

Network Working Group  
Request for Comments: 2841  
Category: Historic  
Obsoletes: 1852

P. Metzger  
Piermont  
W. Simpson  
DayDreamer  
November 2000

## IP Authentication using Keyed SHA1 with Interleaved Padding (IP-MAC)

### Status of this Memo

This memo defines a Historic Document for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

### Abstract

This document describes the use of keyed SHA1 with the IP Authentication Header.

### Table of Contents

|                                |   |
|--------------------------------|---|
| 1. Introduction .....          | 2 |
| 1.1. Keys .....                | 2 |
| 1.2. Data Size .....           | 2 |
| 1.3. Performance .....         | 3 |
| 2. Calculation .....           | 3 |
| A. Changes .....               | 5 |
| Security Considerations .....  | 6 |
| Acknowledgements .....         | 6 |
| References .....               | 7 |
| Contacts .....                 | 8 |
| Editor's Note .....            | 8 |
| Full Copyright Statement ..... | 9 |

## 1. Introduction

The Authentication Header (AH) [RFC-1826] provides integrity and authentication for IP datagrams. This specification describes the AH use of keys with the Secure Hash Algorithm (SHA1) [FIPS-180-1]. This SHA1-IP-MAC algorithm uses a leading and trailing key (a variant of the "envelope method"), with alignment padding between both keys and data.

It should be noted that this document specifies a newer version of SHA than that described in [FIPS-180], which was flawed. The older version is not interoperable with the newer version.

This document assumes that the reader is familiar with the related document "Security Architecture for the Internet Protocol" [RFC-1825], that defines the overall security plan for IP, and provides important background for this specification.

### 1.1. Keys

The secret authentication key shared between the communicating parties SHOULD be a cryptographically strong random number, not a guessable string of any sort.

The shared key is not constrained by this transform to any particular size. Lengths of 160-bits (20 octets) MUST be supported by the implementation, although any particular key may be shorter. Longer keys are encouraged.

### 1.2. Data Size

SHA1's 160-bit output is naturally 32-bit aligned. However, many implementations require 64-bit alignment of the following headers.

Therefore, several options are available for data alignment (most preferred to least preferred):

- 1) only the most significant 128-bits (16 octets) of output are used.
- 2) an additional 32-bits (4 octets) of padding is added before the SHA1 output.
- 3) an additional 32-bits (4 octets) of padding is added after the SHA1 output.
- 4) the SHA1 output is variably bit-positioned within 192-bits (24 octets).

The size and position of the output are negotiated as part of the key management. Padding bits are filled with unspecified implementation dependent (random) values, which are ignored on receipt.

#### Discussion:

Although truncation of the output for alignment purposes may appear to reduce the effectiveness of the algorithm, some analysts of attack verification suggest that this may instead improve the overall robustness [PO95a].

### 1.3. Performance

Preliminary results indicate that SHA1 is 62% as fast as MD5, and 80% as fast as DES hashing. That is:

$$\text{SHA1} < \text{DES} < \text{MD5}$$

This appears to be a reasonable performance tradeoff, as SHA1 internal chaining is significantly longer than either DES or MD5:

$$\text{DES} < \text{MD5} < \text{SHA1}$$

#### Nota Bene:

Suggestions are sought on alternative authentication algorithms that have significantly faster throughput, are not patent-encumbered, and still retain adequate cryptographic strength.

### 2. Calculation

The 160-bit digest is calculated as described in [FIPS-180-1]. A portable C language implementation of SHA1 is available via FTP from <ftp://rand.org/pub/jim/sha.tar.gz>.

The form of the authenticated message is:

```
SHA1( key, keyfill, datagram, datafill, key, shalfill )
```

First, the variable length secret authentication key is filled to the next 512-bit boundary, using the same pad-with-length technique defined for SHA1. The padding technique includes a length that protects arbitrary length keys.

Next, the filled key is concatenated with (immediately followed by) the invariant fields of the entire IP datagram (variant fields are zeroed). This is also filled to the next 512-bit boundary, using the same pad-with-length technique defined for SHA1. The length includes the leading key and data.

Then, the filled data is concatenated with (immediately followed by) the original variable length key again. A trailing pad-with-length to the next 512-bit boundary for the entire message is added by SHA1 itself.

Finally, the 160-bit SHA1 digest is calculated, and the result is inserted into the Authentication Data field.

#### Discussion:

The leading copy of the key is padded in order to facilitate copying of the key at machine boundaries without requiring re-alignment of the following datagram. Filling to the SHA1 block size also allows the key to be prehashed to avoid the physical copy in some implementations.

The trailing copy of the key is not necessary to protect against appending attacks, as the IP datagram already includes a total length field. It reintroduces mixing of the entire key, providing protection for very long and very short datagrams, and robustness against possible attacks on the IP length field itself.

When the implementation adds the keys and padding in place before and after the IP datagram, care must be taken that the keys and/or padding are not sent over the link by the link driver.

## A. Changes

Changes from RFC 1852:

Use of SHA1 term (as always intended).

Added shortened 128-bit output, and clarify output text.

Added tradeoff text.

Changed padding technique to comply with Crypto '95 recommendations.

Updated references.

Updated contacts.

Minor editorial changes.

## Security Considerations

Users need to understand that the quality of the security provided by this specification depends completely on the strength of the SHA1 hash function, the correctness of that algorithm's implementation, the security of the key management mechanism and its implementation, the strength of the key, and upon the correctness of the implementations in all of the participating nodes.

The SHA algorithm was originally derived from the MD4 algorithm [RFC-1320]. A flaw was apparently found in the original specification of SHA [FIPS-180], and this document specifies the use of a corrected version [FIPS-180-1].

At the time of writing of this document, there are no known flaws in the SHA1 algorithm. That is, there are no known attacks on SHA1 or any of its components that are better than brute force, and the 160-bit hash size of SHA1 is substantially more resistant to brute force attacks than the 128-bit hash size of MD4 and MD5.

However, as the flaw in the original SHA1 algorithm shows, cryptographers are fallible, and there may be substantial deficiencies yet to be discovered in the algorithm.

## Acknowledgements

Some of the text of this specification was derived from work by Randall Atkinson for the SIP, SIPP, and IPv6 Working Groups.

Preliminary performance analysis was provided by Joe Touch.

Padding the leading copy of the key to a hash block boundary for increased performance was originally suggested by William Allen Simpson.

Padding the leading copy of the key to a hash block boundary for increased security was suggested by [KR95]. Including the key length for increased security was suggested by David Wagner.

Padding the datagram to a hash block boundary to avoid (an impractical) key recovery attack was suggested by [PO95b].

## References

- [FIPS-180] "Secure Hash Standard", Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department Of Commerce, May 1993.
- Also known as: 58 Fed Reg 27712 (1993).
- [FIPS-180-1] "Secure Hash Standard", National Institute of Standards and Technology, U.S. Department Of Commerce, April 1995.
- Also known as: 59 Fed Reg 35317 (1994).
- [KR95] Kaliski, B., and Robshaw, M., "Message authentication with MD5", CryptoBytes (RSA Labs Technical Newsletter), vol.1 no.1, Spring 1995.
- [PO95a] Preneel, B., and van Oorshot, P., "MDx-MAC and Building Fast MACs from Hash Functions", Advances in Cryptology -- Crypto '95 Proceedings, Santa Barbara, California, August 1995.
- [PO95b] Preneel, B., and van Oorshot, P., "On the Security of Two MAC Algorithms", Presented at the Rump Session of Crypto '95, Santa Barbara, California, August 1995.
- [RFC 1320] Rivest, R., "The MD4 Message-Digest Algorithm", RFC 1320, April 1992.
- [RFC 1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.
- [RFC 1825] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1825, July 1995.
- [RFC 1826] Atkinson, R., "IP Authentication Header", RFC 1826, July 1995.

## Contacts

Comments about this document should be discussed on the photuris@adk.gr mailing list.

This document is a submission to the IP Security Working Group of the Internet Engineering Task Force (IETF). The working group can be contacted via the current chairs:

Questions about this document can also be directed to:

Perry Metzger  
Piermont Information Systems Inc.  
160 Cabrini Blvd., Suite #2  
New York, NY 10033

EMail: perry@piermont.com

William Allen Simpson  
DayDreamer  
Computer Systems Consulting Services  
1384 Fontaine  
Madison Heights, Michigan 48071

EMail: wsimpson@UMich.edu  
wsimpson@GreenDragon.com (preferred)

## Editor's Note

This paper was originally submitted May 1, 1996.



## Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

