Public Comments on the Decision Proposal to Withdraw NIST Special Publication (SP) 800-107

Comment period: June 8, 2022 – July 30, 2022

On June 8, 2022, NIST's Crypto Publication Review Board announced a proposal to **withdraw** Special Publication (SP) 800-107, *Recommendation for Applications Using Approved Hash Algorithms*. The public comments that NIST received on the proposal are collected below.

More information about this review is available from NIST's Crypto Publication Review Project site.

LIST OF COMMENTS

1. Jeffery Ellison, June 8, 2022

TWIC,

BLUF: I do not believe withdraw is a good option at this time.

Justification:
Granted this publication is old/outdated. It still has its uses and really needs updated.

NIST reasons that hashtags, derivation functions, random number generation, etc. are described in more detail in other publications, correct.
The point being missed is the consolidation value. Instead of hunting in over 9 different publications you can go to one (which should refer to the others anyway). It may be a cliff note version to heavily detailed but a one source is better than jumping around. On top of that one has to know of them, be able to find the subject within, and utilize it without a desk full of references opened. Clean, concise, and referable is the bases of SPs like this.

Now I would withdraw my position if the NIST system was easier to search. Try searching for a general subject you get 100s of references. A book like 800-17 brings sufficient knowledge to a user without overloading them. If they need additional information, that's where references within the pub comes in handy.

Most need the general information with connection to the in-depth (down and dirty) when needed.

Would you remove the table of contents, acronym page, or the glossary because the information is somewhere within?

In closing:

Sometimes you need a little up front.

VR,

Jeffery Ellison

2. Boaz Shahar, July 21, 2022

Hello NIST,

As a user of NIST security standards, I want to take this opportunity to thank you for sharing your work with the public, and for your continuous effort for a more secure world.

In regards to HMAC security strength, I think that what is expected from NIST standard is to provide the HMAC resistance to Existential Forgery attack (or unforgeability - aka EUN). Existential forgery is the attack created (by an adversary) of at least one message/signature pair, (*m*,*s*), where *s* was not produced by the legitimate signer.

The security strength of EUN is a very important property to the system security designer that is required to provide a certain security strength of his product. When HMAC scheme is designed into a product, the system security designer must know what is the security strength he can get from the function he chooses.

In SP 800-107, NIST claims that the security strength of HMAC is min(security strength of K, 2C), where C is the length of the internal state variable (chaining variable) in the Merkle Damgard function of the HASH the HMAC relies on. Let's take HMAC-SHA-256 as an example.

The length of C in HMAC-SHA-256 is 256 bit. If the key used is larger than 512 bits, the Key is truncated to L bits, which is 256 bit. However, if the Key is smaller or equal 512 bit, the Key is not truncated. Now, let's assume that the key length is 384 bit. According to section 5.3.4 in SP 800-107, the strength of HMAC-SHA-256 with this key is min(384, 2*256) = 384. However, since the output HMAC size is 256 bits, an Adversary can guess the MAC for an arbitrary message with success probability of 2^-256, which yields a strength of 256 bit at most. Obviously, the strength is not 384, as dictated by the NIST formula. I think that it should be corrected to be: min(Key-size,|T|), where |T| is the length of the tag, after truncation, if occured.

Thank you for your attention and effort,

Boaz Shahar