From: Kerman, Sara J. (Fed)
To: Moody, Dustin (Fed)
Subject: RE: update PQC
Date: Tuesday, April 11, 2017 10:38:44 AM

It 'twas a quick fix!

From: Moody, Dustin (Fed)
Sent: Tuesday, April 11, 2017 10:38 AM
To: Kerman, Sara J. (Fed) <sara.kerman@nist.gov>
Subject: RE: update PQC

Mucho gracias! That was fast!

From: Kerman, Sara J. (Fed)
Sent: Tuesday, April 11, 2017 10:37 AM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: RE: update PQC

It is updated on csrc http://csrc.nist.gov/groups/ST/post-quantum-crypto/faq.html#Q15

And should show up on the beta site soon.

Let me know if there is anything else.
Sara

From: Moody, Dustin (Fed)
Sent: Tuesday, April 11, 2017 10:13 AM
To: Kerman, Sara J. (Fed) <sara.kerman@nist.gov>
Subject: RE: update PQC

Yes. Only the first paragraph was modified.

From: Kerman, Sara J. (Fed)
Sent: Tuesday, April 11, 2017 10:13 AM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: RE: update PQC

Sure. So all the code at the end remained the same, right? In the new beta site, I remember having an issue getting that font to work so if I don't need to change any of that – that's good.

Sara

From: Moody, Dustin (Fed)
Sent: Tuesday, April 11, 2017 10:06 AM
To: Kerman, Sara J. (Fed) <sara.kerman@nist.gov>

Subject: update PQC

Sara,

We want to update the last PQC FAQ that we created (Question 15 dealing with randomness). The first paragraph was where we made the only changes. The answer we want is below. Can you update the webpage? Thanks,

Dustin

A15: The function randombytes() will be available to the submitters. This is a function from the SUPERCOP test environment and should be used to generate seed values for an algorithm. Randombytes should only be used to seed a NIST-approved DRBG.

For functional and timing tests a deterministic generator is used inside randombytes() to produce the seed values. If security testing is being done simply substitute calls to a true hardware RBG inside randombytes().

Function prototype for randombytes() is:

```
// The xlen parameter is in bytes
void randombytes(unsigned char *x,unsigned long long xlen)
```

The following demonstrate the use of the KAT and non-KAT versions of the functions to generate a key pair for encryption:

```
int crypto_encrypt_keypair_KAT(
            unsigned char *pk,
            unsigned char *sk,
            const unsigned char *randomness
        )

int crypto_encrypt_keypair(unsigned char *pk, unsigned char *sk)
{
    unsigned char pk[CRYPTO_PUBLICKEYBYTES];
    unsigned char sk[CRYPTO_SECRETKEYBYTES];
    unsigned char seed[CRYPTO_RANDOMBYTES];

    randombytes(seed, CRYPTO_RANDOMBYTES);
    crypto_encrypt_keypair_KAT(pk, sk, seed);
}
```