Comments Received in Response to:

Request for Comments on Draft FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

From: Wallner, Debbie M <dmwalln@tycho.ncsc.mil>
Sent: Monday, July 07, 2014 11:27 AM
To: internal-hash
Subject: Comments on Draft FIPS 202
Attachments: Comments on Draft FIPS PUB 202 dated April 2014.docx

Please accept these comments on Draft FIPS PUB 202.

NSA Comments on Draft of FIPS PUB 202, dated April 2014

Page 1, Section 1, Footnote 4: Consider changing "relatively small" to "sufficiently small". For example, if the output length for SHAKE128 is d = 224, the collision security is 112 bits, which is smaller than 128. Thus, this is an example of an "exception" as alluded to in the footnote. As such, it suggests that an output size of d = 224 must therefore be "relatively small". However, d = 224 is the output size of SHA3-224, one of the hashes specified in this very standard, so it would seem somewhat odd to classify this output size as being "relatively small". A similar discussion is applicable to SHAKE256 with output length d = 384 (i.e., the footnote would seem to suggest that d = 384 is a "relatively small" output size, when in fact it's the output size for one of the four hashes specified in the standard).

Various sections: Make sure when defining the state array, the limits of the parameters are 0 <= x < 5, 0 <= y < 5, and 0 <= z <w. Incorrect limits are given in sections 3.1 (last paragraph), 3.1.2 (2nd paragraph), 3.1.3 (definitions of Lane(i,j) and Plane(j)), 3.2.1 (Algorithm 1), 3.2.3 (Algorithm 3), 3.2.4 (Algorithm 4), and 3.2.5 (Algorithm 6).

NIST RESPONSE: The suggested changes were accepted.

From: Babbage, Steve, Vodafone Group <Steve.Babbage@vodafone.com>
Sent: Friday, July 18, 2014 8:46 AM
To: internal-hash
Subject: Comment on Draft FIPS 202

ETSI is the European Telecommunications Standards Institute. Within ETSI, TC SAGE is the Technical Committee "Security Algorithms Group of Experts", which specifies many of the cryptographic algorithms for mobile and other telecoms standards.

ETSI TC SAGE would like to express its support for the inclusion of the Extendable-Output Functions SHAKE-128 and SHAKE-256 in the SHA-3 standard (although we prefer the word "Extensible" …). We believe that these add genuine value.

In particular we would like to draw attention to the TUAK algorithm set (http://www.3gpp.org/DynaReport/35231.htm, together with http://www.3gpp.org/DynaReport/35232.htm and http://www.3gpp.org/DynaReport/35233.htm), an authentication and key generation algorithm standardised by 3GPP for mobile telephony. The TUAK functions can all be defined very straightforwardly in terms of SHAKE-256, so that a TUAK implementation could directly and quickly be built from a SHAKE-256 implementation.
Steve Babbage, Vodafone
Chair of ETSI SAGE

NIST RESPONSE: No change to the Standard was requested.

From: Nicholls, Tom <Tom.Nicholls@thalesesec.com>
Sent: Tuesday, July 22, 2014 2:55 PM
To: internal-hash
Subject: Comment on Draft FIPS 202
Attachments: FIPS 202 Comments-Thales e-Security.pdf

Comments on Draft FIPS 202 on behalf of Thales e - Security.

Best Regards,
Tom Nicholls
Security Engineer
THALES Information Systems Security
Phone: 954.888.6271
tom.nicholls@thalesesec.com

NIST RESPONSE: The editorial comments were accepted, with a modification to the suggested resolution in one case. Although the stated rationale for the general comment is reasonable, it is preferable to omit the hyphens, as originally specified, in order to help distinguish the different roles of the parameters. In particular, the numerical suffixes in "SHAKE128" and "SHAKE256" indicate security strengths, while for the SHA-3 hash functions such as SHA3-256, the suffix indicates the digest length of the hash function.

Legend (type of comment)

E = Editorial
G= General
T= Technical

ID	ORGANIZATION	SECTION, SUBSECT & PARA.	TYPE	COMMENT	RESOLUTION
1	Thales e-Security	Section 3, 2nd paragraph	E	"The set of values for the b-bit input to the permutation, as it undergoes successive applications of the step mappings, culminating in the output, is called the state." This could be expressed more clearly. It starts off talking about the input, which is fixed, and ends up describing the state, which is mutable.	Recommend replacing the text with: "The permutation, as it undergoes successive applications of the step mappings, maintains a b-bit state, which is initially set to the input values."
2	Thales e-Security	Section 3.2.5, Algorithm 5, Step 3	E	The four 'plus' symbols should be 'xor' symbols.	Please amend.
3	Thales e-Security	Section 7, 2nd paragraph	E	"SHA3-224, SHA3-256, SHA3-384, SHA3-512 are approved hash functions …"	Missing "and" before "SHA3-512".

ID	ORGANIZATION	SECTION, SUBSECT & PARA.	TYPE	COMMENT	RESOLUTION
4	Thales e-Security	Section 6.2 (and elsewhere)	G	It would be preferable to name the XOFs 'SHAKE-128' and 'SHAKE-256' instead of 'SHAKE128' and 'SHAKE256'. This would be consistent with the naming of the hash functions 'SHA3-X'. Separation of the symbol and number with a hyphen gives a clearer indication that the number is not intrinsic to the symbol, but is a parameter of the construction.	Please amend.

ID	ORGANIZATION	SECTION, SUBSECT & PARA.	TYPE	COMMENT	RESOLUTION
5	Thales e-Security	All	G	This document defines: • Keccak[c], a family of sponge functions of width 1600 bits (parametrized by their capacity, $0 < c < 1600$); • SHA3-X, a family of hash functions parametrized by their output length X in {224, 256, 384, 512}, defined in terms of Keccak[2X]; • SHAKE-X, a pair of extendable-output functions parametrized by their security level X in {128, 256}, defined in terms of Keccak[2X]. Suggest that this standard is decomposed into constituent primitives.	Recommend splitting this standard into three standards, one for each of the defined primitives: 1. a standard defining an approved family of sponge functions, namely Keccak[c]; 2. a standard defining an approved construction for hash functions in terms of arbitrary approved sponge functions, namely SHA3-X; 3. a standard defining an approved construction for extendable-output functions in terms of arbitrary approved sponge functions, namely SHAKE-X. This would allow greater flexibility in future. For example, NIST could then: • update the XOF standard without touching the hash standard, or vice versa; • approve a different sponge function and thereby get alternative hash and XOF functions for free; • define new primitives based on the sponge construction (in addition to hashes and XOFs) with minimal disruption to existing standards.

From: clinton bowen <clinton.bowen@gmail.com>
Sent: Thursday, August 14, 2014 10:32 AM
To: internal-hash
Subject: My Comments on DRAFT FIPS 202

1) The comment:

I'm addressing this for the cause of trouble in the future. For cryptography (i.e. not security, e.g. security -> FIPS 140 & 199), I recognize that FIPS documents that are purposed to define cryptographic primitives and SP documents related to cryptography build upon the cryptographic primitives defined in FIPS publications. All functions before SHA3 defined in FIPS publications are primitives. SHA3 isn't a primitive cryptographic function. SHA3 is a composition of a primitive cryptographic permutation function with a sponge (and a pad). FIPS has never seen a permutation function as a cryptographic primitive. FIPS 202 should emphasize approved permutations (e.g. Keccak-p[b, n_r]), approved sponges with corresponding padding function(s). The SHA3 definition should be an Annex of FIPS 202 because it is a specific instance of a sponge with corresponding pad and a cryptographic primitive Keccak-p[b, n_r].

The reason for all of this is to accommodate the possibility of future approved sponges and permutations. The SHA-3 standardization page has pdf's of what is planned after FIPS 202: authenticated encryption, PRF, tree hashing, RNG. Secondly the call for papers for the NIST Hash workshop at CRYPTO 2014 already implies that the duplex sponge will be approved in some instance in the future by NIST. Whether these modes belong in a FIPS or SP document is up to NIST. The way FIPS 202 is written now, I don't see how it leaves room for other planned uses of sponges and permutations. Decades from now we'll want to read verbiage from FIPS 202 like "…use this sponge with a FIPS 202 Annex A approved permutation function…" or "… SHA4 is defined as Sponge#2[Permutation#4, Pad#6, r]". This kind of verbiage is consistent with other FIPS and SP documentation. A fortiori, the Keccak team presented a concept of this in "Keccak and the SHA-3 Standardization" that is found on the SHA-3 standardization page (see page 50/60).

2) Proposal:

FIPS 202 could be compartmentalized like FIPS 140 is compartmentalized (I don't care if it is one document or several documents). Annexes below C (i.e. D, E, F, and G) should specify the composition and modes of uses of the permutations and sponges defined in Annex A and Annex B respectively. The outline of FIPS 202 could be conceptualized as follows:

a. FIPS 202 "Permutation-Based Cryptography":
    i. A high level description of sponges and cryptographic permutations
    ii. A disclaimer that cryptographic permutations are to be used with sponges. Something similar to section 7 of the current draft.
    iii. Some verbiage on how FIPS 202 is compartmentalized.
b. Annex A: Approved Permutation Functions
    i. Keccak-p[b, n_r] -> place section3 to 3.4 of the current draft in here.
    ii. Part 2 could be left for future approved permutations.
c. Annex B: Approved Sponge Functions

i. Sponge[f,pad,r](M,d) -> place section 4 and 5.1 of the current draft in here.

ii. reserve part 2 for Duplex[f,pad,r](sigma, L) should it be approved in the near future.

iii. Part 3 could be left for future approved sponges. Examples of other types of sponges are the donkey sponge and the monkey duplex sponge

d. Annex C: Security Analysis of Permutation-Based Cryptography

i. Ask the keccak team and the academic cryptographic community nicely for help with Annex C.

e. Annex D: Approved Permutation-Based Hash and Extendable-Output Functions

i. Fixed length hash functions:

1. SHA3 -> Place section 5.2, 6, 6.1 of the current draft in here. My opinion is that 5.2, 6, and 6.1 could be merged.

ii. Extendable Output functions:

2. SHAKE -> Place section 6.2 of the current draft in here

iii. Object Identifiers

3. Don't be lazy. Place the actual identifiers of Appendix A.3 of the current draft in here. Remember, we're not going to have access to that page of the internet at all times, but perhaps we'll have a copy of the annex.

f. Annex E: Approved Permutation-Based Pseudo-random functions

i. …

g. Annex F: Approved Permutation-Based Encryption & Authenticated Encryption Methods

i. Encryption Methods:

1. …

ii. Authenticated Encryption Methods:

1. …

h. Annex G: Approved Permutation-Based Stream Ciphers

i. …

My opinion is that tree hashing and DRBG's using sha-3 or any other approved permutation based hash functions belong in SP documents and not Annexes of a FIPS since they are built around functions defined in Annex D.

3) Justification of proposal:

While this competition was supposed to result in a new hash function, the winner is really a new category(ies) of cryptographic functions, permutations with sponges. The resulting FIPS document should not be written as a document for a focused intent of hashing. It should be written for a new category of cryptography – a category that is quite flexible and can serve multiple purposes in cryptography and accommodate new uses of permutation based cryptography in the future.

Thanks,

--

-Clinton M. Bowen

NIST RESPONSE: The restructuring proposal was not accepted. The text in Section 7 on conformance already explicitly accommodates the possibility of future approved sponge functions based on the KECCAK-p permutations and other intermediate functions. Moreover, the primary goal of FIPS 202 is to standardize the winning algorithm from the SHA-3 Competition, as initiated in the Federal Register Notice on November 2, 2007. The proposed restructuring would detract from the perception of the Standard as fulfilling that goal.

From: Peter Rombouts <peter.rombouts@nxp.com>
Sent: Tuesday, August 19, 2014 9:26 AM
To: internal-hash
Subject: Comment on Draft FIPS 202

Hi,

I would like to submit the following comment on Draft FIPS 202:

FIPS 198-1 defines how to compute a keyed-hash message authentication code based on a hash function with given input block size (B) and output block size (L). In FIPS 202 the length of the digest of the hash function (d) is clearly defined, however there is no clear definition of the input block size. I suggest adding a clarification such as the one provided in section 5.1 of the round 3 submission of Keccak (Keccak-submission-3.pdf) which states that the input block size for Keccak is equal to the rate (r).

Regards,
Peter

NIST RESPONSE: The comment was accepted and addressed with new text in the conformance section.

From: Harris, Michael W. (CDC/OCOO/OCIO) <fnb0@cdc.gov>
Sent: Tuesday, August 26, 2014 7:41 AM
To: internal-hash
Cc: CDC OCOO-OCISO Data Call; Gatland-Lightner, Cheri (CDC/OCOO/OCIO); Robinson, Colleen M. (CDC/OCOO/OCIO)
Subject: Comment on Draft FIPS 202

CDC has no comments to provide on the *DRAFT FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*.

Thank you for the opportunity to review and comment.

Michael W. Harris, CISSP, Information Technology Specialist, Office of the Chief Information Security Officer (OCISO), Centers for Disease Control and Prevention (CDC)

Office: 770.488.8052, Cell: 770.283.9589, E-mail: fnb0@cdc.gov

NIST RESPONSE: No change was requested.

From: Scott Fluhrer (sfluhrer) <sfluhrer@cisco.com>
Sent: Tuesday, August 26, 2014 6:33 PM
To: internal-hash
Subject: Comments on FIPS-202

In the FIPS 202 draft, you introduce a new cryptographical primitive called a XOF. Now, you list two things people may want to do with it:

- Generate a variable length hash
- Use it as a random-looking function (as in a KDF, or an OAEP masking function)

Now these are two separate scenarios; as a hash, we assume that the attacker picks the input (and we hope he can't control the output); as a KDF, he has only probabilistic information about the input (and we hope he can't use that to obtain probabilistic information about the output).

Now, a use of a XOF as a hash (at least, as you define SHAKE-128 and SHAKE-256) isn't very interesting; for XOF output length less than the security level, that hash obviously has a weaker security level, and so it's no better than taking (say) SHA3 output and truncating it. For XOR output length greater than the security level, you don't claim any extra security, and so (from a cryptographical perspective) it's no better than taking (again) SHA3 output and adding a bunch of 0 bits.

Now, using a XOF as a KDF is rather more interesting; however defining the security properties is a lot trickier. You state the hope that SHAKE128/SHAKE256 would defend against "attacks that would be resisted by a random function of the requested length…"; that certainly states what our intuition says we want (however, it's not clear how you'd be able to come up with a formal definition of that).

My suggestion is that you don't approve an XOF as a variable length hash (because it doesn't really bring anything to the table); instead you treat it strictly as a KDF (or something that needs similar security properties).

Also, you mention that the primitive might be a bit tricky to use (because of the prefix property); on the other hand, if the user dislikes the prefix capability, he can easily avoid it by including the output length as part of the data being hashed. You might want to make that suggestion in the (future) document that describes how XOF's are allowed to be used.

--

Scott Fluhrer

NIST RESPONSE: The text in Section 7 on conformance explicitly asserts that approved uses of the extendable-output functions will be specified in NIST special publications. NIST will consider these comments in the development of those publications. Also, text was added to clarify that extendable-output functions are not yet approved as variable-length hash functions.