From: Dang, Quynh H. (Fed)
To: Cooper, David (Fed); internal-pqc; (b) (6) Dang, Thinh H. (Fed)
Subject: Re: FYI: KEM-only TLS
Date: Monday, June 1, 2020 6:34:31 AM

Hi Dave,

Thank you for pointing out the paper. I was not aware of the paper before.

Naturally, for a protocol, one can see that for a server if it has a status KEM key pair, then the public key can be used as its identity/authentication key, but this public key must be delivered to the client(s) in an authenticated manner.

In TLS, only the server is required to get authenticated. Client authentication with a digital signature is situational when the server requires/requests.

So, when public key authentication is required for the client without a digital signature algorithm, the client must have a static KEM key pair and the server must have the client's public key in an authenticated manner.

One can design a protocol which does ephemeral key exchanges. The static KEM key pair(s) is/are used for authentication and an ephemeral KEM key pair is generated for each fresh key exchange.

That was what I explained at our last meeting.

How to make sure that the static KEM public keys are authentic is an issue in real world. Currently, PKIs using digital signatures are used for that.

It would be not practical that the browser vendor takes all servers' static KEM public keys (in an authenticated way), then load them into the clients (in an authenticated way). This solution won't work.

So, digital signatures are absolutely required for the internet to function.

Quynh.

From: David A. Cooper <david.cooper@nist.gov>
Sent: Friday, May 29, 2020 3:37 PM
To: internal-pqc <internal-pqc@nist.gov>
Subject: FYI: KEM-only TLS

During this morning's meeting there was a question about implementing TLS without using signatures. I believe that Daniel Apon previously pointed out this paper, which came out a few weeks ago:

> Post-quantum TLS without handshake signatures
> (https://eprint.iacr.org/2020/534.pdf)

The proposal would involve a change to the TLS 1.3 protocol, so implementing it would be more difficult than would be adding a post-quantum signature algorithm. However, the paper seems to argue that using only KEMs (Kyber or NTRU) would result in faster connections and less computation than using Dilithium or Falcon for authentication.
So, even if SPHINCS+ were the only signature scheme we could standardize, it seems that TLS could move forward by avoiding the use of any signature algorithm in the handshake protocol. (Signatures would still be needed to sign certificates, certificate status information, and certificate transparency information.)
David