RFC2104 - Keyed-Hashing for Message Authentication (HMAC)

return to the IETF Standards

Table 1: Data sheet for Keyed-Hashing for Message Authentication (HMAC)

Title	Keyed-Hashing for Message Authentication
Acronym	HMAC
Version	1997
Document Number	RFC2104
Release Date	February 1997
Reference	https://tools.ietf.org/html/rfc2104

Note: The following is an excerpt from the official IETF RFC. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

Introduction

Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity check based on a secret key are usually called "message authentication codes" (MAC). Typically, message authentication codes are used between two parties that share a secret key in order to validate information transmitted between these parties. In this document we present such a MAC mechanism based on cryptographic hash functions. This mechanism, called HMAC, is based on work by the authors [1)] where the construction is presented and cryptographically analyzed. We refer to that work for the details on the rationale and security analysis of HMAC, and its comparison to other keyed-hash methods.
HMAC can be used in combination with any iterated cryptographic hash function. MD5 and SHA-1 are examples of such hash functions. HMAC also uses a secret key for calculation and verification of the message authentication values. The main goals behind this construction are

- *To use, without modifications, available hash functions. In particular, hash functions that perform well in software, and for which code is freely and widely available.*
- *To preserve the original performance of the hash function without incurring a significant degradation.*
- *To use and handle keys in a simple way.*
- *To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function.*
- *To allow for easy replaceability of the underlying hash function in case that faster or more secure hash functions are found or required.*

This document specifies HMAC using a generic cryptographic hash function (denoted by H). Specific instantiations of HMAC need to define a particular hash function. Current candidates for such hash functions include SHA-1 [2)], MD5 [3)], RIPEMD-128/160 [4)]. These different realizations of HMAC will be denoted by HMAC-SHA1, HMAC-MD5, HMAC-RIPEMD, etc.

***Note:** To the date of writing of this document MD5 and SHA-1 are the most widely used cryptographic hash functions. MD5 has been recently shown to be vulnerable to collision search attacks [5)]. This attack and other currently known weaknesses of MD5 do not compromise the use of MD5 within HMAC as specified in this document (see Dobb); however, SHA-1 appears to be a cryptographically stronger function. To this date, MD5 can be considered for use in HMAC for applications where the superior performance of MD5 is critical. In any case, implementers and users need to be aware of possible cryptoanalytic developments regarding any of these cryptographic hash functions, and the eventual need to replace the underlying hash function. (See section 6 for more information on the security of HMAC.)*

1)

M. Bellare, R. Canetti, and H. Krawczyk, “Keyed Hash Functions and Message Authentication”, Proceedings of Crypto'96, LNCS 1109, pp. 1-15. (http://www.research.ibm.com/security/keyed-md5.html)

2)

NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995.

3)

Rivest, R., “The MD5 Message-Digest Algorithm”, RFC 1321, April 1992.

4)

H. Dobbertin, A. Bosselaers, and B. Preneel, “RIPEMD-160: A strengthened version of RIPEMD”, Fast Software Encryption, LNCS Vol 1039, pp. 71-82. ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/bosselae/ripemd/

5)

H. Dobbertin, “The Status of MD5 After a Recent Attack”, RSA Labs' CryptoBytes, Vol. 2 No. 2, Summer 1996. http://www.rsa.com/rsalabs/pubs/cryptobytes.html