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

From: pqc-forum@list.nist.gov on behalf of Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Sent: Tuesday, November 28, 2023 4:21 PM
To: pqc-forum
Cc: Markku-Juhani O. Saarinen
Subject: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

Hi Again,

First of all, I can see that I had some missing words in my previous comment. Apologies -- the result of some hasty editing.

Anyway, it is important to understand that even though the collision resistance of Ascon-XOF/Hash matches that of SHA2-256 and SHA3-256, its pre-image resistance is different; it also only 2**128 due to the small capacity of this lightweight sponge construction. To make it clear that the Ascon security bound of 128 bits is also the limit of the security of Ascon-Sign, I've included some simple cryptanalysis; how one can break the proposed "Category 3" Ascon-Sign-192{s, f} with 2**128 effort.

This illustrates that to switch a hash function in a hash-based signature scheme, one needs to understand the various security requirements/assumptions of that scheme towards hash functions and also the security properties of the particular hash function. That's why SHA2 and SHAKE are instantiated in quite different ways in FIPS 205 SLH-DSA. The instantiation for SHA2 is significantly more complex (with MGF1 and HMAC wrappers and a combination of SHA2-256 and SHA2-512 at Categories 3,5) due to its well-known functional and security limitations. Any mode of Ascon really has no hope of meeting those requirements at Category 3 due to its 320-bit state limitation.

FORGERY ATTACK ON ASCON-SIGN-192

We create a forgery by performing a pre-image attack on the message digest in the verification function; one valid signature query is required. This is a trivial time-memory tradeoff attack; think of the classical D-H 1977 Double-DES attack. Memory usage is large when presented like this, but this is commonly ignored in classical complexity.

0. Let "h" be the target hash from H_msg, or more precisely h = Ascon-XOF(R || PK.seed || PK.root || M, 8m). Here M, R are from any valid signature. The attack aims to find another message so that the same h is produced. Then, the same signature variables are also valid for that message. This is a forgery and a clear violation of EUF-CMA.

1. Let S_h be the 320-bit state of Ascon-XOF after absorbing (R || PK.seed || PK.root || M) but before moving to the squeezing phase. We know that from the S_h state, the exact same XOF/hash output h will be output.

2. We choose 2**128 initial message chunks X and store the 320-bit state S_x of Ascon after processing (R || PK.seed || PK.root || X).

3. Ascon -- like SHA-3 or any sponge hash -- can be efficiently computed in the reverse direction if the entire state and all inputs are known. We further choose 2**128 message chinks Z and calculate the hash in the inverse direction from state S_h. The resulting 320-bit inverse state is S_z.

4. Now, by the birthday paradox, there is a non-negligible probability that a 256-bit match between some state S_x and S_z exists in the "upper" state words s[1..4]. We choose a middle 64-bit word Y so that S_x xor Y = S_z (single absorption step to force s[0] to match). Now we have a second message M' = (X || Y || Z), that also satisfies h = Ascon-XOF(R ||

PK.seed || PK.root || M', 8m) = h.

[Sidenote: The fact that only a single hash is computed in verification is not apparent from the pseudocode spec due to various errors it has. Fortunately, the Ascon-Sign explicitly claims to work like SPHINCS+ v3.1. This message hash operation Hmsg is on line 6 of ASCON-Verify (Alg 21). However, the message M is also passed on line 17 of the verification function as an argument to FORS-PK-FROM-SIGN() -- this is not accurate and would not work -- both SPHINCS+ and the implementation pass the hash of the message instead. The pseudocode doesn't even use the returned value PK_FORS from line 17. The message M is also seemingly passed on line on line 19 to HT-VERIFY(), and from there via XMSS-SIGN() to WOTS-SIGN(), where the message is suddenly constant-length for the checksum computation. Again, this is an error in the specification.]

Cheers,
- Markku

On Tuesday, November 28, 2023 at 3:02:10 PM UTC Markku-Juhani O. Saarinen wrote:

> 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.

From: pqc-forum@list.nist.gov on behalf of Anubhab Baksi <anubhab91@gmail.com>
Sent: Saturday, December 9, 2023 4:16 AM
To: Markku-Juhani O. Saarinen
Cc: pqc-forum
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

Dear Markku,

Thank you for your kind interest in our research and for taking time to go through the submission package. We will carefully go through each of the points and communicate with you in each step.

Dear community,

We shall discuss with Markku in more detail and update the community about it. To avoid spamming, we will move our discussion with Markku in a private email thread for the time being.

Thanks and best regards,
Anubhab
(on behalf of Ascon-Sign team)

বুধ, ২৯ নভেম্বর, ২০২৩ ০৬:২০ তারিখে Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> লিখেছেন:

> Hi Again,
>
> First of all, I can see that I had some missing words in my previous comment. Apologies -- the result of some hasty editing.
>
> Anyway, it is important to understand that even though the collision resistance of Ascon-XOF/Hash matches that of SHA2-256 and SHA3-256, its pre-image resistance is different; it also only 2**128 due to the small capacity of this lightweight sponge construction. To make it clear that the Ascon security bound of 128 bits is also the limit of the security of Ascon-Sign, I've included some simple cryptanalysis; how one can break the proposed "Category 3" Ascon-Sign-192{s, f} with 2**128 effort.
>
> This illustrates that to switch a hash function in a hash-based signature scheme, one needs to understand the various security requirements/assumptions of that scheme towards hash functions and also the security properties of the particular hash function. That's why SHA2 and SHAKE are instantiated in quite different ways in FIPS 205 SLH-DSA. The instantiation for SHA2 is significantly more complex (with MGF1 and HMAC wrappers and a combination of SHA2-256 and SHA2-512 at Categories 3,5) due to its well-known functional and security limitations. Any mode of Ascon really has no hope of meeting those requirements at Category 3 due to its 320-bit state limitation.
>
> FORGERY ATTACK ON ASCON-SIGN-192
>
> We create a forgery by performing a pre-image attack on the message digest in the verification function; one valid signature query is required. This is a trivial time-memory tradeoff attack; think of the classical D-H 1977 Double-DES attack. Memory usage is large when presented like this, but this is commonly ignored in classical complexity.
>
> 0. Let "h" be the target hash from H_msg, or more precisely h = Ascon-XOF(R || PK.seed || PK.root || M, 8m). Here M, R are from any valid signature. The attack aims to find another message so that the same h is produced. Then, the same

From: pqc-forum@list.nist.gov on behalf of Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Sent: Monday, December 11, 2023 1:46 PM
To: pqc-forum
Cc: Anubhab Baksi; pqc-forum; Markku-Juhani O. Saarinen
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

Hi All,

I saw a posting from the Ascon-Sign team stating they would like to go through the details with me "to avoid spam." There has been no communication from them. Anyway, I think my cryptanalytic notes are getting in the way of some broader questions about the originality of the work:

- Ascon-Sign = SPHINCS+ v3.1 with Ascon plugged as the hash function. NIST should note that there is no intersection between the authors of Ascon-Sign and the authors of Ascon and / or SPHINCS+ v3.1. Perhaps the real authors of those algorithms would care to check the Ascon-Sign document and comment if they see any merit or originality in it.

- What I can say (as an occasional thesis supervisor and journal editor) is that the work would be automatically dismissed for plagiarism if submitted as scholarly work. Even "Turnitin" automated checks judge it as "high risk." There are direct copy & paste sections, etc -- it is basically unpublishable with this authorship.

- As often happens with such works, the resulting technical quality is poor. The on-ramp specification document has internal inconsistencies and does not properly match the implementation (which is, again, just copied from the SPHINCS+). It would be much better to dismiss it and use the actual specification documents for SLH-DSA.

And of course: That "instantiation" itself -- plugging Ascon into SPHINCS+. the only potential area of original contribution in Ascon-Sign is quite clearly botched due to a lack of proper security analysis.

I previously reported a simple 2**128 forgery attack against "192-bit" Ascon-Sign. I've now implemented a reduced version of this attack (e.g., those inverse Ascon permutations) to verify it. The implementation uses a simple 2-function stepping trick, making the pre-image matching memoryless with standard birthday/cycle finding methods. Note that the same type of attack doesn't work with Davies-Meyer mode hash constructions like SHA2.

Sincerely,
- markku

On Saturday, December 9, 2023 at 9:16:22 AM UTC Anubhab Baksi wrote:

> Dear Markku,
>
> Thank you for your kind interest in our research and for taking time to go through the submission package. We will carefully go through each of the points and communicate with you in each step.
>
> Dear community,
>
> We shall discuss with Markku in more detail and update the community about it. To avoid spamming, we will move our discussion with Markku in a private email thread for the time being.
>
> Thanks and best regards,

From: 'Scott Fluhrer (sfluhrer)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Monday, December 11, 2023 2:30 PM
To: Markku-Juhani O. Saarinen; pqc-forum
Cc: Anubhab Baksi; pqc-forum
Subject: RE: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

I just went through the ASCON-SIGN document, and I have to agree with Markku.

The ASCON-SIGN draft states (in the ASCON Security Claim) that Ascon-Hash has 128 bits of (second) pre-image resistance (just as Markku pointed out).

Given that we all appear to agree with that, I do not understand why they list the "Expected security level" for the 192s and 192f parameter sets as "3". They claim ("Security claim for Ascon-Sign") that

> These properties are derived from the characteristics of the ASCON hash functions used to instantiate those function families. Note that ASCON cipher suite is well analyzed, and therefore, Ascon-Sign is expected to have the same security strength as SPHINCS+.

However, ASCON-Hash does not have the same security characteristics as SHAKE-256, and so just blithely claiming that nothing has changed doesn't make it so.

Perhaps they do not believe that a second preimage attack directly leads to a forgery – however, it does. The easiest way is to take a known signature/message pair, and perform a second preimage attack to find another message that has the same final ASCON internal state during the initial message hash as the good message (and so the known signature would be a valid signature for this second message).

There are more complex ways to use internal collisions in the one-time signatures to come up with a partial second key that can be used to efficiently sign any message the attacker wants (without performing this 2^{128} work each time) – however, that doesn't matter – the existence of the obvious 2^{128} attack suffices.

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> **On Behalf Of** Markku-Juhani O. Saarinen
Sent: Monday, December 11, 2023 1:46 PM
To: pqc-forum <pqc-forum@list.nist.gov>
Cc: Anubhab Baksi <anubhab91@gmail.com>; pqc-forum <pqc-forum@list.nist.gov>; Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: Ascon-Sign

Hi All,

I saw a posting from the Ascon-Sign team stating they would like to go through the details with me "to avoid spam." There has been no communication from them. Anyway, I think my cryptanalytic notes are getting in the way of some broader questions about the originality of the work:

- Ascon-Sign = SPHINCS+ v3.1 with Ascon plugged as the hash function. NIST should note that there is no intersection between the authors of Ascon-Sign and the authors of Ascon and / or SPHINCS+ v3.1. Perhaps the real authors of those algorithms would care to check the Ascon-Sign document and comment if they see any merit or originality in it.

- What I can say (as an occasional thesis supervisor and journal editor) is that the work would be automatically dismissed for plagiarism if submitted as scholarly work. Even "Turnitin" automated checks judge it as "high risk." There are direct copy & paste sections, etc -- it is basically unpublishable with this authorship.