Update on the NIST Post-Quantum Cryptography Project

Dustin Moody
National Institute of Standards and Technology (NIST)
Classical vs Quantum Computers

• The security of crypto relies on intractability of certain problems to modern computers
  • Example: RSA and factoring

• Quantum computers
  • Exploit quantum mechanics to process information
  • Use quantum bits = “qubits” instead of 0’s and 1’s
  • Superposition – ability of quantum system to be in multiples states at the same time
  • Potential to vastly increase computational power beyond classical computing limit
The Sky is Falling?

• If a large-scale quantum computer could be built then....

• Public key crypto:
  • RSA
  • ECDSA (and Elliptic Curve Cryptography)
  • DSA (and Finite Field Cryptography)
  • Diffie-Hellman key exchange

• Symmetric key crypto:
  • AES
  • Triple DES

• Hash functions:
  • SHA-2 and SHA-3
The Sky is Falling?

• If a large-scale quantum computer could be built then....

• Public key crypto:
  • RSA
  • ECDSA (and Elliptic Curve Cryptography)
  • DSA (and Finite Field Cryptography)
  • Diffie-Hellman key exchange

• Symmetric key crypto:
  • AES
  • Triple DES

• Hash functions:
  • SHA-2 and SHA-3
The Sky is Falling?

• If a large-scale quantum computer could be built then....

• Public key crypto:
  • RSA
  • ECDSA (and Elliptic Curve Cryptography)
  • DSA (and Finite Field Cryptography)
  • Diffie-Hellman key exchange

• Symmetric key crypto:
  • AES
  • Triple DES

• Hash functions:
  • SHA-2 and SHA-3

• Vulnerable NIST standards
  • FIPS 186, *Digital Signature Standard*
    • Digital Signatures: RSA, DSA, ECDSA
  • SP 800-56A/B, *Recommendation for Pair-Wise Key Establishment Schemes*
    • Discrete Logs: Diffie-Hellman, MQV
    • Factorization based: RSA key transport
The Sky is Falling?

• If a large-scale quantum computer could be built then....

• Public key crypto:
  • RSA
  • ECDSA (and Elliptic Curve Cryptography)
  • DSA (and Finite Field Cryptography)
  • Diffie-Hellman key exchange

• Symmetric key crypto:
  • AES
  • Triple DES

• Hash functions:
  • SHA-2 and SHA-3

• Vulnerable NIST standards
  • FIPS 186, Digital Signature Standard
    • Digital Signatures: RSA, DSA, ECDSA
  • SP 800-56A/B, Recommendation for Pair-Wise Key Establishment Schemes
    • Discrete Logs: Diffie-Hellman, MQV
    • Factorization based: RSA key transport

Need longer keys
Use longer output
How soon do we need to worry?

• Potentially **as early as 15 years** to break RSA-2048
  • 15 years, $1 billion USD, small nuclear power plant (Mariantoni, 2014)
  • 50% chance (Michele Mosca)

• **PQC needs time** to be ready for applications
  • Confidence – cryptanalysis
  • Implementations
  • Usability and interoperability (IKE, TLS, etc. ... use public key crypto)
  • Standardization

• Transition has to be **soon enough** that any data compromised by quantum computers is no longer sensitive when compromise occurs
Possible Replacements

- Lattice-based
- Code-based
- Multivariate
- Others
  - Hash-based signatures
  - Isogeny-based signatures
  - Etc....

- All have their pros and cons
Initial Observations

• For most of the potential PQC replacements, the times needed for encryption, decryption, signing, verification are acceptable

• Some key sizes are significantly increased
  • For most protocols, if the public keys do not need to be exchanged, it may not be a problem

• Some ciphertext and signature sizes are not quite plausible

• Key pair generation time for the encryption schemes is not bad at all

• No easy “drop-in” replacements

• Would be nice to have more benchmarks
Gathering Steam

• PQCrypto Workshop series
• ETSI workshops
• European PQCrypto project, Quantum flagship
• Japan’s SAFECRYPTO project
• IETF hash-based signatures
• ISO/IEC JTC 1 SC 27 – study period on PQC
• Fall 2015: NSA announced it would be transitioning in the “not too distant” future [https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm]
The NIST PQC Project  http://www.nist.gov/pqcrypto

• Biweekly seminars since 2012
• Guest researchers and invited speakers
• Research: publications and presentations
  • PQCrypto, AWACS, ICICS, CRYPTO, Qcrypt, Eurocrypt, ETSI Quantum-safe workshops, etc.
• Out Reach
  • PKI community, Automotive industry talks

Collaboration

- IETF – CFRG
- ISO/IEC JTC 1 SC 27
- ETSI
  - Workshops, white papers
- Universities
  - University of Maryland (QuiCS)
  - University of Waterloo (Cryptoworks 21)
- Guest Researchers and Speaker
  - Lyubachevsky, Ding, Takagi, Petzoldt, Faugere, Gligoroski, Perret, etc...
Timeline

• June 2016 – Draft Call For Proposals released for public comment
• Fall 2016 – formal Call For Proposals finalized
• Nov 2017 – Deadline for submissions
• 3-5 years – Analysis phase
  • NIST will report its findings
• 2 years later - Draft standards ready (2023-2025)

• Workshops
  • Early 2018 – submitter’s presentations
  • One or two during the analysis phase
Call for Proposals

• NIST is calling for quantum-resistant cryptographic algorithms for new public-key crypto standards
  • Digital signatures
  • Encryption/key-establishment

• We see our role as managing a process of achieving community consensus in a transparent and timely manner

• We do not expect to “pick a winner”
  • Ideally, several algorithms will emerge as ‘good choices’

• We may pick one (or more) for standardization
  • Only algorithms publicly submitted considered
Differences with AES/SHA-3 competitions

- Post-quantum cryptography is more complicated than AES or SHA-3
  - No silver bullet - each candidate has some disadvantage
  - Not enough research on quantum algorithms to ensure confidence for some schemes

- We do not expect to “pick a winner”
  - Ideally, several algorithms will emerge as “good choices”

- We may narrow our focus at some point
  - This does not mean algorithms are “out”

- Requirements/timeline could potentially change based on developments in the field
Minimal acceptability requirements

- Publicly disclosed and available with no IPR
  - Signed statements, disclose patent info
- Implementable in wide range of platforms
- Provides at least one of: signature, encryption, or key exchange
- Theoretical and empirical evidence providing justification for security claims
- Concrete values for parameters meeting target security levels
Specification

• Implementation
  • Reference version
  • Optimized version

• Cryptographic API will be provided
  • Can call approved hash functions, block ciphers, modes, etc...

• Known Answer tests

• Optional – constant time implementation
Evaluation criteria

• To be detailed in the formal Call
  • Security
  • Cost (computational and memory)
  • Algorithm and implementation characteristics

• Draft criteria will be open for public comment

• We strongly encourage public evaluation and publication of results concerning submissions

• NIST will summarize the evaluation results and report publicly
Security Analysis

- Security definitions
  - IND-CCA2 for encryption, EUF-CMA for signatures, CK best for key exchange?
  - Used to judge whether an attack is relevant

- Quantum/classical algorithm complexity
  - Stability of best known attack complexity
  - Precise security claim against quantum computation
  - Parallelism?

- Security proofs (not required but considered as support material)

- Quality and quantity of prior cryptanalysis
# Target Security Levels

<table>
<thead>
<tr>
<th></th>
<th>Classical Security</th>
<th>Quantum Security</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>128 bits</td>
<td>64 bits</td>
<td>AES128 (brute force key search)</td>
</tr>
<tr>
<td>II</td>
<td>128 bits</td>
<td>80 bits</td>
<td>SHA256/SHA3-256 (collision)</td>
</tr>
<tr>
<td>III</td>
<td>192 bits</td>
<td>96 bits</td>
<td>AES192 (brute force key search)</td>
</tr>
<tr>
<td>IV</td>
<td>192 bits</td>
<td>128 bits</td>
<td>SHA384/SHA3-384 (collision)</td>
</tr>
<tr>
<td>V</td>
<td>256 bits</td>
<td>128 bits</td>
<td>AES256 (brute force key search)</td>
</tr>
</tbody>
</table>
Cost

• Computational efficiency
  • Hardware and software
    • Key generation
    • Encryption/Decryption
    • Signing/Verification
    • Key exchange

• Memory requirements
  • Concrete parameter sets and key sizes for target security levels
  • Ciphertext/signature size

• May need more than one algorithm for each function to accommodate different application environments
Algorithm and Implementation Characteristics

• Ease of implementation
  • Tunable parameters
  • Implementable on wide variety of platforms and applications
  • Parallelizable
  • Resistance to side-channel attacks

• Ease of use
  • How does it fit in existing protocols (such as TLS or IKE)
  • Misuse resistance

• Simplicity
The Evaluation Process

- Initial evaluation phase (12-18 months)
  - No tweaks/modifications allowed
  - Workshops at beginning and end of initial evaluation phase
- Report findings and narrow candidate pool
- Second evaluation phase (12-18 months)
  - Small modifications allowed
  - Workshop towards end of second phase
- Report findings and narrow candidates
- Select algorithms for standardization or decide more evaluation needed
Call for Feedback

• How is the timeline?
  • Do we need an ongoing process, or is one time enough?

• How to determine if a candidate is mature enough for standardization?
  • hash-based signatures for code signing

• We are focusing on signatures and encryption/key-establishment. Should we also consider other functionalities?

• How can we encourage people to study practical impacts on the existing protocols?
  • For example, key sizes may be too big
Conclusion

• NIST is calling for quantum-resistant algorithms
  • We see our role as managing a process of achieving community consensus in a transparent and timely manner
  • Different from (but similar to) AES/SHA-3 competitions

• PQC Standardization is going to be a long journey

• We don’t have all the answers

• Be prepared to transition to new (public-key) algorithms in 10 years
  • The transition will not be painless
    • NIST will provide transition guideline when PQC standards are developed
  • Prepare the application designers
    • Focus on crypto-agility