

Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256,
HMAC-SHA-384, and HMAC-SHA-512

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document provides test vectors for the HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 message authentication schemes. It also provides ASN.1 object identifiers and Uniform Resource Identifiers (URIs) to identify use of these schemes in protocols. The test vectors provided in this document may be used for conformance testing.

Table of Contents

1.	Introduction	2
2.	Conventions Used in This Document	2
3.	Scheme Identifiers	3
3.1.	ASN.1 Object Identifiers	3
3.2.	Algorithm URIs	3
4.	Test Vectors	3
4.1.	Introduction	3
4.2.	Test Case 1	4
4.3.	Test Case 2	4
4.4.	Test Case 3	5
4.5.	Test Case 4	5
4.6.	Test Case 5	6
4.7.	Test Case 6	6
4.8.	Test Case 7	7
5.	Security Considerations	7
6.	Acknowledgements	8
7.	References	8
7.1.	Normative References	8
7.2.	Informative References	8

1. Introduction

This document provides test vectors for the HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 message authentication schemes. It also provides ASN.1 object identifiers and URIs to identify use of these schemes in protocols using ASN.1 constructs (such as those built on Secure/Multipurpose Internet Mail Extensions (S/MIME) [4]) or protocols based on XML constructs (such as those leveraging XML Digital Signatures [5]).

HMAC-SHA-224 is the realization of the HMAC message authentication code [1] using the SHA-224 hash function, HMAC-SHA-256 is the realization of the HMAC message authentication code using the SHA-256 hash function, HMAC-SHA-384 is the realization of the HMAC message authentication code using the SHA-384 hash function, and HMAC-SHA-512 is the realization of the HMAC message authentication code using the SHA-512 hash function. SHA-224, SHA-256, SHA-384, and SHA-512 are all described in [2].

2. Conventions Used in This Document

The key word "SHOULD" in this document is to be interpreted as described in RFC 2119 [3].

3. Scheme Identifiers

3.1. ASN.1 Object Identifiers

The following ASN.1 object identifiers have been allocated for these schemes:

```
rsadsi OBJECT IDENTIFIER ::=
    {iso(1) member-body(2) us(840) rsadsi(113549)}

digestAlgorithm OBJECT IDENTIFIER ::= {rsadsi 2}

id-hmacWithSHA224 OBJECT IDENTIFIER ::= {digestAlgorithm 8}
id-hmacWithSHA256 OBJECT IDENTIFIER ::= {digestAlgorithm 9}
id-hmacWithSHA384 OBJECT IDENTIFIER ::= {digestAlgorithm 10}
id-hmacWithSHA512 OBJECT IDENTIFIER ::= {digestAlgorithm 11}
```

When the "algorithm" component in a value of ASN.1 type AlgorithmIdentifier (see, e.g., [4], Section 10) identifies one of these schemes, the "parameter" component SHOULD be present but have type NULL.

3.2. Algorithm URIs

The following URIs have been allocated for these schemes:

```
http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#hmac-sha-224
http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#hmac-sha-256
http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#hmac-sha-384
http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#hmac-sha-512
```

As usual, when used in the context of [5], the <ds:HMACOutputLength> element may specify the truncated length of the scheme output.

4. Test Vectors

4.1. Introduction

The test vectors in this document have been cross-verified by three independent implementations. An implementation that concurs with the results provided in this document should be interoperable with other similar implementations.

Keys, data, and digests are provided in hex.

4.2. Test Case 1

```
Key =      0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b    (20 bytes)
Data =     4869205468657265                        ("Hi There")
```

```
HMAC-SHA-224 = 896fb1128abdbf196832107cd49df33f
                47b4b1169912ba4f53684b22
```

```
HMAC-SHA-256 = b0344c61d8db38535ca8afceaf0bf12b
                881dc200c9833da726e9376c2e32cff7
```

HMAC-SHA-384 = afd03944d84895626b0825f4ab46907f
15f9dadbe4101ec682aa034c7cebc59c
faea9ea9076ede7f4af152e8b2fa9cb6

HMAC-SHA-512 = 87aa7cdea5ef619d4ff0b4241a1d6cb0
2379f4e2ce4ec2787ad0b30545e17cde
daa833b7d6b8a702038b274eaea3f4e4
be9d914eeb61f1702e696c203a126854

4.3. Test Case 2

Test with a key shorter than the length of the HMAC output.

```
Key = 4a656665 ("Jefe")
Data = 7768617420646f2079612077616e7420 ("what do ya want ")
      666f72206e6f7468696e673f ("for nothing?")
```

```
HMAC-SHA-224 = a30e01098bc6dbbf45690f3a7e9e6d0f
                8bbea2a39e6148008fd05e44
```

```
HMAC-SHA-256 = 5bdcc146bf60754e6a042426089575c7
                5a003f089d2739839dec58b964ec3843
```

HMAC-SHA-384 = af45d2e376484031617f78d2b58a6b1b
9c7ef464f5a01b47e42ec3736322445e
8e2240ca5e69e2c78b3239ecfab21649

```
HMAC-SHA-512 = 164b7a7bfbcf819e2e395f7be73b56e0a3
87bd64222e831fd610270cd7ea250554
9758bf75c05a994a6d034f65f8f0e6fd
caeab1a34d4a6b4b636e070a38bce737
```

4.4. Test Case 3

Test with a combined length of key and data that is larger than 64 bytes (= block-size of SHA-224 and SHA-256).

Key	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	
	aaaaaaaaa	(20 bytes)
Data =	dddddddddddddddddddddddddddddddddd	
	dddddddddddddddddddddddddddddddddd	
	dddddddddddddddddddddddddddddddddd	
	dddd	(50 bytes)
HMAC-SHA-224 =	7fb3cb3588c6c1f6ffa9694d7d6ad264 9365b0c1f65d69d1ec8333ea	
HMAC-SHA-256 =	773ea91e36800e46854db8ebd09181a7 2959098b3ef8c122d9635514ced565fe	
HMAC-SHA-384 =	88062608d3e6ad8a0aa2ace014c8a86f 0aa635d947ac9febe83ef4e55966144b 2a5ab39dc13814b94e3ab6e101a34f27	
HMAC-SHA-512 =	fa73b0089d56a284efb0f0756c890be9 b1b5dbdd8ee81a3655f83e33b2279d39 bf3e848279a722c806b485a47e67c807 b946a337bee8942674278859e13292fb	

4.5. Test Case 4

Test with a combined length of key and data that is larger than 64 bytes (= block-size of SHA-224 and SHA-256).

```
Key = 0102030405060708090a0b0c0d0e0f10
      111213141516171819 (25 bytes)
Data = cdcddcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
      cdcddcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
      cdcddcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
      cdd (50 bytes)
HMAC-SHA-224 = 6c11506874013cac6a2abc1bb382627c
                ec6a90d86efc012de7afec5a
HMAC-SHA-256 = 82558a389a443c0ea4cc819899f2083a
                85f0faa3e578f8077a2e3fff46729665b
HMAC-SHA-384 = 3e8a69b7783c25851933ab6290af6ca7
                7a9981480850009cc5577c6e1f573b4e
                6801dd23c4a7d679ccf8a386c674cfff
HMAC-SHA-512 = b0ba465637458c6990e5a8c5f61d4af7
                e576d97ff94b872de76f8050361ee3db
                a91ca5c11aa25eb4d679275cc5788063
                a5f19741120c4f2de2adebeeb10a298dd
```

4.6. Test Case 5

Test with a truncation of output to 128 bits.

[illegible]

```

HMAC-SHA-224 = 0e2aea68a90c8d37c988bcd9b9fca6fa8
HMAC-SHA-256 = a3b6167473100ee06e0c796c2955552b
HMAC-SHA-384 = 3abf34c3503b2a23a46efc619baef897
HMAC-SHA-512 = 415fad6271580a531d4179bc891d87a6

```

4.7. Test Case 6

Test with a key larger than 128 bytes (= block-size of SHA-384 and SHA-512).

```
Key =
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaa                                     (131 bytes)

Data =
54657374205573696e67204c61726765      ("Test Using Large")
72205468616e20426c6f636b2d53697a      ("r Than Block-Siz")
65204b6579202d2048617368204b6579      ("e Key - Hash Key")
204669727374                          (" First")
```

```
HMAC-SHA-224 = 95e9a0db962095adaebe9b2d6f0dbce2
                d499f112f2d2b7273fa6870e
HMAC-SHA-256 = 60e431591ee0b67f0d8a26aacbf5b77f
                8e0bc6213728c5140546040f0ee37f54
HMAC-SHA-384 = 4ece084485813e9088d2c63a041bc5b4
                4f9ef1012a2b588f3cd11f05033ac4c6
                0c2ef6ab4030fe8296248df163f44952
HMAC-SHA-512 = 80b24263c7c1a3ebb71493c1dd7be8b4
                9b46d1f41b4aeec1121b013783f8f352
                6b56d0f37e05f2598bd0fd2215d6a1e52
                95e64f73f63f0aec8b915a985d786598
```

4.8. Test Case 7

Test with a key and data that is larger than 128 bytes (= block-size of SHA-384 and SHA-512).

```

Key =      aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
           aaaaaa                                     (131 bytes)

Data =     54686973206973206120746573742075 ("This is a test u")
           73696e672061206c6172676572207468 ("sing a larger th")
           616e20626c6f636b2d73697a65206b65 ("an block-size ke")
           7920616e642061206c61726765722074 ("y and a larger t")
           68616e20626c6f636b2d73697a652064 ("han block-size d")
           6174612e20546865206b6579206e6565 ("ata. The key nee")
           647320746f2062652068617368656420 ("ds to be hashed ")
           6265666f7265206265696e6720757365 ("before being use")
           642062792074686520484d414320616c ("d by the HMAC al")
           676f726974686d2e ("gorithm.")

HMAC-SHA-224 = 3a854166ac5d9f023f54d517d0b39dbd
               946770db9c2b95c9f6f565d1
HMAC-SHA-256 = 9b09ffa71b942fcb27635fbcd5b0e944
               bfdc63644f0713938a7f51535c3a35e2
HMAC-SHA-384 = 6617178e941f020d351e2f254e8fd32c
               602420feb0b8fb9adccebb82461e99c5
               a678cc31e799176d3860e6110c46523e
HMAC-SHA-512 = e37b6a775dc87dbaa4dfa9f96e5e3ffd
               debd71f8867289865df5a32d20cdc944
               b6022cac3c4982b10d5eeb55c3e4de15
               134676fb6de0446065c97440fa8c6a58

```

5. Security Considerations

This document is intended to provide the identifications and test vectors for the four identified message authentication code schemes to the Internet community. No assertion of the security of these message authentication code schemes for any particular use is intended. The reader is referred to [1] for a discussion of the general security of the HMAC construction.

6. Acknowledgements

The test cases in this document are derived from the test cases in [6], although the keys and data are slightly different.

Thanks to Jim Schaad and Brad Hards for assistance in verifying the results.

7. References

7.1. Normative References

- [1] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [2] National Institute of Standards and Technology, "Secure Hash Standard", FIPS 180-2, August 2002, with Change Notice 1 dated February 2004.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

7.2. Informative References

- [4] Housley, R., "Cryptographic Message Syntax (CMS)", [RFC 3852](#), July 2004.
- [5] Eastlake 3rd, D., Reagle, J., and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", [RFC 3275](#), March 2002.
- [6] Cheng, P. and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1", [RFC 2202](#), September 1997.

Author's Address

Magnus Nystrom
RSA Security

EMail: magnus@rsasecurity.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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