The Keyed-Hash Message Authentication Code Validation System (HMACVS)

Updated: May 6, 2016
Original: December 3, 2004

Sharon Keller

Lawrence E. Bassham III

Timothy A. Hall

National Institute of Standards and Technology

Information Technology Laboratory

Computer Security Division

TABLE OF CONTENTS

Update Log

5/06/16

- Added SHA-3 to SHA functions lists.

7/23/12

- Updated References to FIPS 180-4 throughout document.
- SHA-512/224 and SHA-512/256 to SHA functions lists.
- Minor corrections and edits.

1 Introduction

This document, *The Keyed-Hash Message Authentication Code Validation System (HMACVS)* specifies the procedures involved in validating implementations of the Keyed-Hash Message Authentication Code (HMAC) as specified and approved in FIPS 198-1, *The Keyed-Hash Message Authentication Code (HMAC)* [1]. The HMACVS is designed to perform automated testing on Implementations Under Test (IUTs). This document provides the basic design and configuration of the HMACVS.

This document defines the purpose, the design philosophy, and the high-level description of the validation process for HMAC. The requirements and administrative procedures to be followed by those seeking formal validation of an implementation of an HMAC are presented. The requirements described include a specification of the data communicated between the IUT and the HMACVS, the details of the tests that the IUT must pass for formal validation, and general instruction for interfacing with the HMACVS. Additionally, an appendix is also provided containing samples of input and output files for the HMACVS.

2 Scope

This document specifies the tests required to validate IUTs for conformance to HMAC specified in [1]. When applied to an IUT, the HMACVS provides testing to determine the correctness of the implementation of HMAC. The HMACVS is composed of a single test that determines the conformance to the cryptographic specification.

3 Conformance

The successful completion of the tests contained within the HMACVS is required to be validated as conforming to the HMAC standard. Testing for the cryptographic module in which the HMAC is implemented is defined in FIPS PUB 140-2, *Security Requirements for Cryptographic Modules* [2].

4 Definitions and Abbreviations

4.1 Definitions

DEFINITION	**MEANING**
CST laboratory	Cryptographic and Security Testing laboratory that operates the HMACVS
Keyed-Hash Message Authentication Code	The algorithm specified in FIPS 198-1, *The Keyed-Hash Message Authentication Code (HMAC)*

Secure Hash Algorithm	The algorithm specified in FIPS 180-4, *Secure Hash Standard (SHS)* [3] and the algorithm specified in FIPS 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*[4]

4.2 Abbreviations

ABBREVIATION	**MEANING**
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code specified in FIPS 198-1
IUT	Implementation Under Test
SHA	Secure Hash Algorithm(s) specified in FIPS 180-4
SHA-3	Secure Hash Algorithm(s) specified in FIPS 202

5 Design Philosophy of the Keyed-Hash Message Authentication Code Validation System

The HMACVS is designed to test conformance to the HMAC specification rather than provide a measure of a product's security. The validation tests are designed to assist in the detection of accidental implementation errors, and are not designed to detect intentional attempts to misrepresent conformance. Thus, validation should not be interpreted as an evaluation or endorsement of overall product security.

The HMACVS has the following design philosophy:

1. The HMACVS is designed to allow the testing of an IUT at locations remote to the HMACVS. The HMACVS and the IUT communicate data via *REQUEST* and *RESPONSE* files. The HMACVS also generates *SAMPLE* files to provide the IUT with a sample of what the *RESPONSE* file should look like.

2. The testing performed within the HMACVS utilizes statistical sampling (i.e., only a small number of the possible cases are tested); hence, the successful validation of a device does not imply 100% conformance with the standard.

6 HMACVS Test

The HMACVS tests the implementation of HMAC for its conformance to the HMAC standard. The testing for HMAC consists of one test called the Random Message Test. The Random Message Test provides 15 sets of messages and keys for each hash algorithm and hash size/key size/MAC size combination supported by the IUT.

6.1 Configuration Information

To initiate the validation process of the HMACVS, a vendor submits an application to an accredited laboratory requesting the validation of its implementation of HMAC. The vendor's implementation is referred to as the Implementation Under Test (IUT). The request for validation includes background information describing the IUT along with information needed by the HMACVS to perform the specific tests. More specifically, the request for validation includes:

1. Vendor Name;
2. Product Name;
3. Product Version;
4. Implementation in software, firmware, or hardware;
5. Processor and Operating System with which the IUT was tested if the IUT is implemented in software or firmware;
6. Brief description of the IUT or the product/product family in which the IUT is implemented by the vendor (2-3 sentences); and
7. Configuration information for the HMAC tests, including:
    - Which underlying hash algorithms are supported
        - SHA-2 algorithms: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, or SHA-512/256
        - SHA-3 algorithms: SHA3-224, SHA3-256, SHA3-384, SHA3-512
    - For each of the underlying hash algorithms specified above, supported key sizes, *K*, related to the block size, *B*, of the underlying hash algorithm. That is, does the IUT support key sizes of $K<B$, $K=B$, or $K>B$.
        - If the IUT supports key sizes of $K<B$: provide up to 2 distinct key sizes less than the block size that the IUT supports

- If the IUT supports key sizes of *K>B*: provide up to 2 distinct key sizes greater than the block size that the IUT supports; and,

- For each of the underlying hash algorithms supported, the length, *t*, in bytes, of the MAC the IUT is able to produce. Choices for the MAC lengths are:
  - SHA-2: SHA-1: 10, 12, 16, 20
  - SHA-2 and/or SHA-3: SHA-224: 14, 16, 20, 24, 28
  - SHA-2 and/or SHA-3: SHA-256: 16, 24, 32
  - SHA-2 and/or SHA-3: SHA-384: 24, 32, 40, 48
  - SHA-2 and/or SHA-3: SHA-512: 32, 40, 48, 56, 64
  - SHA-2: SHA-512/224: 14, 16, 20, 24, 28
  - SHA-2: SHA-512/256: 16, 24, 32

6.2 The Random Message Test

The Random Message Test provides a series of random message and keys to the IUT. The IUT generates an HMAC for the messages using the keys provided. The HMACVS verifies the correctness of the HMACs produced by the IUT.

The HMACVS:

A. Creates a *REQUEST* file (Filename: HMAC.req) containing:
   1. The Product Name;
   2. The algorithm being tested; and
   3. The messages and keys used as input to the HMAC algorithm.

   Note: The CST laboratory sends the *REQUEST* file to the IUT.

B. Creates a *FAX* file (Filename: HMAC.fax) containing:
   1. The information from the *REQUEST* file; and
   2. The MAC generated by the HMAC algorithm.

   Note: The CST laboratory retains the *FAX* file.

The IUT:

A. Generates the requested MACs from the messages and keys specified in the *REQUEST* file.

B. Creates a *RESPONSE* file (Filename: HMAC.rsp) containing:

1. The information from the *REQUEST* file; and
2. The MAC generated by the HMAC algorithm.

Note: The IUT sends the *RESPONSE* file to the CST laboratory for processing by the HMACVS.

The HMACVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

Appendix A References

[1] *The Keyed-Hash Message Authentication Code (HMAC)*, FIPS Publication 198-1, National Institute of Standards and Technology, July 2008.

[2] *Security Requirements for Cryptographic Modules*, FIPS Publication 140-2, National Institute of Standards and Technology, May 2001.

[3] *Secure Hash Standard (SHS)*, FIPS Publication 180-4, National Institute of Standards and Technology, August 2015.

[4] *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, FIPS Publication 202, National Institute of Standards and Technology, August 2015.