From: Vamshidhar Muppidi <vamshi11us@outlook.com>
Sent: Tuesday, July 5, 2022 10:27 PM
To: pqc-comments
Cc: pqc-forum; Vamshidhar Reddy
Subject: Selected Algorithm 2022 OFFICIAL COMMENT: CRYSTALS-KYBER

Follow Up Flag: Follow up
Flag Status: Completed

Team,

I am writing this comment on behalf of "**CRYSTALS-KYBER"** to support this as most optimal one for next gen encryption model.

Tech environments are evolving every year, as more footprint of tech stack appears on cloud and on premises the data exchange and secrecy remains more important to government and businesses which need trust from people in securing data.

Existing encryption protocols should be fast deprecated, and standard should start at 512

Thanks for standarding these for consumption & providing an opportunity to comment

Regards,
Vamshi

From:	'John Mattsson' via pqc-forum <pqc-forum@list.nist.gov>
Sent:	Saturday, July 9, 2022 7:24 AM
To:	Peter Schwabe
Cc:	pqc-forum
Subject:	Re: [pqc-forum] OFFICIAL COMMENT: CRYSTALS-Kyber

Hi Peter,

Thanks for the quick answer. I see now that the design choice of using four different SHA-3 function because of domain separation is clearly explained in the Kyber specification. Sorry for missing that. I don't have a strong opinion. Maybe using four different SHA-3 functions is the best tradeoff. But requiring four different functions like (SHAKE128, SHAKE256, SHA2-256, SHA3-512) or (AES-256, SHA-256, SHA-512, SHAKE256) might not work for future hash functions like the "winner" of the NSIT LWC project. That problem could however be solved with explicit domain separation at a later time.

Cheers,
John

On 2022-07-09, 11:06, "Peter Schwabe" <peter@cryptojedi.org> wrote:

'John Mattsson' via pqc-forum <pqc-forum@list.nist.gov> wrote:

Dear John, dear all,

Just speaking for myself.

> NIST should carefully consider which versions of Kyber to standardize
> as well as the algorithm choices in the standardized versions of
> Kyber.
>
> - The Kyber specification makes use of no less then 4 different SHA-3
> functions (SHAKE128, SHAKE256, SHA2-256, SHA3-512). While the four
> SHA-3 function can be implemented with a single Keccak API
>
> SHAKE128(M,d) = Keccak[256](M||1111, d)
> SHAKE256(M,d) = Keccak[512](M||1111, d)
> SHA3-256(M) = Keccac[512](M||01, 256)
> SHA3-512(M) = Keccac[1024](M||01, 256)
>
> it looks a bit strange to use four different functions. The
> fixed-length SHA-3 hash functions (drop-in for SHA-2) have to our
> knowledge seen little or no practical use. Instead the variable-length
> SHAKE functions have seen significant practical use in implementations
> as well as in published and upcoming standards such as EdDSA (RFC
> 8032), XMSS (RFC 8391), LMS (NIST SP 800-208), CMS (RFC 8702),
> RSASSA-PSS and ECDSA (FIPS 186-5 (Draft), RFC 8692), COSE
> (draft-ietf-cose-hash-algs), EDHOC (draft-ietf-lake-edhoc), CPace
> (draft-irtf-cfrg-cpace), FROST (draft-irtf-cfrg-frost), OPRF