Compilation of Public Comments on SP 800-224 ipd

October 3, 2024

1 Introduction

On June 28th, 2024, NIST published the initial public draft of SP 800-224 ipd, *Keyed-Hash Message Authentication Code (HMAC): Specification of HMAC and Recommendations for Message Authentication*. NIST requested public comments on all technical and editorial aspects of the publication via `SP800-224-comments@list.nist.gov` by September 6, 2024. This was announced in the SP 800-224 ipd document (contacts page and note to reviewers), in the CSRC publication page, in a NIST news item, and via a GovDelivery email.

The "note to reviewers" in the document included the following questions:

1. ***Hash functions.*** *This draft publication lists (in R1) hash functions for use in HMAC-based message authentication. Are there applications that would justify additionally approving TupleHash [6] (a variable-length hash function designed to hash tuples of input strings) and ParallelHash [6] (an efficiently parallelizable hash function, when hashing long messages) for HMAC-based message authentication?*
2. ***Maximum length of the HMAC key.*** *When using HMAC for message authentication, this draft publication recommends (in R4) not using, but does not disallow, keys with length greater than the block size b of the underlying hash function. Should NIST disallow HMAC keys longer than the block size?*
3. ***Fixed truncation length.*** *When using HMAC for message authentication, the revised requirement (R7) about the truncation length now explicitly requires that this length be fixed across the life-span of each key. Are there applications that would justify an exception to this requirement? See more details in Section 6.3.3.*

This document compiles the received public comments, after removing personalized headers and footers, branding, and contact information (e.g., addresses).

2 Received Public Comments

2.1 Comment Set #1, from anonymous sender

Thank you for the opportunity to comment on NIST SP 800-224! Please consider these comments as anonymously submitted. I have a background in mathematics but limited knowledge of cryptography and am only commenting as an interested reader.

General comments:

Overall, I found the document well-written and easy to work through for learning purposes. Specifically, the clarity of explanation in section 2, HMAC Construction, is useful. The Summary of Changes appendix is additionally useful, and a comparison of SP 800-224ipd against FIPS 198-1 does not show any substantive changes missing.

Specific comments:

The following comments address specific portions of the document:

Section	Line	Comment
1	204	Is the "M" in "Message authentication code" intended to be capitalized?
1	218	I suggest adding a hyphen in "SP 800-107r1" for consistency with other Special Publication (SP) references
2	258	As a first-time shopper, the values of ipad (0x36) and opad (0x5C) come across as unexplained magic numbers. Reviewing the paper Keying Hash Functions for Message Authentication by Bellare, Canetti, and Krawczyk, it is stated that the "values of opad and ipad were chosen to have a very simple representation (to simplify the function's specification and minimize the potential of implementation errors), and to provide a high Hamming distance between the pads." I am unsure of the intended audience of SP 800-224, so I cannot say that it should be updated to explain the reasoning of the ipad and opad values. However, to the extent the audience of this publication encompasses those new to HMAC construction, it may be useful to add a blurb or footnote addressing the reason they've been defined as 0x36 and 0x5C, respectively.
4	343	For consistency, I would remove the quotation marks around Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program, which is already italicized. In contrast, in section 1, line 166, quotation marks are not used for FIPS 198-1's full title (rather, it is only italicized).

Section	Line	Comment
5	361	The phrasing "Some computation [...] is independent of the message" comes across as odd on first read, though it seems to be an artifact of the way the word "computation" is used rather than a grammar issue (e.g., "Some work is independent of the message" reads as fine). Probably just me, here.
6.1	397	I suggest adding a hyphen in "SP 800-90" for consistency with other Special Publication (SP) references.
6.2	416	One of the "Note to Reviewers" questions asks if NIST should disallow HMAC keys longer than the block size. After reviewing SP 800-224 and RFC 2104 (and its errata), I am not seeing what utility would be gained from allowing such key sizes. Indeed, it only seems like there are downsides.
6.2	423	Should the reference to RFC 2101, IPv4 Address Behavior Today, be RFC 2104 , HMAC: Keyed-Hashing for Message Authentication? It looks like the latter is used elsewhere throughout SP 800-224.
Ref's	615	The following URL does not work when clicked (but it does work when pasted into a web browser's URL bar): https://ia.cr/2023/861.
App'x C	718	Is the em dash ("—") preceding "$\log_2$" intentional?
App'x D	748	I suggest adding a hyphen in "SP 800-107r1" for consistency with other Special Publication (SP) references.

2.2 Comment Set #2, from Harald von Fellenberb

I am a retired physicist / software engineer / security consultant, and I am the author of FEHASHMAC, a Generic Hash and HMAC Program with 52 hash functions and HMAC and/or KMAC (see https://sourceforge.net/projects/fehashmac/). All SHA3 and SHAKE variants are included, with test vectors.

Here my comments to SP 800-224.

line 190, Maximum length of the HMAC key: I would not disallow HMAC keys longer than the block size, since the block sizes of the various SHA3 variants differ, making it impractical to the casual user not to exceed the block size. An informative message may be issued if the HMAC key is longer than the block size, indicating the recommended maximum size (in bytes). It is clear that HMAX keys longer than the block size do not increase security.

line 255, *b*: please add "Hash processes b bits per round" for clarification.

line 261, *l (ell)*: please add "See Table 2 for concrete values" for clarification.

line 265, *n*: please add "See Table 2 for concrete values" for clarification.

line 297, Table 2: Why are SHAKE128 and SHAKE256 excluded from HMAC?

Question to FIPS PUB 202, August 2015

The standard defines SHA-3: Permutation-based Hash and Extendable Output Functions.

On page 20, chapter 6.1, SHA-3 Hash Functions, the four SHA3 hashes are defined with a suffix of 01, i.e.$N = M||01$. However, when looking at the Keccak reference implementation (https://github.com/gvanas/KeccakCodePackage, download of 2016-08-04), it can be seen that the suffix for the SHA3 hash codes is 0x06, not 0x01. The original KECCAK-* hash codes use the suffix 0x01 as specified in the SHA3 submission.

In SP 800-224 Appendix B. Example Test Vectors, the resulting tags are reproducible with a suffix of 0x06, not 0x01.

Page 20, chapter 6.2, SHA-3 Extendable-Output Functions. The Keccak reference implementation uses a suffix of 0x1F for SHAKE128 and SHAKE256, and not 0x0F (M || 1111).

Please bring FIPS PUB 202 and SP 800-224 in agreement with the reference implementation, which also produced the results for the Test Vectors.

Gockhausen (Switzerland), September 4, 2024

Dr. Harald von Fellenberg

HVF Security Consulting

2.3 Comment Set #3, from Canadian Centre for Cyber Security

Suggested Clarifications:

- In Clause 6.3, Lines 430-433:

  It is written that "In an existential forgery attack, after observing many (M,T) pairs, the goal is to produce a valid tag for some new message (which the adversary can choose during the attack). In a universal forgery attack, the goal is to gain the ability to forge a valid tag for any message." The distinction between these two types of attacks could be made clearer. A suggestion is to reword as: "In an existential forgery attack, after observing many (M,T) pairs, the adversary's goal is to produce a valid tag on any single new message. There are no restrictions on this message, other than it not being previously observed. Specifically, within this security model, the adversary is not concerned with the content of the message (and therefore, it can be nonsensical). Conversely, in a universal forgery attack, the adversary's goal is to gain the ability to forge a valid tag for any given arbitrary message."

- In Clause 6.3, Lines 444, 451, 453, 455, 460, 461, 473:

  NIST should consider avoiding the use of the acronym "MD" when referring to hash functions so that Merkle-Damgård constructions are not confused with the hash functions MD5, MD4, etc. Our suggestion is to replace all uses of the acronym "MD" with "Merkle-Damgård", as well as the 6.3.1 header to "HMAC with Merkle-Damgård-based hash functions". Note that on line 444, the accent on Damgård is on the wrong letter.

- In Clause 6.3.2, Line 470:

  NIST has chosen to omit a detailed comparison between the existential unforgeability when using a Merkle-Damgård versus sponge-based construction. While this is reasonable, it would be helpful to practitioners to have some degree of guidance around how to think about the existential unforgeabilty of HMAC when using a sponge-based construction so that sponge- based constructions are not avoided when they may otherwise be the most suitable option. For example, since the reference provided suggests that the generic security of HMAC is improved by using sponge constructions, NIST might consider noting in the text that the same lower bounds for Merkle-Damgård constructions also apply to sponge constructions.

Feedback: Regarding the maximum length of HMAC Key: Disallowing HMAC keys longer than the block size may simplify CAVP testing and increase confidence in testing boundary cases.