B. Weis  
Cisco Systems  
December 11, 2018

SHA-2 Algorithm for the TCP Authentication Option (TCP-AO)  
draft-nayak-tcp-sha2-03

Abstract

The TCP Authentication Option (TCP-AO) relies on security algorithms to provide connection authentication between the two end-points. This document specifies how to use SHA-256 algorithm and attributes with TCP-AO.

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 <https://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 June 14, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://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 . . . . .	2
2. Requirements . . . . .	2
2.1. Requirements Language . . . . .	2
2.2. Algorithm Requirements . . . . .	3
3. Algorithms Specified . . . . .	3
3.1. Key Derivation Functions (KDF) . . . . .	3
3.1.1. KDF_HMAC_SHA256 . . . . .	4
3.1.2. Tips for User Interfaces Regarding KDFs . . . . .	4
3.2. MAC Algorithm . . . . .	5
3.2.1. The Use of HMAC-SHA256-128 . . . . .	5
4. Interaction with TCP . . . . .	6
5. Security Considerations . . . . .	6
6. IANA Considerations . . . . .	7
7. Acknowledgements . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

[RFC5925] describes TCP-AO mechanism to provide cryptographic authentication and message integrity verification between two end-points of a TCP connection. [RFC5926] specifies HMAC-SHA-1-96 and AES-128-CMAC-96 message authentication codes (MACs) algorithms for TCP-AO.

Although SHA-1 is considered safe for non-digital signature applications at the time of this writing [NIST-SP800-131A], there is a naturally growing demand, especially from the government and service provider community, for protecting their TCP sessions with SHA-2 family of authentication algorithms, which is considered to be relatively stronger. SHA-256, being widely preferred and deployed, provides a reasonable alternative with stronger algorithm and larger MAC length.

This document specifies usage of SHA-256 MAC algorithm on TCP-AO enabled connections. It is a companion to [RFC5925] and [RFC5926].

## 2. Requirements

## 2.1. Requirements Language

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].

When used in lowercase, these words convey their typical use in common language, and they are not to be interpreted as described in [RFC2119].

## 2.2. Algorithm Requirements

This document adopts the style and conventions of [RFC5926] in specifying how the use of new data integrity algorithm is indicated in TCP-AO. It indicates a MAC algorithm and a key derivation function (KDF).

The following table indicates the defined SHA-2 algorithm for TCP-AO:

Requirement	Authentication Algorithm
RECOMMENDED	HMAC-SHA-256-128 [RFC2104] [FIPS-180-4]

Table 1

Requirement	Key Derivation Function (KDF)
RECOMMENDED	KDF_HMAC-SHA-256

Table 2

## 3. Algorithms Specified

TCP-AO requires two classes of algorithms to be used on a particular connection namely, Key Derivation Functions (KDF) and Message Authentication Code (MAC) algorithm. Both these classes are generically described in Section 3 in [RFC5926], while focusing specifically on SHA-1 and AES-128 algorithms.

In this document, the same concept is applied to use SHA-256 algorithm.

### 3.1. Key Derivation Functions (KDF)

KDFs have the following interface:

Traffic\_Key = KDF\_alg(Master\_Key, Context, Output\_Length)

where:

Traffic\_Key, KDF\_alg, Master\_Key, Context, Output\_Length stand for entities, as described in [RFC5926], Section 3.1.

The KDF algorithm paired with corresponding pseudorandom function (PRF) is:

\* KDF\_HMAC-SHA-256 based on PRF-HMAC-SHA256 [RFC2104][FIPS-180-4]

It is based on the iteration-mode KDF specified in [NIST-SP800-108]. It uses an underlying PRF with a fixed length output of 256-bits. The KDF generates an arbitrary number of output bits by operating the PRF in a "counter" mode, where each invocation of the PRF uses a different input block, which is differentiated by a block counter.

Each input block is constructed as follows:

(i || Label || Context || Output\_Length)

Where:

"||", i, Label, Context, Output\_Length stand for entities, as described in [RFC5926], Section 3.1.1.

#### 3.1.1. KDF\_HMAC\_SHA256

For KDF\_HMAC\_SHA256:

- PRF for KDF\_alg: HMAC-SHA256 [RFC2104][FIPS-180-4]
- Use: HMAC-SHA256(Key, Input)
- Key: Master\_Key, configured by user, and passed to KDF
- Input: ( i || Label || Context || Output\_Length)
- Output\_Length: 256 bits
- Result: Traffic\_Key, used in MAC function by TCP-AO

#### 3.1.2. Tips for User Interfaces Regarding KDFs

This section provides suggested representations for the KDFs in implementation of user interfaces (UIs). Following these guidelines across common implementations will make interoperability easier and simpler for users deploying TCP-AO.

UIs SHOULD refer to the choice of KDF\_HMAC\_SHA256 as simply "SHA256".

The IANA registry values reflect this entry.

### 3.2. MAC Algorithm

Each MAC\_alg defined for TCP-AO has three fixed elements as part of its definition:

- KDF\_Alg: Name of the TCP-AO KDF algorithm used to generate the Traffic\_Key.
- Key\_Length: Length, in bits, required for the Traffic\_Key used in this MAC.
- MAC\_Length: The final length of the bits used in the TCP-AO MAC field. This value may be a truncation of the MAC function's original output length.

As described in [RFC5926], Section 3.2, MACs computed for TCP-AO have the following interface:

```
MAC = MAC_alg(Traffic_Key, Message)
```

The MAC\_alg for generating MAC, as used by TCP-AO:

- \* HMAC-SHA256-128 based on [RFC2104] and [FIPS-180-4]

HMAC-SHA256 produces 256 bits output. The MAC output is then truncated to provide a reasonable trade-off between security and message size, for fitting into the TCP-AO option field. As recommended in [RFC2104], Section 5, the HMAC-SHA256 is truncated to 128 bits.

#### 3.2.1. The Use of HMAC-SHA256-128

The three fixed elements for HMAC-SHA256-128 are:

- KDF\_Alg: KDF\_HMAC\_SHA256
- Key\_Length: 256 bits
- MAC\_Length: 128 bits

For:

```
MAC = MAC_alg (Traffic_Key, Message)
```

HMAC-SHA256-128 for TCP-AO has the following values:

- MAC\_alg: HMAC-SHA256
- Traffic\_Key: Variable; the result of the KDF
- Message: The message to be authenticated, as specified in [RFC5925], Section 5.1

#### 4. Interaction with TCP

As described in [RFC5925], Section 7.6, TCP option space is most critical in SYN segments. As compared to 96-bit Mac length of [RFC5925], using a 128-bit MAC length increases the TCP-AO space from 16 bytes to 20 bytes. Since 9 bytes of space was already available in the SYN segment (9 bytes further reduces to 5 in the presence of MSS option), implementors of this document could use it to provide a stronger authentication algorithm for the TCP connections.

For non-SYN segments, TCP-AO with 128-bit Mac length would use 20 bytes, leaving 20 bytes for other options. Out of these, 10 bytes would be consumed by timestamp, leaving around 10 bytes for a single SACK block. This limit remains the same as described in [RFC5925], Section 7.6.

Another important point to be considered by the implementations is that, in the presence of this feature, since the option space is getting pushed further, care SHOULD be taken to ensure all the options are tightly packed to avoid total options length from spilling beyond the available 40 bytes.

#### 5. Security Considerations

This document inherits all the security considerations of the TCP-AO [RFC5925] and HMAC-SHA-1 related to [RFC5926].

##### NOTE REGARDING OTHER SHA-2 ALGORITHMS:

In the SHA-2 family, another widely used algorithm in the industry is SHA512. Adopting SHA512 algorithm would mean using a MAC length of 256-bits, as recommended in [RFC2104], Section 5. At the time of writing this document, there is no sufficient space available in the TCP SYN segment to accommodate this large length, without causing backward incompatibility. To avoid this scenario, usage of SHA512 algorithm is deferred, till the time a larger TCP option space evolves.

## 6. IANA Considerations

As described in [RFC5926], Section 5, IANA has a registry with the following details:

Registry Name: Cryptographic Algorithms for TCP-AO Registration

Procedure: RFC Publication after Expert Review

The following needs to be added to this registry:

Algorithm	Reference
SHA256	This document Number

Table 3

## 7. Acknowledgements

Bertrand Duivivier, M. Rohit and Srinivas Ramprasad first suggested the need for this work.

## 8. References

### 8.1. Normative References

[FIPS-180-4]

FIPS Publication 180-4, "Secured Hash Standard", March 2012.

[NIST-SP800-108]

National Institute of Standards and Technology,  
"Recommendation for Key Derivation Using Pseudorandom  
Functions, NIST SP800-108", October 2009.

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

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP  
Authentication Option", RFC 5925, DOI 10.17487/RFC5925,  
June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.

- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, DOI 10.17487/RFC5926, June 2010, <<https://www.rfc-editor.org/info/rfc5926>>.

## 8.2. Informative References

- [NIST-SP800-131A] National Institute of Standards and Technology, "Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, NIST SP800-131A", January 2011.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.

## Authors' Addresses

Sujeet Nayak Ammunje  
Cisco Systems  
Cessna Business Park  
Bangalore, Karnataka 560 087  
India

Email: [sujeetnayak@yahoo.com](mailto:sujeetnayak@yahoo.com)

Brian Weis  
Cisco Systems

Email: [bew.stds@gmail.com](mailto:bew.stds@gmail.com)