From: pqc-forum@list.nist.gov on behalf of Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Sent: Tuesday, November 28, 2023 10:02 AM
To: pqc-forum
Subject: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

Hi All,

While I quite like the idea of having a hash-based signature scheme with the NIST Lightweight competition winner Ascon, I'd suggest that some major changes are made in relation to the actual Ascon-Sign submission [1]:

- Remove the "Category 3" variants.
- Remove the "Robust" variants.
- Replace the old Ascon-Sign specification text and reference implementation completely.

Proposal: Specify instantiations of FIPS 205 SLH-DSA [3] with Ascon. Logical names would be SLH-DSA-ASCON-128s and SLH-DSA-ASCON-128f. These use Ascon-XOF exactly as SHAKE256 is instantiated in Section 10.1 of [3].

Specific comments;

PQ Category 3 Variants

The submission Ascon-Sign [1] with four main parameter sets, Ascon-Sign-{128s, 128f, 192s, 192f}, The variants 192s and 192f are claimed to have "Category 3" security (Table 5), i.e. equivalent security to AES-192 key search. The 192-bit variants were also provided with the reference implementation.

However, Ascon does not claim more than a 128-bit security level against any attack, including (second) pre-image attacks [2]. The Ascon-Sign submission document does not explain the discrepancy -- how a hash-based signature scheme can be more secure than its hash function.. It actually repeats the 128-bit security claim in Section 2.

(Note: Some portions of the [1] text were copy-pasted from the Ascon specification [2], while others were copied from some older SPHINCS+ v3.1 specification [4]. None of the authors of Ascon-Sign [1] have appeared as authors of either Ascon [2] or SPHINCS+ [4].)

Ascon-XOF vs Ascon-Hash

The submission document for Ascon-sign [1] states that core functions are used; Ascon-XOF is used to instantiate H_msg, while Ascon-Hash is used to instantiate other functions. However, the implementation uses a function "ascon_hash()" to implement everything, with the IV set as 0x00400c0000000100 -- the domain separator of Ascon-Hash. This function is used in XOF mode to extract outputs of various sizes, which is not allowed the domain separator limits output to 256 bits.

Furthermore, [1] states that "Ascon-Hash can be used to construct Ascon-XOF" (pg 6.) -- which shows significant confusion, contradicting the IV discussion at the beginning of Section 2 of the document [1] itself.

The "Robust" Variant

SLH-DSA [3] no longer contains the "robust" variants that were proposed for the original SPHINCS+ [4]. However, Ascon-Sign [1] has its own peculiar Ascon-based "robust" variant.

To illustrate the potential technical implications of ignoring the domain separation of Ascon, we look at "Robust" version of function F is defined in Section 3:

F(PK.seed, ADRS, M1) = Ascon-Hash(PK.seed || ADRS || M1(+)),
where M(+) = M xor Ascon-XOF(PK.seed || ADRS, l).

Since in the implementation Ascon-Hash = Ascon-XOF, the IV of these functions is the same as the hash function "key" (PK.seed || ADRS).. hence the 320-bit state of both "F" and "M(+)" computation matches up at that point of computation. Internal state word cancellation is prevented only because the length of (PK.seed || ADSR) is a multiple of 8, resulting in extra P12 invocation in padding and throwing the two computations off sync. If domain separation were used, the overall construction would behave like these were two actually independent functions.

Documentation: Use SLH-DSA

There were changes from SPHINCS+ 3.1 [4] to SLH-DSA [3] -- some of these came relatively late, and the Ascon-Sign specification does not adapt those. The Ascon-Sign algorithm description is incomplete, and some parts of it are arguably not as good as the SLH-DSA standard [3]. The spec has many technical issues, including expressions such as "s % (1 << z)! = 0" in the pseudocode (Alg 2.). Furthermore, Important parts such as the address (ADRS) structure or its encoding are not defined at all; the pseudocode is peppered with undefined functions such as ADRS.setTreeHeight().

Cheers,
-markku

References

[1] Vikas Srivastava, Naina Gupta, Arpan Jati, Anubhab Baksi, Jakub Breier, Anupam Chattopadhyay, Sumit Kumar Debnath, and Xiaolu Hou. "Ascon-Sign Submission to the NIST Post-quantum Project." June 2023. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/Ascon-sign-spec-web.pdf

[2] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. "ASCON v1.2: Lightweight Authenticated Encryption and Hashing." https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf (Also see: J. Cryptol (2021)34:33 https://doi.org/10.1007/s00145-021-09398-9)

[3] NIST. "Stateless Hash-Based Digital Signature Standard." FIPS 205 Initial Public Draft, August 2023. https://doi.org/10.6028/NIST.FIPS.205.ipd

[4] Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens,
Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, and Bas Westerbaan. "SPHINCS+ Submission to the NIST post-quantum project, v.3.1." June 2022. https://sphincs.org/data/sphincs+-r3.1-specification.pdf