The NIST PQC Standardization Process

Dustin Moody
National Institute of Standards and Technology (NIST)
USA

The Long Road to Standardization

- 2012 – NIST begins PQC project
- 2015 – 1st NIST PQC workshop
- Feb 2016 – NIST Report on PQC
- Feb 2016 – NIST preliminary announcement of standardization plan
- Aug 2016 – Draft submission requirements and evaluation criteria
- Sep 2016 – Comment period ends

- What have we observed in the first mile?

Overview of draft NIST Call For Proposals

- Requirements for Submission Packages
  - Cover sheet, supporting documentation, implementations, IP statements
- Minimal Acceptability Requirements
  - Scope – public key signatures, encryption, key-exchange
  - Basic requirements for each function
- Evaluation Criteria
  - Security: security models, target security strengths,
  - Performance: key sizes, computational efficiency
  - Flexibility
- Plans for the Evaluation Process

Complexities of PQC standardization

- Much broader scope – three crypto primitives
- Both classical and quantum attacks
- Both a theoretical and practical aspect to assess security
- Multiple tradeoff factors (security, key size, signature size, ciphertext expansion, speed, etc.)
- Migrations into new and existing applications
- Many aspects which we haven’t handled in previous standards
- Not exactly a competition

Scope

- Signatures
  - Public-key signature schemes for generating and verifying digital signatures (FIPS 186-4)
- Encryption/key-establishment
  - Encryption scheme used for
    - Key transport from one party to another
    - Exchanging encrypted secret values between two parties to establish shared secret value (see SP 800-56B)
  - Key-establishment
    - Schemes like Diffie-Hellman key exchange (see SP 800-56A)

Security Notions

- Signatures
  - Existentially unforgeable with respect to adaptive chosen message attack (EUF-CMA)
  - Assume the attacker has access to no more than 2^{64} signatures for chosen messages
- Encryption
  - Semantically secure with respect to adaptive chosen ciphertext attack (IND-CCA2)
  - Assume the attacker has access to no more than 2^{64} decryptions for chosen ciphertexts
- These definitions specify security against attacks which use classical (not quantum) queries

Target Security Strengths

	Classical Security	Quantum Security	Examples
I	128 bits	64 bits	AES128 (brute force key search)
II	128 bits	80 bits	SHA256/SHA-3 256 (collision)
III	192 bits	96 bits	AES192 (brute force key search)
IV	192 bits	128 bits	SHA384/SHA-3 384(collision)
V	256 bits	128 bits	AES256 (brute force key search)

- For standardization, need to specify concrete parameters with security estimates
- No clear consensus on best way to measure quantum attacks

Other Properties

- Drop-in replacements
  - Need to consider key sizes, ciphertext/signature size, key generation time, auxiliary functions (hash functions, KDFs, RNGs,…), etc.
  - For some PQC primitives, special features might have security or performance issues, e.g.
    - Public-key reuse – for some primitives public-key reuse can be security problem
    - Decryption failures – some algorithms produce occasional decryption failures
- Perfect forward secrecy
- Resistance to side-channel attacks
- Compatibility with existing protocols and networks
- Simplicity and flexibility

Transition and Migration

- NIST will update guidance when PQC standards are available
  - SP 800-57 Part I specifies "classical" security strength levels 128, 192, and 256 bits are acceptable through 2030

- Even with the upcoming PQC transition, still required to move away from weak algorithms/key sizes:
  - Anything with "classical" security strength less than 112 bits should NOT be used anymore

Initial Actions

- Hybrid modes have been proposed as a transition/migration strategy to PQC crypto
  - Current FIPS 140 validation will only validate the approved component
  - NIST PQC standardization is focused on the PQC component
  - Hybrid modes would be interim stage in the transition

- Stateful hash-based signatures
  - IETF is taking action in specifying stateful hash-based signatures
  - NIST will coordinate with the IETF and possibly other standards organizations
  - NIST may consider stateful hash-based signatures as early adoption candidates for standardization, but only for specific applications like code signing

Summary of Comments Received

- 26 comments submitted
  - Clarifications in the text of the Call For Proposals
  - Require constant-time implementations?
  - More implementation platforms
  - Intellectual Property requirements
  - Decryption failure threshold
  - Public-key encryption and key-exchange issues
  - Quantum security and target security levels
  - API suggestions

First set of comments

- Require constant-time implementations?
  - Encourage, but not require

- More implementation platforms
  - Encourage, but not require

- Intellectual Property requirements
  - Keep mostly the same
  - We strongly prefer royalty-free algorithms, as they lead to more widespread adoption

- Decryption failure threshold
  - No hard bound – any failure rate that would violate security models

Key-establishment comments

- Several comments
  - Our request was too vague or too narrowly defined
- We continue to ask for public-key encryption
- In place of key-exchange, we are asking for Key Encapsulation Mechanisms (KEMs)
- KEMs have three algorithms:
  - Key generation – generates public and private key pairs
  - Encapsulation – uses public key to generate ciphertext and shared secret
  - Decapsulation – uses private key and encapsulation ciphertext to recover shared secret

KEMs

- KEMs and public-key encryption can generally be converted back and forth
- Still requiring IND-CCA2 security
- As a result of comments, we are adding another option:
  - Purely ephemeral key-exchange protocol can be done so that only passive security is required
  - NIST will consider encryption or KEM scheme which provides semantic security with respect to chosen plaintext attack (IND-CPA security)
- Diffie-Hellman type schemes can be submitted as KEMs
- Authenticated key-exchange is out of scope, as it is a protocol, not a primitive

KEM API

```
#define CRYPTO_SECRETKEYBYTES 192
#define CRYPTO_PUBLICKEYBYTES 64
#define CRYPTO_BYTES 64
#define CRYPTO_CIPHERTEXTBYTES 128
#define CRYPTO_RANDOMBYTES 64

int crypto_kem_keygenerate(
        unsigned char *pk,
        unsigned char *sk)
```

```
int crypto_kem_encapsulate(
        const unsigned char *pk,
        unsigned char *ct,
        unsigned char *ss)

int crypto_kem_decapsulate(
        const unsigned char *ct,
        const unsigned char *sk,
        unsigned char *ss)
```

Target Security Strengths Comments

- Comments on definition of security strength in terms of the cost of breaking various symmetric crypto primitives
- Comments questioning NIST’s overall approach to how to define quantum security
- Questions on whether parameters needed for all 5 levels
- Questions on specific amounts of classical or quantum security required
  - Concern that cannot tune classical and quantum parameters separately
- Some suggestions to not use target security levels

A New Approach to Quantum Security

- Not use single number of “bits of security” to define security strength
- Continue to categorize submissions into 5 rough security strength categories
  - Allows for more meaningful performance comparisons
  - Helps us make decisions on transition to longer keys

	Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

Quantum Security Strength Categories

	Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

- Computational resources should be measured using a variety of metrics
  - Number of classical elementary operations, quantum circuit size, etc…
  - Should consider realistic limitations on circuit depth (e.g 2^{40} to 2^{80} logical gates)
  - May also consider expected relative cost of quantum and classical gates.
- Submitters need not provide parameters for all 5 categories
- These are understood to be preliminary estimates

Hypothetical Scenario

- Assume a PQC algorithm has only one tunable parameter, corresponding to classical security
- Assume no quantum attacks, beside generic ones (i.e. Grover-based ones)
- To meet security strengths 1, 3, 5 set classical security to 128, 192, 256 bits respectively
- Security strength 2 means somewhere between 128 and 192 bits of classical security. Where exactly depends on how well the classical attacks "Groverize"
  - i.e., how effective are generic techniques for decreasing the cost of the classical attacks using quantum computers.

Classical Security Analysis - Required

- Classical computers are not going away
- Very possible classical attacks will be cheapest in practice, especially for algorithms not subject to dramatic quantum attacks
- Grover’s algorithm doesn’t parallelize well
- Science for assessing classical security is better developed than that for assessing quantum security
- Classical cryptanalysis can improve our understanding of the structure underlying the primitive, which is also the basis for quantum cryptanalysis

What Lies Ahead?

- Final submission requirements and evaluation criteria will be published soon
- PQC schemes can be submitted up to November 30, 2017
- Submission requirements:
  - Complete specification with concrete parameters
  - Performance analysis (implementations + documentation)
  - Known Answer Test values
  - Security analysis (with preliminary security strength categories)
  - Signed Intellectual Property statements and disclosures

What Lies Ahead?

- Minimal acceptability requirements
  - Publicly disclosed and available for public review
  - Not incorporate components insecure against quantum computers
  - Provide at least one of functionalities:
    - Public-key encryption, KEM scheme, Digital signatures
  - Concrete values for parameters claiming to meet security properties

  - See www.nist.gov/pqcrypto for complete details
  - Submission requirements will not change
  - NIST reserves the right to change evaluation criteria based on developments in the field

Evaluation Criteria

1. Security
   1. Security provided in important applications, such as TLS, IKE, etc.
   2. Meet security definitions (IND-CCA2, IND-CPA, EUF-CMA)
   3. Security strength categories and maturity of analysis
   4. Additional security properties (perfect forward secrecy, side-channel resistance, misuse-resistance, …)
2. Cost
   1. Public/private key, ciphertext, signature size
   2. Computational efficiency of public/private key operations, as well as key generation
3. Algorithm Characteristics
   1. Flexibility (additional functionalities, parameters scale easily, implementable on wide variety of platforms, parallelization, incorporation into existing applications and protocols)
   2. Simplicity
   3. Adoption (any factors hindering adoption?)

The Evaluation Process (3-5 years)

- NIST will post “complete and proper” submissions
- NIST PQC Standardization Conference (with PQCrypto, Apr 2018)
- Initial phase of evaluation (12-18 months)
  - Internal and public review
  - No modifications allowed
- Narrowed pool will undergo second round (12-18 months)
  - Second conference to be held
  - Minor changes allowed
- Possible third round of evaluation, if needed
- NIST will report, which may select algorithms for standardization

Summary

- Post-quantum cryptography standardization is going to be a long journey
- After the first mile, we observed many complexities and challenges
- NIST acknowledges all the feedback received, which has improved the submission requirements and evaluation criteria
- We will continue to work with the community towards PQC standardization
- Send comments to:

  pqc-comments@nist.gov

- See also: www.nist.gov/pqcrypto
  - Sign up for the pqc-forum for announcements and discussion