The Beginning of the End: The First NIST PQC Standards

Dustin Moody
Post-Quantum Cryptography Team

National Institute of Standards and Technology
U.S. Department of Commerce

Rewind 6 years ago…

Feb 2016 – PQCrypto conference – Fukuoka, Japan

Back to today

The Beginning of the End:

- The 3rd Round of the PQC "competition" is ending

Outline:

- What is the NIST PQC standardization process?
- What's happened during the last 6 years
- When will NIST announce? And what to expect
- What the future holds

Motivation

- 1994 – Shor's algorithm
  - a quantum algorithm giving an exponential speed-up over classical computers
    - Factoring large integers
    - Finding discrete logarithms
- 1996 – Grover's algorithm
  - polynomial speed-up in unstructured search, from O(N) to O($\sqrt{N}$)

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

Abstract

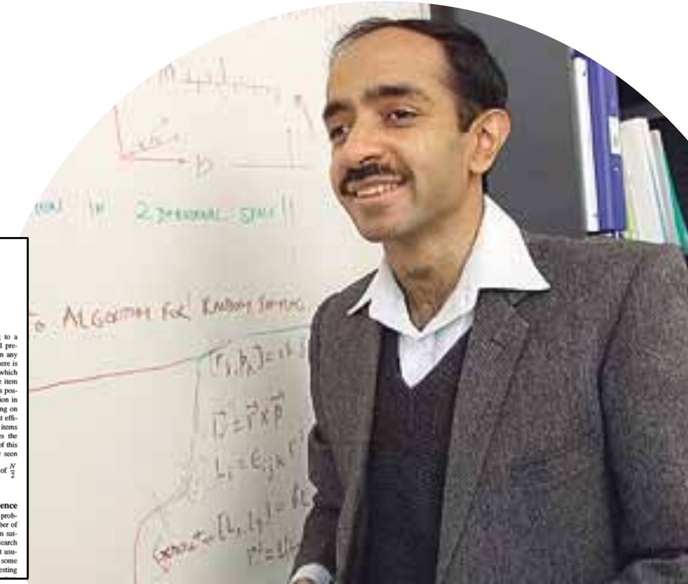
A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their computational properties. This paper gives Las Vegas algorithms for finding discrete logarithms and factoring integers on a quantum computer that take a number of steps which is polynomial in the input size, e.g., the number of digits of the integer to be factored. These two problems are generally considered hard on a classical computer and have been used as the basis of several proposed cryptosystems. (We thus give the first examples of quantum cryptanalysis.)

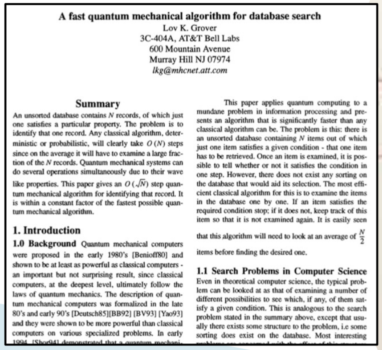
1 Introduction

Since the discovery of quantum mechanics, people have found the behavior of the laws of probability in quantum mechanics counterintuitive. Because of this behavior, quantum mechanical phenomena behave quite differently than the phenomena of classical physics that we are used

[1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum computation relates to classical complexity classes. We will thus first give a brief intuitive discussion of complexity classes for those readers who do not have this background. There are generally two resources which limit the ability of computers to solve large problems: time and space (i.e., memory). The field of analysis of algorithms considers the asymptotic demands that algorithms make for these resources as a function of the problem size. Theoretical computer scientists generally classify algorithms as efficient when the number of steps of the algorithms grows as a polynomial in the size of the input. The class of problems which can be solved by efficient algorithms is known as P. This classification has several nice properties. For one thing, it does a reasonable job of reflecting the performance of algorithms in practice (although an algorithm whose running time is the tenth power of the input size, say, is not truly efficient). For another, this classification is nice theoretically, as different reasonable machine models





A fast quantum mechanical algorithm for database search

Lov K. Grover
3C-404A, AT&T Bell Labs
600 Mountain Avenue
Murray Hill NJ 07974
lkg@mhcnet.att.com

Summary

An unsorted database contains N records, of which just one satisfies a particular property. The problem is to identify that one record. Any classical algorithm, deterministic or probabilistic, will clearly take $O(N)$ steps since on the average it will have to examine a large fraction of the N records. Quantum mechanical systems can do several operations simultaneously due to their wave like properties. This paper gives an $O(\sqrt{N})$ step quantum mechanical algorithm for identifying that record. It is within a constant factor of the fastest possible quantum mechanical algorithm.

1. Introduction

1.0 Background Quantum mechanical computers were proposed in the early 1980's [Benioff80] and shown to be at least as powerful as classical computers - an important but not surprising result, since classical computers, at the deepest level, ultimately follow the laws of quantum mechanics. The description of quantum mechanical computers was formalized in the late 80's and early 90's [Deutsch85][BB92] [BV93] [Yao93] and they were shown to be more powerful than classical computers on various specialized problems. In early

This paper applies quantum computing to a mundane problem in information processing and presents an algorithm that is significantly faster than any classical algorithm can be. The problem is this: there is an unsorted database containing N items out of which just one item satisfies a given condition - that one item has to be retrieved. Once an item is examined, it is possible to tell whether or not it satisfies the condition in one step. However, there does not exist any sorting on the database that would aid its selection. The most efficient classical algorithm for this is to examine the items in the database one by one. If an item satisfies the required condition stop; if it does not, keep track of this item so that it is not examined again. It is easily seen that this algorithm will need to look at an average of $\frac{N}{2}$ items before finding the desired one.

1.1 Search Problems in Computer Science

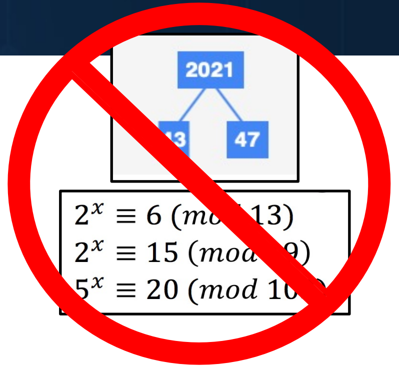
Even in theoretical computer science, the typical problem can be looked at as that of examining a number of different possibilities to see which, if any, of them satisfy a given condition. This is analogous to the search problem stated in the summary above, except that usually there exists some structure to the problem, i.e some sorting does exist on the database. Most interesting

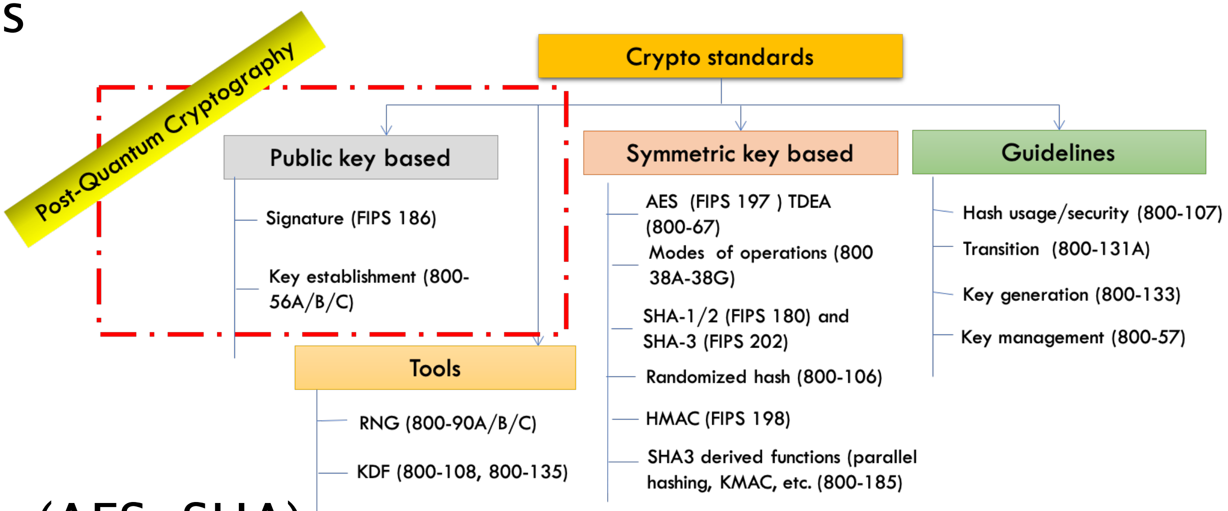
The Quantum Threat

NIST

- NIST public-key crypto standards
  - SP 800-56A: Diffie-Hellman, ECDH
  - SP 800-56B: RSA encryption
  - FIPS 186: RSA, DSA, and ECDSA signatures

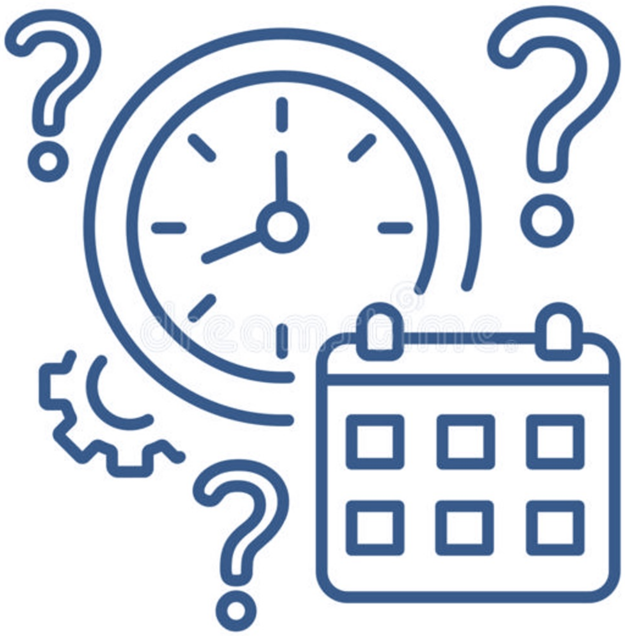
all vulnerable to attacks from a (large-scale) quantum computer





- Symmetric-key crypto (AES, SHA) would also be affected, but less dramatically

How soon do we need to worry?



How soon do we need to worry?

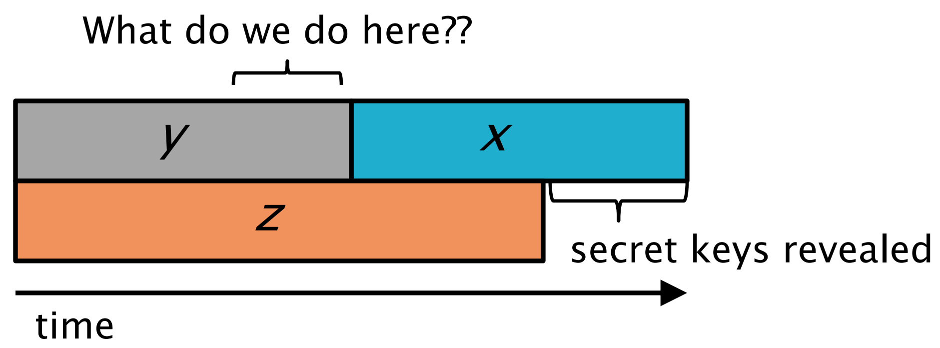
Before quantum computers arrive, obviously

How soon do we need to worry?

~~Before quantum computers arrive, obviously~~

Long before then!

Theorem (Mosca): If $x + y > z$, then problem



x – how long data needs to be safe
y – time for standardization and adoption
z – time until quantum computers

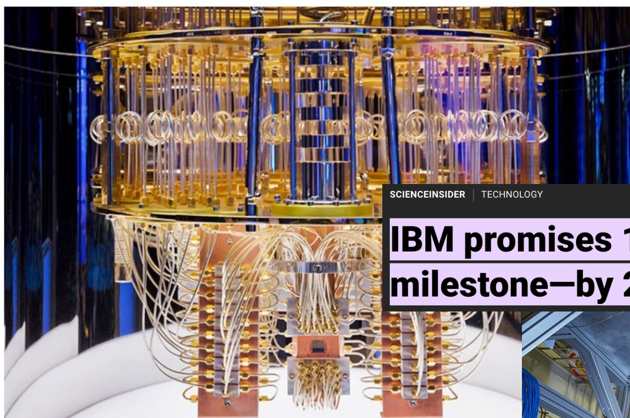
Progress of Quantum Computing

NIST

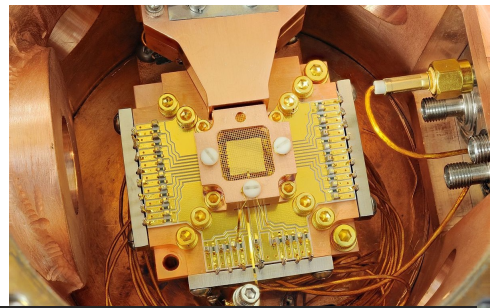
First quantum computer to pack 100 qubits enters crowded race

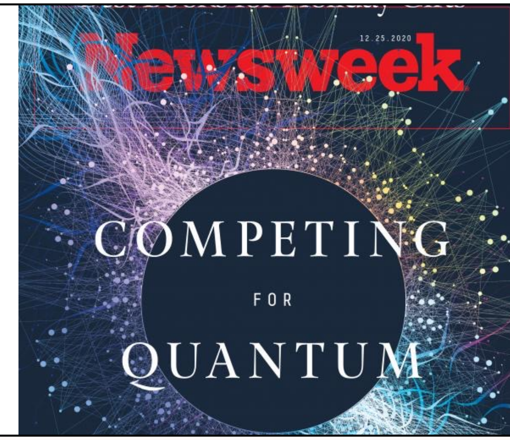
But IBM's latest quantum chip and its competitors face a long path towards making the machines useful.

Philip Ball



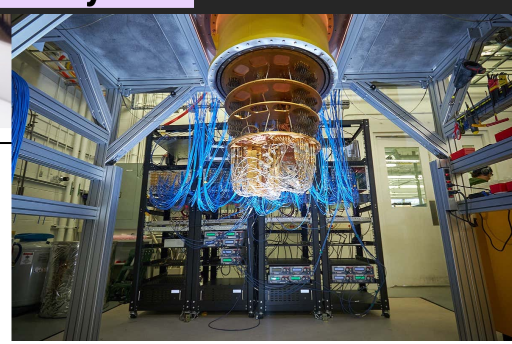
Quantum computers may be able to break Bitcoin sooner than you think





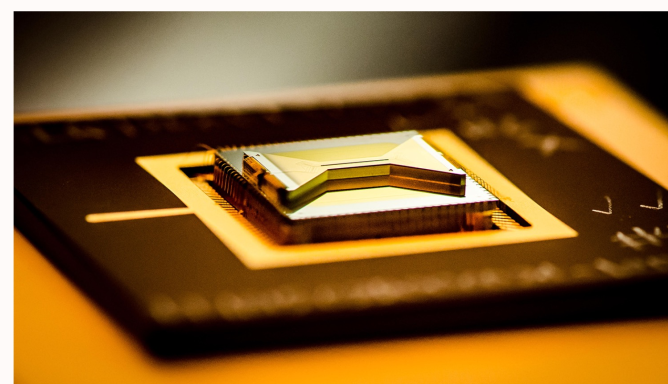
SCIENCEINSIDER | TECHNOLOGY

IBM promises 1000-qubit quantum computer—a milestone—by 2023



Scientists are one step closer to error-correcting quantum computers

Multiple quantum bits were combined into one 'logical qubit' to detect mistakes

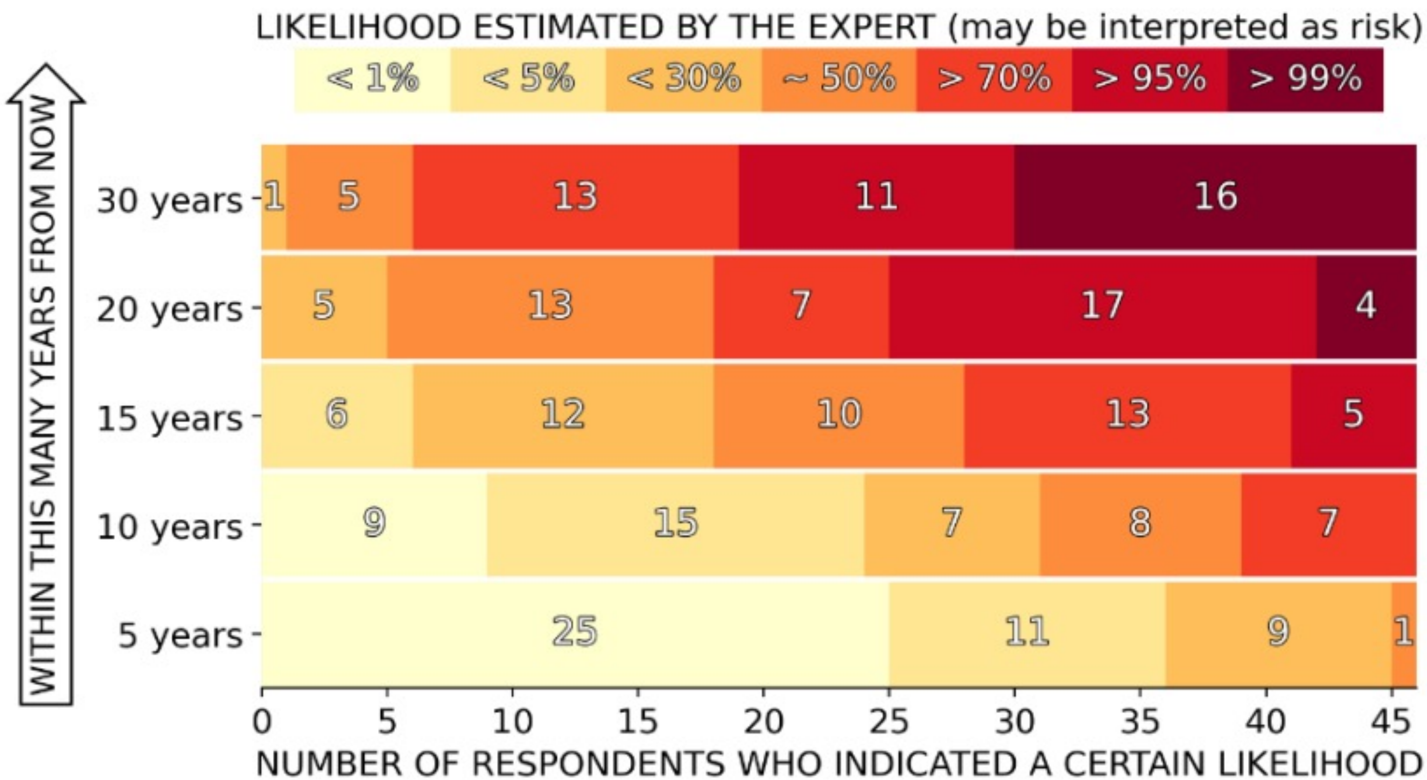


Quantum computing venture backed by Je will leap into public trading with $1.2B valu

When will a Quantum Computer be Built?

EXPERTS' ESTIMATES OF THE LIKELIHOOD OF A QUANTUM COMPUTER ABLE TO BREAK RSA-2048 IN 24 HOURS

The experts were asked to indicate their estimate for the likelihood of a quantum computer that is cryptographically relevant—in the specific sense of being able to break RSA-2048 quickly—for various time frames, from a short term of 5 years all the way to 30 years.



Source: M. Mosca, M. Piani, Quantum Threat Timeline Report, 2021
https://globalriskinstitute.org/publications/2021-quantum-threat-timeline-report//

NIST PQC Milestones and Timelines

NIST

2016
Determined criteria and requirements, published NISTIR 8105
Announced call for proposals

2017
Received 82 submissions
Announced 69 1st round candidates

2018
Held the 1st NIST PQC standardization Conference

2019
Announced 26 2nd round candidates, NISTIR 8240
Held the 2nd NIST PQC Standardization Conference

2020
Announced 3rd round 7 finalists and 8 alternate candidates. NISTIR 8309

2021
Hold the 3rd NIST PQC Standardization Conference

2022 Make 3rd round selection and draft standards

2023 Release draft standards and call for public comments

Call for Proposals

- NIST called for quantum-resistant cryptographic algorithms for new public-key crypto standards
  - Digital signatures
  - Encryption/key-establishment
- Our role: managing a process of achieving community consensus in a **transparent** and timely manner
- Different and more complicated than past AES/SHA-3 competitions
- We will not pick a single “winner”
  - Ideally, several algorithms will emerge as ‘good choices’

Selection criteria

1. **Secure** against both classical and quantum attacks

2. **Performance** – measured on various "classical" platforms

3. **Other properties**
   - Drop-in replacements – Compatibility with existing protocols and networks
   - Perfect forward secrecy
   - Resistance to side-channel attacks
   - Simplicity and flexibility
   - Misuse resistance, and
   - More

Security categories

NIST

Security – against both classical and quantum attacks

Level	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…
  - 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.

The 1st and 2nd Rounds

NIST

Round 1 (Dec ‘17 – Jan ‘18)

- 69 candidates and 278 distinct submitters
- Apr 2018, 1st NIST PQC conference
- Almost 25 schemes broken/attacked
- NISTIR 8240, NIST Report on the 1st Round

Round 2 (Jan ‘18 – Jul ‘20)

- 26 candidates
- Aug 2019 – 2nd NIST PQC conference
- Schemes broken/attacked: LAC, LedaCrypt, Round5, Rollo, RQC, LUOV, MQDSS, qTESLA
- NISTIR 8309, NIST Report on 2nd Round

	Signatures	KEM/Encryption	Overall
Lattice-based	5	21	26
Code-based	2	17	19
Multi-v[illegible]	[illegible]	[illegible]	[illegible]
Symme[illegible]	[illegible]	[illegible]	[illegible]
Other	[illegible]	[illegible]	[illegible]
Total	[illegible]	[illegible]	[illegible]

	Signatures	KEMs/Encryption	Total
Lattice-based	3	9	12
Code-based	0	7	7
Multi-variate	4	0	4
Symmetric-based	2		2
Other	0	1	1
Total	**9**	**17**	**26**

Both rounds: research, cryptanalysis, pqc-forum, official comments, benchmarking, mergers

The 3rd Round Finalists and Alternates

NIST

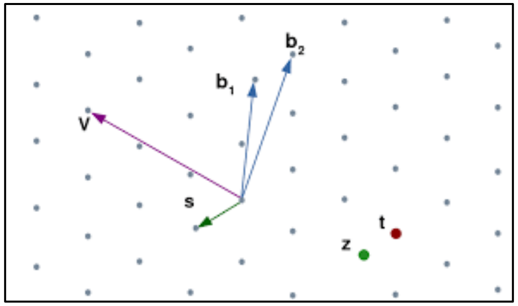
- NIST selected 7 Finalists and 8 Alternates
  - Finalists: most promising algorithms we expect to be ready for standardization at end of 3rd round
  - Alternates: candidates for potential standardization, most likely after another (4th) round
- KEM finalists: Kyber, NTRU, SABER, Classic McEliece
- Signature finalists: Dilithium, Falcon, Rainbow

- KEM alternates: Bike, FrodoKEM, HQC, NTRUprime, SIKE
- Signature alternates: GeMSS, Picnic, Sphincs+

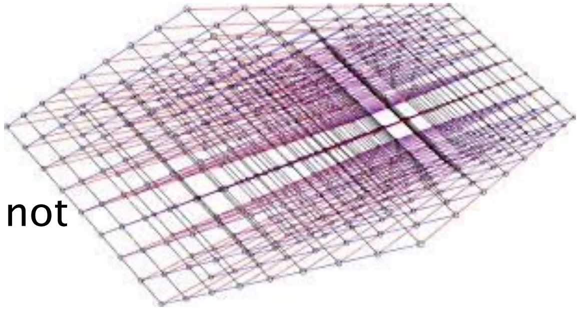
	Signatures		KEM/Encryption		Overall	
Lattice-based	2		3	2	5	2
Code-based			1	2	1	2
Multi-variate	1	1			1	1
Stateless Hash or Symmetric based		2				2
Isogeny				1		1
Total	3	3	4	5	7	8

The Lattice KEMs

- The finalists **Kyber, NTRU, SABER** are based on structured lattices
  - Kyber and SABER are based on module-LWE/LWR
  - NTRU is based on the NTRU problem
  - All three have good performance (in terms of efficiency and key/ciphertext sizes)
  - *NIST expects to select at most one for standardization*



- The alternates **NTRU Prime** and **FrodoKEM** are based on lattices
  - NTRUprime uses structured lattices, while FrodoKEM does not

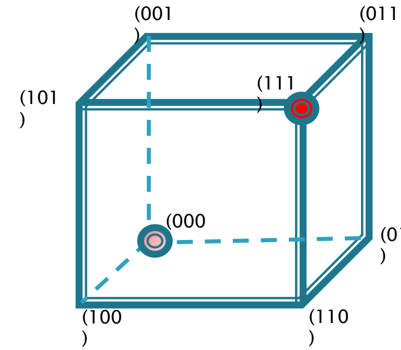


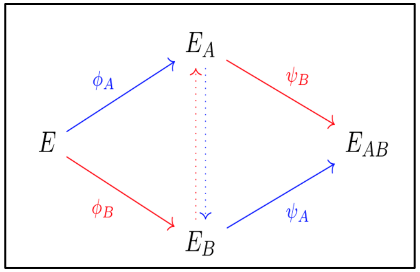
The Other KEMs

- **Classic McEliece**, the other finalist, is code-based
  - Been around since 1978
  - Very large public keys, but very small ciphertexts

- The alternates **BIKE** and **HQC** are based on structured codes
  - Both have much smaller key sizes than Classic McEliece

- The final alternate **SIKE** is based on isogenies of elliptic curves
  - Small key/ciphertext sizes, slower than other candidates





The Signatures

- The finalists **Dilithium** and **Falcon** are both based on structured lattices
  - Dilithium is Fiat-Shamir style, while Falcon is hash then sign
  - Both have good performance

- The alternate **Picnic** is based on zero-knowledge proofs and a block cipher
- The alternate **SPHINCS+** is based on the security of hash functions
  - The security of SPHINCS+ is very well understood
  - SPHINCS+ is stateless

- There are two multivariate schemes: the finalist **Rainbow**, and the alternate **GeMSS**
  - Both have large public keys, and very small signature sizes

The state of the signatures

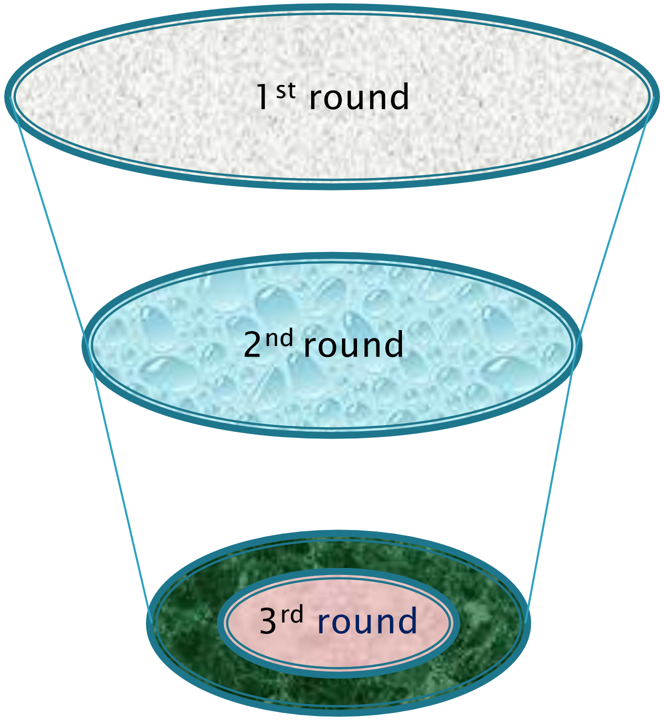
- Cryptanalytic results during the 3rd round have created some concerns about the security of both multivariate schemes **Rainbow** and **GeMSS**
- Beullens recently posted a new attack on **Rainbow**
  - Breaks category 1 parameters in “a weekend on a laptop”
  - Serves as a reminder to not put candidates into products until the standard is done
- In Jan 2021, NIST asked for feedback on two topics:
  - Standardizing SPHINCS+ after 3rd round
  - Introducing a mechanism to consider new signature schemes

How will NIST make its decisions?

NIST

Using the evaluation criteria:

- Security
  - Security levels offered
  - (confidence in) security proof
  - Any attacks
  - Classical/quantum complexity
- Performance
  - Size of parameters
  - Speed of KeyGen, Enc/Dec, Sign/Verify
  - Software and hardware benchmarks
- Algorithm and implementation characteristics
  - IP issues
  - Decryption failures
  - Side channel resistance
  - Simplicity and clarity of documentation
  - Flexible
- Other
  - Official comments/pqc-forum discussion
  - Papers published/presented



How will NIST make its decisions?

- For the lattice KEMs, the main decision will be **Kyber/NTRU/Saber**
- Similarly for lattice signatures, the main decision will be **Dilithium/Falcon**
- Any other algorithms selected will be their own distinct decision
  - Other Finalists: Classic McEliece and Rainbow
  - KEM alternates: Bike, HQC, FrodoKEM, NTRUprime, SIKE
  - Signature alternates: GeMSS, Picnic, Sphincs+

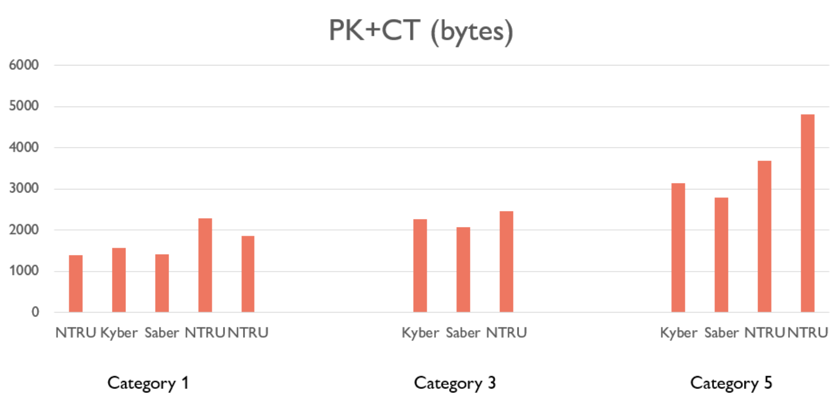
Kyber vs NTRU vs Saber

- Kyber and Saber based on Module-Learning With Errors/Rounding
- NTRU is based on NTRU problem

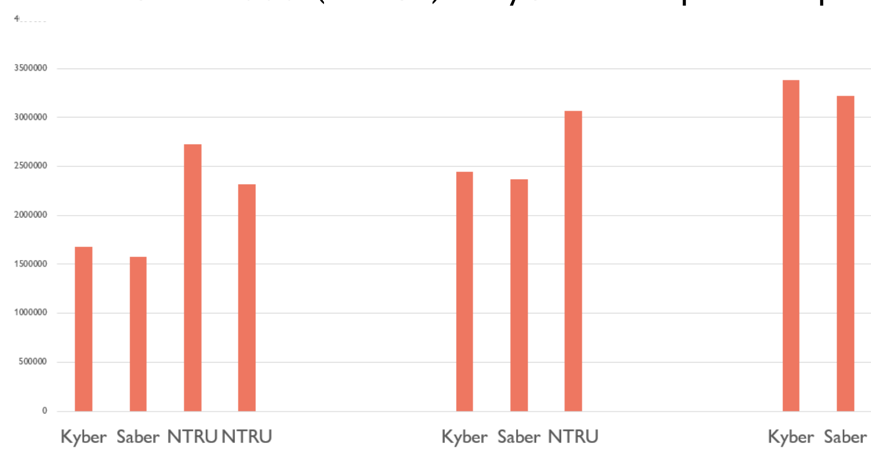
- Each has an IND-CCA2 proof, constructed from PKEs using some type of Fujisaka-Okamoto transform
  - Kyber and Saber have decryption failure, NTRU does not

- Kyber, Saber use modules with ring $\mathbb{Z}_q[x]/\left\langle x^{2^k}+1\right\rangle$, NTRU uses ring $\mathbb{Z}_q[x]/\langle x^p - 1\rangle$

Performance – bandwidth graph



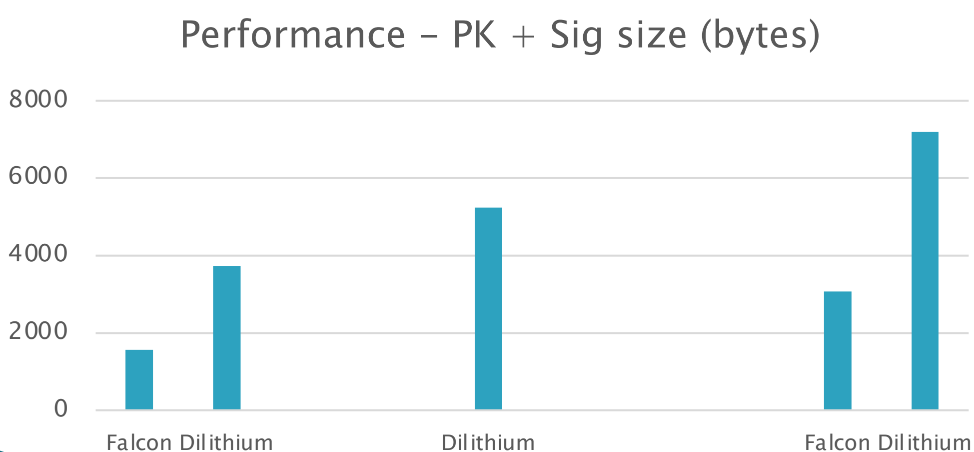
Total Cost: 1000*(PK+CT)+KeyGen+Encaps+Decaps



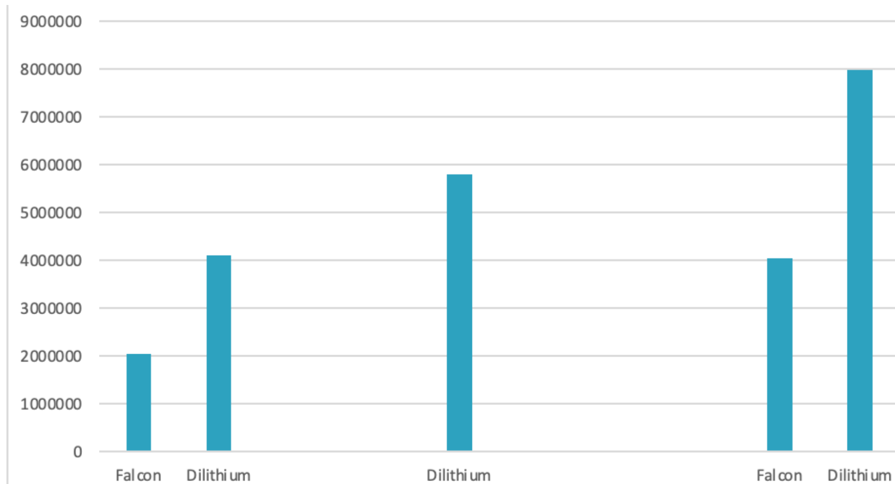
Software – AVX2 processor

Dilithium vs Falcon

- Dilithium is based on module-LWE, Falcon is based on SIS over NTRU lattices
- Dilithium uses Fiat-Shamir with aborts, uniform sampling
- Falcon uses Hash-then-sign paradigm, Gaussian sampling.
    - Falcon has a very complex implementation, KeyGen is comparatively slow
- Both use rings of the form $\mathbb{Z}_q[x]/\langle x^{2^k}+1\rangle$
- Each has an EUF-CMA proof



Total Cost: 1000*(PK+Sig)+Sign+Verify



Software – AVX2 processor

Patent and IPR issues

- "NIST does not object in principle to algorithms or implementations which may require the use of a patent claim, where technical reasons justify this approach, ***but will consider any factors which could hinder adoption in the evaluation process.***"

- This is a very complicated area
- We acknowledge the impact of encumbered technology on adoption

- NIST is actively engaging to try to resolve known IPR issues on the candidates
- When we have something concrete, we will share it

Note: it may not be possible for NIST to resolve all IP concerns

Timeline

- The 3^{rd} Round will end any day now!
  - NIST will announce which finalist algorithms it will standardize
    - Including potentially the alternate SPHINCS+
  - This will include algorithms which will be able to be used by most applications
  - NIST will issue a Report on the 3^{rd} Round to explain our decisions

- NIST will also announce any candidates advancing to 4^{th} round
  - The 4^{th} round will similarly be 18-24 months
  - These algorithms will be for a diversified portfolio

- We'll likely hold a workshop in fall/winter 2022
- We plan to release draft standards for public comment in 2022-2023
- The first set of standards should be finalized by 2024

Standardization

- NIST's public-key crypto is standardized in:
  - FIPS 186-5, digital signatures
  - SP 800-56A, 800-56B, encryption/key-establishment
- NIST will create new standards, in consultation with the candidate teams
  - NIST will determine which specific parameter sets to include, and give their security strength
  - NIST will seek feedback from community, if needed
- The draft standards will be put out for public comment
  - Feedback received will be made public
  - NIST will make any necessary revisions and then publish the Standard

An on-ramp for signatures

- After the conclusion of the 3rd Round, NIST will issue a new Call for Signatures
  - There will be a deadline for submission, likely Jan 2023
  - This will be much smaller in scope than main NIST PQC effort
  - The main reason for this call is to diversify our signature portfolio
  - These signatures will be on a different track than the candidates in the 4th round

- We are **most interested** in a general-purpose digital signature scheme which is not based on structured lattices
  - We may be interested in other signature schemes targeted for certain applications. For example, a scheme with very short signatures.

- The more mature the scheme, the better.

- NIST will decide which (if any) of the received schemes to focus attention on

Research Challenges

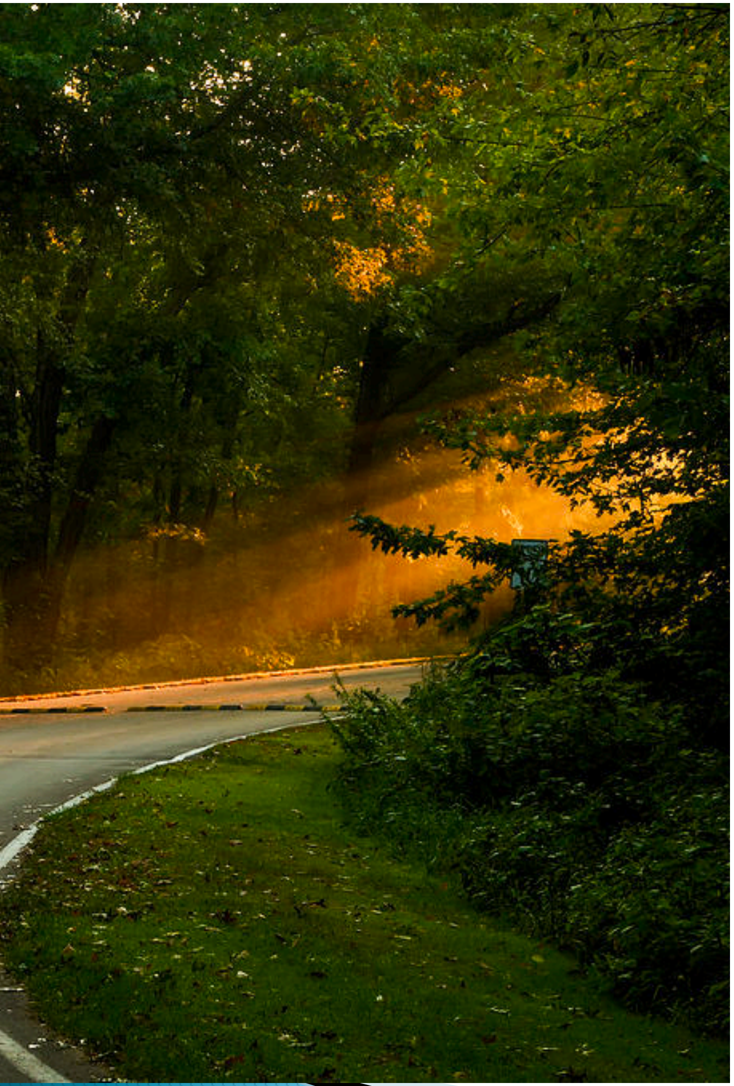
- Many important topics studied:
  - Security proofs in both the ROM and QROM
  - Does the specific ring/module/field choice matter for security?
    - Or choice of noise distribution?
    - Does "product" or "quotient" style LWE matter?
  - Finer-grained metrics for security of lattice-based crypto (coreSVP vs. real-world security)
    - More generally, what cost models should we be using to measure attacks?
  - Are there any important attack avenues that have gone unnoticed?
  - Side-channel attacks/resistant implementations
  - More hardware implementations
  - Ease of implementations – decryption failures, floating point arithmetic, noise sampling, etc.
  - Algebraic cryptanalysis of cyclotomics for lattices

Other Standards Organizations

- We are aware that many standards organizations and expert groups are working on PQC
  - IEEE P1363.3 has standardized some lattice-based schemes
  - IETF has standardized stateful hash-based signatures LMS/XMSS
  - ETSI has released quantum-safe cryptography reports
  - EU expert groups PQCRYPTO and SAFEcrypto made recommendations and released reports
  - ISO/IEC JTC 1 SC27 had a study period for quantum-resistant cryptography and released a standing document (SD)

- NIST is interacting and collaborating with these organizations and groups

- Some countries have begun standardization activities



Conclusion

- The Beginning of the End is here!
- NIST is grateful for everybody’s efforts
- Check out www.nist.gov/pqcrypto
  - Sign up for the pqc-forum for announcements & discussion
  - send e-mail to pqc-comments@nist.gov