

---

**From:** Joost Rijneveld <joost@joostrijneveld.nl>  
**Sent:** Tuesday, March 13, 2018 10:03 AM  
**To:** pqc-comments  
**Cc:** pqc-forum@list.nist.gov  
**Subject:** OFFICIAL COMMENT: SPHINCS+

Dear all,

It was brought to our attention that the SPHINCS+ reference code contains an error that causes a longer-than-necessary message digest to be generated and signed (contradicting the specification). We have fixed the code and updated it on our web page. Note that, as this also affects the leaf index selection, the fix causes incompatibility with the previous version of the code.

Thanks go to Dorian Amiet for spotting the bug and alerting us to it.

Cheers,  
Joost Rijneveld, on behalf of the SPHINCS+ team

---

**From:** Thomas Prest <thomas.prest@ens.fr>  
**Sent:** Friday, April 13, 2018 4:28 PM  
**To:** pqc-comments  
**Cc:** pqc-forum@list.nist.gov  
**Subject:** OFFICIAL COMMENT: SPHINCS+

Dear all,

This mail is to inform you of a fault injection attack against SPHINCS+ and Gravity-SPHINCS.

In schemes of the SPHINCS family, the signing algorithm essentially consists of reconstructing a bunch of Merkle trees and signing their roots with one-time signatures (OTS).

Introducing one single random fault during the reconstruction of the penultimate Merkle tree has the effect that one OTS signs two different values, which can then be exploited to forge signatures for any message for a computational cost of computing about  $2^{34}$  hashes.

More detailed information can be found here:

- Article:

<https://eprint.iacr.org/2018/102>

- Slides: <https://tprest.github.io/Slides/grafting-trees-pqcrypto.pdf>

We have informed the submitters about this attack, and to the best of our knowledge they agree with the claims made in our article.

We want to emphasize that this is a fault injection attack: it does not question the security of these schemes when the signing algorithm is executed normally.

Best,

Laurent, Ange and Thomas

---

**From:** Christopher J Peikert <cpeikert@alum.mit.edu>  
**Sent:** Thursday, May 24, 2018 3:16 AM  
**To:** pqc-comments  
**Cc:** pqc-forum@list.nist.gov  
**Subject:** OFFICIAL COMMENT: SPHINCS+

This comment refers to the SPHINCS+ hash-based signature proposal at <https://sphincs.org/>  
(specification document at <https://sphincs.org/data/sphincs+-specification.pdf>).

Summary:

1. The specification claims that Theorem 9.1 supports the concrete security of the submission.
2. This claim is incorrect, because the theorem apparently does not apply to the proposed instantiations: it requires the component function  $F$  to have a structural property that it almost certainly does not have. (Indeed, it seems hard to find \*any\* suitable instantiation having the property.)
3. Using a known non-tight (but presumably applicable) security reduction yields much weaker security bounds, by roughly 60 (classical) bits or more.
4. I'm not aware of (nor have I looked for) any attack that violates the claimed security levels. This comment is solely about what known reductions imply (or don't) about the concrete security of the proposed SPHINCS+ instantiations.

Details:

Section 9 claims that "the security of SPHINCS+ is based on standard properties of the used function families" (and a random-oracle assumption), and says that "we give a security reduction for SPHINCS+ underpinning the above claim." Specifically, Theorem 9.1 states a concrete security bound for SPHINCS+, which is tightly related to the security of its components.

In Section 7.1.2 (Table 3), Theorem 9.1 is used to derive the proposed SPHINCS+ parameters, for classical bit-security levels of roughly 128, SPHINCS+ 196, 256 (and corresponding quantum ones).

Unfortunately, Theorem 9.1 apparently does not apply to the proposed SPHINCS+ instantiations -- in particular, the choices of  $F$  (Section 7.2) SPHINCS+ -- because it requires  $F$  to have the structural property given in Equation (9), p. 42:

For every key  $k$ , every value in the image of  $F_k : \{0,1\}^n \rightarrow \{0,1\}^n$  has at least two preimages.

Even for relatively small  $n$ , this property is highly unlikely to hold for even a single random function, much less a huge collection. So it almost certainly does not hold for the concrete  $F$  derived from SHA-256, SHAKE-256, and Haraka. Indeed,

it seems difficult to guarantee the property without sacrificing  $F$ 's practicality or second-preimage resistance (another requirement of Theorem 9.1).

The source of the reduction technique behind Theorem 9.1 is <https://eprint.iacr.org/2015/1256> . As explained in Section 5.1 (page 24) the property from Equation (9) is essential to the reduction's tightness:

"This additional requirement is needed to not [have] to use the one-wayness of  $F$ . If we had to use the one-wayness of  $F$ ... This would imply a security loss of roughly  $h$  bits."

According to this, such a loose reduction would yield security bounds that are about  $h \geq 60$  (classical) bits worse than the claimed security of the proposed SPHINCS+ parameters.

Sincerely yours in cryptography,  
Chris

---

**From:** A. Huelsing <ietf@huelsing.net>  
**Sent:** Tuesday, May 29, 2018 12:37 PM  
**To:** pqc-forum@list.nist.gov  
**Subject:** Re: [pqc-forum] OFFICIAL COMMENT: SPHINCS+

Dear Chris, dear list

thanks for your comment. You are absolutely right with your analysis.  
We made the same observation a few weeks ago. We currently have a scientific paper on SPHINCS+ in submission where we deal with this differently.

The paper should be online soonish (subject to vacation caused delay).

Best,

Andreas (for the SPHINCS+ team)