Comment on Post-Quantum Cryptography Requirements and Evaluation Criteria

Derek Atkins <datkins@securerf.com>

Fri 9/16/2016 2:46 PM

To: pqc-comments <pqc-comments@nist.gov>;

To whom it may concern:

I have read through the proposed document and supporting materials and have two comments:

1 --

In section 2.C.1 (Implementations) Submitting an implementation solely on the Intel x64 processor ignores the vast and ever-growing population of smaller processors that make up the Internet of Things. Quantum-resistant solutions optimized for such a capable machine may not scale down to 8 or 16 bit microcontrollers. To that end we propose that you include such smaller devices (e.g. 16-bit MSP430, and 8-bit 8051 and/or AVR8) in your testing and evaluation.

2 --

The proposed testing API is extremely problematic. Specifically, it assumes that Keys and Signatures are a constant size. There are definitely real algorithms where this is not the case, and each keypair (and signature) generated requires dynamic memory. In order to apply these variable-length algorithms to the process would require a change to the testing API that allows for dynamic sizes.

We see two possibilities to handle this extremely important use case:

1. Set the sizes so high as to be sure to include even the largest possible keys/signatures. The problem is that this would necessarily increase the amount of memory/storage required, and it's still potentially possible to hit a sample that goes beyond the boundaries, in which case the system either has to try again or give up.
2. Fix the APIs themselves to handle dynamic-size responses. This would allow an algorithm to return data objects of varying lengths.

We would encourage taking approach #2.

To this end we would propose a change to the API that enables dynamic responses, perhaps something like the following (with similar changes for the KAT versions):

```
typedef struct {unsigned long long len; unsigned char* buf;}buffer_t;

typedef buffer_t PublicKey;

typedef buffer_t PrivateKey;

typedef buffer_t Signature;


int crypto_sign_keypair_dyn(

PublicKey* pk,

PrivateKey* sk

);
```

```
int crypto_sign_dyn(
Signature *sig,
const unsigned char *m, unsigned long long mlen,
const PrivateKey sk
);

int crypto_sign_open_dyn(
const unsigned char *m, unsigned long long mlen,
const Signature sig,
const PublicKey pk
);

void free_buffer(buffer_t buf);
```

Thanks for your consideration,

-derek
--
Derek Atkins
Chief Technology Officer
SecureRF Corporation

Office: 203.227.3151 x1343
Direct: 617.623.3745
Mobile: 617.290.5355
Email: DAtkins@SecureRF.com

This email message may contain confidential, proprietary and / or legally privileged information and intended only for the use of the intended recipient(s) and others specifically authorized. Any disclosure, dissemination, copying, distribution or use of the information contained in this email message, including any attachments, to or by anyone other than the intended recipient is strictly prohibited. If you received this in error, please immediately advise the sender by reply email or at the telephone number above, and then delete, shred, or otherwise dispose of this message.