

Stateful Hash-Based Signatures

Feedback to NIST's Requests for Input – Issued: June 21, 2018

Table of Contents

Tim Hollebeek – DigiCert.....	2
Russ Housley – Vigilsec.....	2
Scott Fluhrer – Cisco.....	2
Donald Matthews – AMD.....	3
Robert Clifford – qubit report.....	5
David Hook – Bouncy Castle.....	6
David Hua.....	6
Dave Finlay – Everyone counts.....	6
Aline Gouget – Gemalto.....	7
Charles Sheehe – Glenn Research Center – NASA.....	7
Mike Powers – Leidos (AT&E labs).....	8
Quynh Dang – NIST.....	8
Mukesh Saini.....	8
Howard Haney.....	9
Rafael Misoczki – Intel.....	9
Oscar Garcia-Morchon – Phillips.....	9
Panos Kampanakis - Cisco.....	10
Mike Gardiner.....	10
Paul Hoffman – ICANN.....	10
Andrea Beatty – Delapcpa.....	11
Sandra Lambert – member of ANSI X9F4.....	11
Tim Hollebeek – DigiCert.....	11

Tim Hollebeek – Digicert

We think both should be standardized as soon as possible, and we don't think one should be held up for the other. The differences are small enough that there's no need to pick a "winner" and a "loser" as part of the standardization process.

We also think that in addition to code signing, NIST should standardize stateful hash-based signatures for TLS roots and intermediates, so they can start being deployed now. Roots can take 10+ years to reach ubiquity, and this has hindered many previous crypto transitions (due to the extreme popularity of Windows XP and XP embedded). However, progress on that front is currently blocked because there is no standard for hash-based signatures.

If there were a standard, forward-thinking companies could start planning for and deploying the technology they will need to upgrade to NIST PQC candidates in the future. But right now, they're waiting for NIST to act.

Russ Housley – Vigilsec

Dustin:

This paper describes the difference between the HSS/LMS and XMSS: <https://eprint.iacr.org/2017/349.pdf>

I suspect that you already know about the paper, but others may find it useful.

I am an advocate of the HSS/LMS hash-based algorithm, mostly because it is more straightforward. There is some speed improvement for the additional complexity in XMSS at the cost of a larger signature value. In my opinion, the speed improvement is not big enough to justify the larger signature size.

In addition, Dave McGrew and his co-authors were very careful to use techniques that are at least 20 years old in the HSS/LMS specification. This provides a great deal of confidence that there are no lurking patent concerns. On the other hand, XMSS depends on some techniques that were published as recently as 2016. I believe that the open source community will embrace the conservative approach taken in the HSS/LMS specification.

For these reasons, I would like to see NIST progress HSS/LMS and XMSS at the same time.

Russ

Scott Fluhrer – Cisco

Dustin:

This paper describes the difference between the HSS/LMS and XMSS: <https://eprint.iacr.org/2017/349.pdf>

I suspect that you already know about the paper, but others may find it useful.

I am an advocate of the HSS/LMS hash-based algorithm, mostly because it is more straightforward.

Thank you.

There is some speed improvement for the additional complexity in XMSS at the cost of a larger signature value.

I'm not sure if I understand this, though. For equivalent parameter sets, the signature sizes of HSS/LMS and XMSS are fairly close; XMSS tends to win on slightly deeper tree hierarchies, but that isn't large; see table 3 of the paper.

As for the speed, LMS/HSS is consistently several times (3x-5x) faster than XMSS for equivalent parameter sets; this is because most of the time for both is spent computing hashes, and XMSS uses 3-5 times as many hashes (actually, hash compression computations) compared to LMS/HSS.

One place where you might say that LMS is slower and has a smaller signature is that the currently supported parameter sets in LMS allows that as an option; it allows $W=256$ (I'm using the XMSS terminology for the Winternitz parameter), and this approximately halves the signature size over $W=16$, while slowing things down by a factor of 8.

However:

- This is an option; LMS supports $W=16$ parameter sets as well
- There is no reason why XMSS couldn't be extended to support $W=256$ parameter sets

In my opinion, the speed improvement is not big enough to justify the larger signature size.

In addition, Dave McGrew and his co-authors were very careful to use techniques that are at least 20 years old in the HSS/LMS specification. This provides a great deal of confidence that there are no lurking patent concerns. On the other hand, XMSS depends on some techniques that were published as recently as 2016. I believe that the open source community will embrace the conservative approach taken in the HSS/LMS specification.

For these reasons, I would like to see NIST progress HSS/LMS and XMSS at the same time.

Russ

Donald Matthews – AMD

NIST Computer Security Division,

Thank you for soliciting our feedback on the topic of stateful hash-based signature algorithm development.

AMD has been following the post quantum cryptography development and are very interested in solutions. As discussed at the NIST workshop in April, AMD has been following the IETF developments of XMSS and LMS with anticipation of using these standards once published. AMD is investigating the use of these stateful hash-based signature schemes for quantum resistance in our future projects. A major concern regarding the formalized publishing of these standards would take a considerable amount of

time to become official and published. Due to the longer timeframe for developing hardware components vs either software or firmware, the concern is that product intersection, with published algorithms, could potentially be a considerable distance in the future.

We do understand that the NIST standard may only be for limited use cases. AMD believes its use cases will be consistent with the NIST standard use cases.

As to NIST's questions, the answer depends on a few things. In general, we are on a short timeline for being able to include hash-based firmware authentication in our next product. Due to that, in our opinion, NIST should start working on a standard around XMSS. Below are a few questions that could alter the answer.

As for waiting for the LMS specification, it once again depends on some further questions about the two different algorithms. These questions are also below.

The final question about anyone currently using these standards, we do not know of anyone that is currently using stateful hash-based signatures, but AMD intends to move to them in a relatively short timeframe.

Below are the questions and comments that AMD has in relation to the process and the two algorithms.

Our first question has to do with the expected timeframe for NIST to publish a standard based on XMSS or LMS. What is the expected time for NIST from start, with people assigned to the task, to a finished published specification?

We understand that any standardization process will potentially encounter delays at the various steps of the process. Your email referenced an expectation that IETF will publish the LMS specification within a few months ("likely in the coming months"). Recommendation for discussion, is that the NIST standard process could be started before LMS is published even if there are plans to include an LMS option. Our concern in this scenario would be some added delay in the NIST standard process if it was decided to wait for LMS publication.

Between the two algorithms, LMS has better performance for verification, which is a critical function for AMD. Does XMSS have security properties that are much better than LMS to justify selecting the slower algorithm? Can you elaborate on these security properties?

With the advent of quantum computers, AMD understands that the current hash algorithms are still secure but at an effective reduced security strength. We also understand that the hash algorithms have a common computer security strength that is $\frac{1}{2}$ of their resulting hash size. Are there features that are a part of XMSS and LMS that increase the security strength of the signature beyond $\frac{1}{2}$ of the bits size of the hash algorithm? Does XMSS with SHA256 essentially provide > 128 bits of security strength (collision resistance)?

Frequently, NIST will alter submitted algorithms when a standard is created (Rijndael -> AES block sizes). Do you currently expect to alter the XMSS and LMS algorithms for the standard?

There are three areas that AMD would like to see addressed:

- 1) XMSS specifies that to implement the algorithm, then there must be implementations of multiple versions where each version differs in Merkle tree height. Will NIST allow for a single parameter version to be fielded?
- 2) Will NIST allow the height parameter to be non-fixed (XMSS fixes to 10, 16, 20) but a variable that an implementor chooses before fielding a product?
- 3) Will NIST define a 384-bit variant to any of these algorithms?
 - a. If 256-bit is only 128-bit secure and 512 is only 256-bit secure, then 384 may not make sense for many applications since most use 128 or 256 bit keys.

AMD appreciates the considerations, and we look forward to closely engaging and following the progress of the NIST standard. Please feel free to contact us with any additional questions that you may have, especially concerning what AMD is looking for in a solution that includes these algorithms. We would certainly appreciate any time available for setting up a meeting where we can discuss these questions and intersection with AMD products.

Thank you for your time,

Don Matthews

Tom Woller

G Zhuang

Robert Clifford – qubit report

Mr. Moody:

One cryptocurrency claims use of XMSS.

From: <https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fblocktribune.com%2Fnew-blockchain-claims-it-is-resistant-to-quantum-computing-attacks%2F&data=02%7C01%7Cdustin.moody%40nist.gov%7C47d3622d25524e3c37c908d5e018b284%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C1%7C636661319038351219&sdata=f6Vcq9JiwvnxLybtqTXzNaAYeyla2IHZB0AndYcT1Ng%3D&reserved=0>

"Unlike other blockchains and cryptocurrencies, QRL [Quantum Research Ledger - qrl.org] utilizes a type of hash-based signature scheme known as the Extended Merkle signature scheme, or XMSS. Unlike ECDSA, the cryptography standard favored by today's most popular blockchain networks, XMSS is resistant to a sufficiently powerful quantum computer running Shor's algorithm."

FWIW.

Robert Clifford
qubitreport.com

David Hook – Bouncy Castle

Hi,

The Bouncy Castle Cryptography APIs have had an implementation of RFC 8391(XMSS/XMSS^MT) for a couple of releases now. We do have people both using and reviewing the implementation as well. I can say this with confidence as I've just had to issue a CVE on the 1.59 implementation (I delayed replying to the original request on the PQC forum while we got the CVE sorted out) and we also had to extend our KeyStore implementation to handle the keys in 1.59.

We have just issued 1.60 at https://www.bouncycastle.org/latest_releases.html . We will ask for feedback on who is using XMSS as part of the release announcement. If we get any responses that would allow us to provide you with more specific feedback we will pass them on. From our perspective it wouldn't hurt to know either.

Any questions about our implementation, please let us know.

Regards,

David

David Hua

To my knowledge, this is an "Informational RFC". Any one can create an informational RFC and publish it in IETF as RFC. Informational RFC does not mean that it has been standardized by IETF. The similar scenario is that any one can publish a technical report in ArXiv or ePrint. One can interpret Information RFC as a technical report in ePrint or ArXiv.

thanks!

Hua

Dave Finlay – Everyone counts

I think you are selling short the RFC Informational Category in general, and egregiously so when it comes to the publication process. Even the IRTF fast track ([RFC 5743](#)) used for RFC 8391 resulted in a [3 year iteration cycle](#). Independent Submissions, such as the Draft [JSON Schema](#) RFP, go through an even more arduous process. Having never published, maybe my comprehension of the relative effort between an RFC and a publication on arVix is lacking here.

For further review:

RFC Publication Process: <https://www.rfc-editor.org/pubprocess/>

arXiv Publication Process: <https://arxiv.org/help/submit>

Many Informational RFCs also have a large impact, even though the IETF did not Standards Track them. [RFC 1591](#) comes to mind.

Apologies for tangential rant, just wanted to clarify a perceived misconception.

[Aline Gouget – Gemalto](#)

Dear NIST Computer Security Division,

This is a good news for us to know that NIST will coordinate with other standards organizations on stateful hash-based signatures like LMS and XMSS.

We take this opportunity to share with you some questions and suggestions about the range of signature applications you are considering for (standalone) stateful hash-based signatures.

1. Does code signing includes any type of patchability use-cases, e.g. update of the OS, of the application or of a crypto library?
2. Does the root PKI certificate for code signing application could be based on a standalone stateful hash-based signature?
3. Does a root certificate of a PKI could be based on a standalone state-full hash-based signature, regardless of the PKI use-case?
4. Do you consider that secure notification messages could be allowed as part of the crypto agility strategy, even if the secure notification messages are not “code”?

Another important point for us is to know whether you will allow different types of parameters for LMS and XMSS such that it will be possible for us to manage trade-offs between performances and number of expected signature during the lifetime of the secret key.

Best regards

Aline

[Charles Sheehe – Glenn Research Center – NASA](#)

Hi

I would like NIST to address the Stateful hash-based signatures.
Address specific use cases where they would be allowed, especially for government organizations.
And recommended parameters.

This is my opinion and does not reflect the position of my agency.

Thanks
Chuck

Mike Powers – Leidos (AT&E labs)

Hello,

In response to the following questions, I have these comments:

- **Is anybody aware of industry using stateful hash-based signatures at this time?** The only usages that I'm aware of in the industry at this time are in the context of crypto-currencies. I believe there are at least a few like "The Quantum Resistant Ledger (QRL)" which uses XMSS to sign transactions and "IOTA" which uses a Lamport signature to sign transactions.

Beyond that, I don't really have an opinion either way on whether NIST should start moving forward with standardization of XMSS.

Thanks,

Mike Powers

Quynh Dang – NIST

Right, LMS is a couple of times faster than XMSS and it also has a security proof based on the assumption that the compression function is a random oracle/ (pseudo)random function (

<https://eprint.iacr.org/2017/553.pdf>) .

The proof is for MD-hashes (SHA2s). I believe it should work on Keccak sponge-hashes as well because the output of the permutation is truncated to r bits therefore it works exactly as a compression function. Or, simply, a Keccak sponge-hash is a prf, so the proof should work.

Since LMS draft came out a lot earlier than XMSS and LMS is a lot faster than XMSS, many vendors already had code running or testing for LMS.

Maybe adopting both of them could be arguably a best approach.

Quynh.

Mukesh Saini

Dear team NIST,

My suggestion is that NIST should wait till RFC for LMS is completed. Lack of APIs functionality in XMSS will create implementation challenges.

Thanks & Regards

Commander Mukesh Saini (Retd)

Howard Haney

Would Strongly Recommend a little coordination with the ISO.

I am sure they would enjoy the opportunity to work with the USA on any thing that would enhance Security rather than our previous methods of words with no substance, and the result would be a Global Hash-Bashed Signature Security Standard led by our ISO Representative.

Worked for seven years with the Fed-CIO Council's Security committee as a DOD/DA Rep.

V/R HRH

Rafael Misoczki – Intel

Dear All,

We believe there is an industry wide benefit in accelerating NIST endorsement for Hash-Based Signatures (HBS).

Intel is researching/experimenting with HBS schemes and we would be inclined to bring sharper focus on schemes that do get formally endorsed by NIST.

Since XMSS is already a published RFC scheme, we see significant benefits in starting its NIST approval process now.

Best Regards,

Rafael Misoczki

Research Scientist

Intel Labs

Oscar Garcia-Morchon – Phillips

Hi Dustin,

I am replying to you only. Please, keep this information confidential to NIST.

We are working together with our Philips business to determine an internal transition strategy. An important aspect refers to digital signatures for code signing. Thus, we thank you for your email and for the question.

At this stage, we would not recommend NIST to work on XMSS only. LMS also seems to be very competitive, in particular, CPU wise. During the next months we will have a more clear opinion.

We have been asked for timelines to prepare a good transition strategy for Philips. While for KEM, PKE, generic signatures this timeline is relatively clear to us, we are missing some information:

- For the code-signing application, what timeline could we expect to have a NIST standard for quantum-resistant signatures applicable to code-signing?
- Today, AES128 is often used, e.g., for data encryption at rest. However, symmetric-algorithms such as AES will be less secure once a quantum-computer is available. Will NIST have a

recommendation to start using, e.g., AES256 instead of AES128, due to the quantum-threat? If so, when do you expect such a recommendation to come?

- Once a new NIST standard is in place (either for KEM, PKE, generic signatures, or special code-signing signatures), do you expect to have a timeline for deprecating classical algorithms? If yes, how long do you expect this period of time to be?

We fully understand that these questions are difficult to answer, but any feedback here is very welcome.

Kind regards, Oscar.

Panos Kampanakis - Cisco

Hi Dustin,

LDWM (the earlier version of the LMS draft) signatures are used in some Cisco chips for FPGA firmware signing. We also have heard interest in software package signing with stateful schemes.

We would like to see LMS and XMSS approved by NIST for some usecases. We tried to compare the two schemes for potential adopters in <https://eprint.iacr.org/2017/349> Personally, I would prefer for NIST to evaluate them together after they are both IETF RFCs. FWIW, there might be usecases of stateful schemes that have not been realized yet. For example in PKI, smaller size trees could be used as the Offline Root CA signing scheme given that the Root CA is offline and does not sign live. Such a usecase would assume different stateless schemes are used at the leaves of the cert chain of course.

Rgs,

Panos

Mike Gardiner

Hi Dustin,

I'm aware of some partners in progress with implementing LMS for PKI / Code Signing use but I cannot disclose the names.

Paul Hoffman – ICANN

On a process note, the IETF did not standardize XMSS, nor is it standardizing LMS. Informational RFCs are explicitly not standards in the IETF sense. They are publications that, in these cases, have been heavily vetted by a group associated with, but not part of, the IETF process, namely the CFRG, which is part of the IRTF.

Informational RFCs are often updated when errors are found or new operational considerations are brought up. It is much easier to update an RFC than, say, a NIST spec. The real question is what value having a parallel NIST specification for XMSS or LMS would be to the community. NIST could just publish documents that reference the RFCs and add NIST's point of view on some topics, add NIST identifiers, and so on.

--Paul Hoffman

[Andrea Beatty – Delapcpa](#)

Good Afternoon,

Please move forward with XMSS. It is necessary for the industry to move forward with the things that can be done now, while awaiting the NIST-Approved post QC algorithms. Thank you for the opportunity to provide feedback.

Regards,

Andrea

[Sandra Lambert – member of ANSI X9F4](#)

Hi Dustin,

Yes, NIST should start moving forward now with XMSS. I believe in the importance of standardizing XMSS signatures ASAP, so that people can start transitioning their infrastructures to quantum-safe technologies.

Standardization by NIST would a be very helpful step towards being able to create a quantum-safe PKI for the financial services industry.

Thanks for the opportunity for feedback,

Sandra Lambert

[Tim Hollebeek – DigiCert](#)

DigiCert would like to be able to offer stateful hash-based signatures to our customers, but the most common objection we get is that it has not yet been standardized by NIST. If NIST were to create a standard based on the excellent XMSS RFC from IETF, it would substantially accelerate the adoption of stateful hash-based signatures.

Such signatures are extremely useful, because the verification code is small and can be put into devices (for example, IoT) that are shipping today, and would allow those devices to be securely upgraded to the official NIST PQC algorithms when those become available

-Tim