From: Liu, Yi-Kai (Fed)
To: Dworkin, Morris J. (Fed); internal-pqc
Subject: Re: Revocation guidance for stateful HBS schemes
Date: Tuesday, December 10, 2019 5:14:28 PM

Fair enough. I've said my piece... I'll leave it to you to decide what is the most sensible thing to do.

--Yi-Kai

From: Dworkin, Morris J. (Fed) <morris.dworkin@nist.gov>
Sent: Tuesday, December 10, 2019 3:49 PM
To: internal-pqc
Subject: Re: Revocation guidance for stateful HBS schemes

Yi-Kai,

You're right, revocation is covered in a general way in the three parts of SP 800-57.

Like David said, I am open to including some text about revocation in SP 800-208, but I want us to take more time to work on it, without delaying the release of the draft for public comment.

MD

On 12/10/19, 1:17 PM, "Liu, Yi-Kai (Fed)" <yi-kai.liu@nist.gov> wrote:

Hi Morrie,

Thanks for considering this.

> In addition to the difficulty of what is best to say about revocation itself, there’s a question of scope, because I don’t think we have any significant requirement on revocation in FIPS 186 or elsewhere.

Is this really true? I thought that we do give guidance about key revocation, in SP800-57 Parts 1, 2 and 3. I just looked it up, and it seems like pretty good advice to me. It even talks about storing keys in cryptographic modules, making backups, etc.

I really don't see why we wouldn't want to mention this.

Cheers,

--Yi-Kai

From: Dworkin, Morris J. (Fed) <morris.dworkin@nist.gov>
Sent: Monday, December 9, 2019 4:58 PM
To: internal-pqc
Subject: Revocation guidance for stateful HBS schemes

Thanks, Yi-Kai, for raising this important issue.

On top of David, John, and Quynh’s comments, I consulted several people on the team, with the result that I prefer to post the draft for public comment without any new text about revocation. In addition to the difficulty of what is best to say about revocation itself, there’s a question of scope, because I don’t think we have any significant requirement on revocation in FIPS 186 or elsewhere. I would suggest that we discuss the issue further during the public comment period. In my opinion, if we can agree on useful, general guidance, we could include it when finalize the document, without giving an opportunity for public comment.

Morrie

From: "Dang, Quynh H. (Fed)" <quynh.dang@nist.gov>
Date: Wednesday, December 4, 2019 at 2:31 PM
To: "Liu, Yi-Kai (Fed)" <yi-kai.liu@nist.gov>, "Cooper, David A. (Fed)" <david.cooper@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Draft SP on Stateful Hash-Based Signature Schemes

Hi David and Yi-Kai,

Changing to a new public key, 1 stateful hash-based signature application would do just fine by signing the new public key with the current private key.

When knowing that the regular signing application has been compromised, using the other un-compromised stateful hash-based signature application to sign a revocation message seems to make sense. But, how to build a system which can detect when a OTS private key is re-used ? If a system has that functionality, then it should have the functionality to avoid the risk of key re-use.

If we say that when there is suspicion that the regular signing application has been compromised,.... then the next question would be that under what circumstances the suspicion should arise ?

Sure, in a carefully designed system, the signature receiver can generate some kind of an alert when receiving an irregular signed message such as arriving at a wrong time etc..., or the signer needs to generate a new signing key when there is some evidence that the private seed might have been stolen/compromised such as the signing module has been tamper-red with etc...

Quynh.

From: Liu, Yi-Kai (Fed) <yi-kai.liu@nist.gov>
Sent: Wednesday, December 4, 2019 1:16 PM
To: Cooper, David A. (Fed) <david.cooper@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: RE: Draft SP on Stateful Hash-Based Signature Schemes

Hi David,

I agree with the issues you're raising. But I think this is a situation where we shouldn't let the best be the enemy of the good. Obviously, the best solution is a stateless post-quantum signature. But if we're talking about sub-optimal solutions, we should at least help users make the best of it.

How to sign the revocation message? I see the same options that you do, including elliptic curve crypto, and hash-based signature with a backup key. I like the idea of having a separate backup key whose only use is to revoke the primary key. I still would not recommend any of these methods, but I think they are worth mentioning, because they could be useful in practice.

To put it another way, the current document is very focused on *preventing* key re-use. I think it would be helpful to add at least a sentence or two encouraging users to think about how to *react* to that bad event, in some way that makes sense for their particular application.

Finally, I think we ought to acknowledge that, no matter how much effort we put into hardware security modules, there will *always* be a danger of key re-use. So key revocation is always going to be relevant, even if we don't have a good way of doing it. So I think the words "key revocation" (or some more appropriate terminology) ought to appear somewhere in the document.

Cheers,

--Yi-Kai

From: David A. Cooper <david.cooper@nist.gov>
Sent: Wednesday, December 4, 2019 12:09 PM
To: Liu, Yi-Kai (Fed) <yi-kai.liu@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Draft SP on Stateful Hash-Based Signature Schemes

Hi Yi-Kai,

I'd be hesitant to make a last-minute change like this without knowing what we would suggest. I don't see the benefit of writing a sentence or two saying that having the ability of revoke keys is important without providing any suggestions on how it can be done. Of course, this is just a draft going out for public comment, so something could still be adding to the final document.

The problem with revocation is that one needs a secure way to distribute the revocation information. If the reason that the ability to revoke is especially important for stateful HBS is the possibility of one-time key reuse, then would it make sense to use a stateful HBS scheme to sign revocation information? If not, then how would it be done? In some way that avoids the use of asymmetric cryptography?

How could one switch to a backup signature mechanism? The backup mechanism can't be RSA or ECDSA, since they aren't post-quantum secure, and the whole point is to have a scheme that can be deployed now that is post-quantum secure. If deployed now, the backup mechanism couldn't be any other post-quantum secure signature scheme, since none are NIST approved. Perhaps the idea is that the post-quantum secure backup signature mechanism would be deployed in the future, sometime after the standardization process is complete. However, we say in our document that if one could do something like this, then one should stick with RSA or ECDSA for now and then deploy a stateless post-quantum secure digital signature scheme later, once one has been standardized.

On 12/4/19 11:42 AM, Liu, Yi-Kai (Fed) wrote:

Hey,

So I was thinking it over some more, and I was wondering, can we add a sentence or two about the importance of key revocation?

I think this would fit in nicely in sections 1.2 and 9.1, where we are talking about how to avoid key re-use… the logical next step is to think about what to do if the unthinkable happens and a key does get re-used.

I understand there is not an easy way to revoke/replace a key in many of these situations. However I think we should at least point people in that direction, because maybe there is something they can do, like switching over to a backup signature mechanism, or warning the user.

Cheers,

--Yi-Kai