Network Working Group
S. Chang (NIST)
R. Glenn (NIST)
November 20, 1996

Internet Draft

HMAC-SHA IP Authentication with Replay Prevention

<draft-ietf-ipsec-ah-hmac-sha-04.txt>

Status of This Memo

Distribution of this memo is unlimited.

This document is an Internet-Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

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 not appropriate to use Internet Drafts as reference material, or to cite them other than as a ``working draft'' or ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the internet-drafts Shadow Directories on:

ftp.is.co.za (Africa)
nic.nordu.net (Europe)
ds.internic.net (US East Coast)
ftp.isi.edu (US West Coast)
munnari.oz.au (Pacific Rim)

Abstract

This document describes a keyed-SHA transform to be used in conjunction with the IP Authentication Header [RFC-1826]. The particular transform is based on [HMAC-MD5]. An option is also specified to guard against replay attacks.

Contents

1. Introduction

The IP Authentication Header (AH) provides integrity and authentication for IP datagrams [RFC-1826]. The transform specified in this document uses a keyed-SHA mechanism based on [HMAC-MD5]. The mechanism uses the (key-less) SHA hash function [FIPS-180-1] which produces a message digest. When combined with an AH Key, authentication data is produced. This value is placed in the Authentication Data field of the AH [RFC-1826]. This value is also the basis for the data integrity service offered by the AH protocol.

To provide protection against replay attacks, a Replay Prevention field is included as a transform option. This field is used to help prevent attacks in which a message is stored and re-used later, replacing or repeating the original. The Security Parameters Index (SPI) [RFC-1825] is used to determine whether this option is included in the AH.

Familiarity with the following documents is assumed: "Security Architecture for the Internet Protocol" [RFC-1825], "IP Authentication Header" [RFC-1826], and "HMAC-MD5: Keyed-MD5 for Message Authentication" [HMAC-MD5].

All implementations that claim conformance or compliance with the IP Authentication Header specification [RFC-1826] SHOULD implement this HMAC-SHA transform.

1.1 Terminology

In this document, the words that are used to define the significance of each particular requirement are usually capitalized. These words are:

- MUST

This word or the adjective "REQUIRED" means that the item is an absolute requirement of the specification.

- SHOULD

This word or the adjective "RECOMMENDED" means that there might exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before taking a different course.

1.2 Keys

The AH Key is used as a shared secret between two communicating parties. The Key is not a cryptographic key as used in a traditional sense. Instead, the AH key (shared secret) is hashed with the transmitted data and thus, assures that an intervening party cannot duplicate the authentication data.

Even though an AH key is not a cryptographic key, the rudimentary concerns of cryptographic keys still apply. Consider that the algorithm and most of the data used to produce the output is known. The strength of the transform lies in the singular mapping of the key (which needs to be strong) and the IP datagram (which is known) to the authentication data. Thus, implementations should, and as frequently as possible, change the AH key. Keys need to be chosen at random, or generated using a cryptographically strong pseudo-random generator seeded with a random seed. [HMAC-MD5]

All conforming and compliant implementations MUST support a key length of 160 bits or less. Implementations SHOULD support longer key lengths as well. It is advised that the key length be chosen to be the length of the hash output, which is 160 bits for SHA. For other key lengths the following concerns MUST be considered.

A key length of zero is prohibited and implementations MUST prevent key lengths of zero from being used with this transform, since no effective authentication could be provided by a zero-length key. SHA operates on 64-byte blocks. Keys longer than 64-bytes are first hashed using SHA. The resulting hash is then used to calculate the authentication data.

1.3 Data Size

SHA generates a message digest of 160 bits. To maintain 64-bit word alignment, all conforming and compliant implementations MUST include the ability to pad the message digest to 192 bits as described in this paragraph. Implementations MAY also include the ability to use the 160 bit message digest without the pad when 64-bit alignment is not required. Padding is added by appending 32 zero bits to SHA message digest. The length of the Authentication Data, specified in the Length field of the AH in 32-bit words, should include the padding bits, if present. Upon receipt, the value of the padded bits MUST be zero and are otherwise ignored.

2. Packet Format

```
+---------------+---------------+---------------+---------------+
| Next Header   | Length        |           RESERVED            |
```

```
        +---------------+---------------+---------------+---------------+
        |                              SPI                              |
        +---------------+---------------+---------------+---------------+
        |                     Replay Prevention                         |
        |                                                               |
        +---------------+---------------+---------------+---------------+
        |                                                               |
        +                     Authentication Data                       |
        |                                                               |
        +---------------+---------------+---------------+---------------+
         1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
```

The Next Header, RESERVED, and SPI fields are specified in [RFC-1826]. The Length field is the length of the Replay Prevention field and the Authentication Data in 32-bit words.

2.1 Replay Prevention

The Replay Prevention field is a 64-bit value used to guarantee that each packet exchanged between two parties is different. Each IPsec Security Association specifies whether Replay Prevention is used for that Security Association. If Replay Prevention is NOT in use, then the Authentication Data field will directly follow the SPI field. This field is used to help prevent attacks in which a message is stored and re-used later, replacing or repeating the original.

The 64-bit field is an up counter starting at a value of 1.

The secret shared key must not be used for a period of time that allows the counter to wrap, that is, to transmit more than 2^64 packets using a single key.

Upon receipt, the replay value is assured to be increasing. The implementation may accept out of order packets. The number of packets to accept out of order is an implementation detail. If an "out of order window" is supported, the implementation shall ensure that any and all packets accepted out of order are guaranteed not to have arrived before. That is, the implementation will accept any packet at most once.

When the destination address is a multicast address, replay protection is in use, and more than one sender is sharing the same IPsec Security Association to that multicast destination address, then Replay Protection SHOULD NOT be enabled. When replay protection is desired for a multicast session having multiple senders to the same multicast destination address, each sender SHOULD have its own IPsec Security Association.

[ESP-DES-MD5] provides example code that implements a 32 packet replay window and a test routine to show how it works.

2.2 Authentication Data Calculation

The computation of the 160-bit SHA digest is described in [FIPS-180-1]. The digest is calculated over the entire IP datagram. Fields within the datagram that are variant during transit and the authentication data field itself must contain all zeros prior to the computation [RFC-1826]. The Replay Prevention field, if present, is included in the calculation.

To compute HMAC-SHA over the data 'text', the following is calculated:

    SHA (K XOR opad, SHA (K XOR ipad, text))

K denotes the secret key shared by the parties. If K is longer than 64-bytes it MUST first be hashed using SHA. In this case, K is the resulting hash. The variables 'ipad', 'opad' denote fixed strings for inner and outer padding respectively. The two strings are:

    ipad = the byte 0x36 repeated 64 times,
    opad = the byte 0x5C repeated 64 times.

The calculation of the authentication data consists of the following steps:

(1) append zeros to the end of K to create a 64 byte string (e.g., if K is of length 20 bytes it will be appended with 44 zero bytes 0x00)
(2) XOR (bitwise exclusive-OR) the 64 byte string computed in step (1) with ipad
(3) concatenate to the 64 byte string resulting from step (2) the data stream 'text'
(4) apply SHA to the stream generated in step (3)
(5) XOR the 64 byte string computed in step (1) with opad
(6) concatenate to the 64 byte string resulting from step (5) the SHA result of step (4)
(7) apply SHA to the stream generated in step (6)
(8) The sender then zero pads the resulting hash to a 64-bit boundary for word alignment. IPv4 implemenations choosing not to pad will not zero pad the resulting hash. The receiver then compares the generated 160-bit hash to the first 160-bits of authentication data contained in the AH.

A similar computation is described in more detail, along with example code and performance improvements, in [HMAC-MD5]. Implementers should consult [HMAC-MD5] for more information on this technique for keying a cryptographic hash function.

3. Security Considerations

The security provided by this transform is based on the strength of SHA, the correctness of the algorithm's implementation, the security of the key management mechanism and its implementation, the strength of the associated secret key, and upon the correctness of the implementations in all of the participating systems.

At this time there are no known cryptographic attacks against SHA [SCHNEIER]. The 160-bit digest makes SHA more resistant to brute force attacks than MD4 and MD5 which produce a 128-bit digest.

Acknowledgments

This document is largely based on text written by Hugo Krawczyk. The format used was derived from work by William Simpson and Perry Metzger. The text on replay prevention is derived directly from work by Jim Hughes.

References

[RFC-1825] R. Atkinson, "Security Architecture for the Internet Protocol", RFC-1825, August 1995.

[RFC-1826] R. Atkinson, "IP Authentication Header", RFC-1826, August 1995.

[RFC-1828] P. Metzger, W. A. Simpson, "IP Authentication using Keyed MD5", RFC-1828, August 1995.

[HMAC-MD5] H. Krawczyk, M. Bellare, R. Canetti, "HMAC-MD5: Keyed-MD5 for Message Authentication", Internet Draft, March, 1996.

[FIPS-180-1] NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995.
[URL] http://csrc.nist.gov/fips/fip180-1.txt (ascii)
[URL] http://csrc.nist.gov/fips/fip180-1.ps (postscript)

[SCHNEIER] B. Schneier, "Applied Cryptography Protocols, Algorithms, and Source Code in C", John Wiley & Sons, Inc. 1994.

[ESP-DES-MD5] J. Hughes, "Combined DES-CBC, MD5, and Replay Prevention Security Transform", Internet Draft, April, 1996.

Authors' Addresses

Shu-jen Chang
NIST
Building 820, Room 456
Gaithersburg, MD 20899

shu-jen.chang@nist.gov

Robert Glenn
NIST
Building 820, Room 455
Gaithersburg, MD 20899

rob.glenn@nist.gov