From: Kelsey, John M. (Fed)
To: Regenscheid, Andrew R. (Fed); Perlner, Ray A. (Fed); Chen, Lily (Fed); Bassham, Lawrence E. (Fed); Smith,Daniel C; Moody, Dustin (Fed); Petzoldt, Albrecht R. (IntlAssoc)
Subject: What we decided today
Date: Wednesday, May 3, 2017 5:44:58 PM

Everyone,

This is what I think we agreed on today:

a. For random number generation, we want the performance tests to use AES CTR DRBG exactly as it appears in 90A—each request for random bytes leads to a Generate() call, and thus costs three extra AES encryptions and one AES key scheduling.

b. We will explicitly allow the submitters to use a "seed expanding" algorithm to expand the RNG output to a larger number of bytes in some more convenient way.

c. We will define a seed-expanding algorithm based on AES CTR mode, and one based on KMAC. We could also provide one based on SHA2.

d. We will encourage submitters to allow flexibility in their seed-expanding algorithm, so that changes are possible.

e. We will provide some reference code for these things, and also will provide some code to Dan and the open quantum safe people as needed. We'll also ask Dan and the OQS people to make sure submissions have access to fast implementations of AES, SHA2, or SHA3 as needed.

The seed-expanding algorithms will have the following interface:

S = a 32 byte seed value
D = a 4-byte diversifier value (to "name" the stream)
L = a 32-bit unsigned integer value for the output length requested.

The simplest way to use this is to call:

X = expand_seed(S, D, L)

A more complicated way to call it is:

make_stream(&rs, S,D,L)

X1 = get_bytes(&rs, n1)
X2 = get_bytes(&rs, n2)
etc.

If you call X = expand_seed(S, D, L) and then call

```
make_stream(&rs, S, D, L)
x1 = get_bytes(&rs, L-K)
x2 = get_bytes(&rs, K)
```

x1 || x2 will be the same as X.

When we call make_stream(S,D,L), L is the maximum allowed output length.

Comments?

--John