THE DRIVE TOWARDS POST-QUANTUM COMPUTING RESISTANT CRYPTOGRAPHIC ALGORITHM STANDARDIZATION

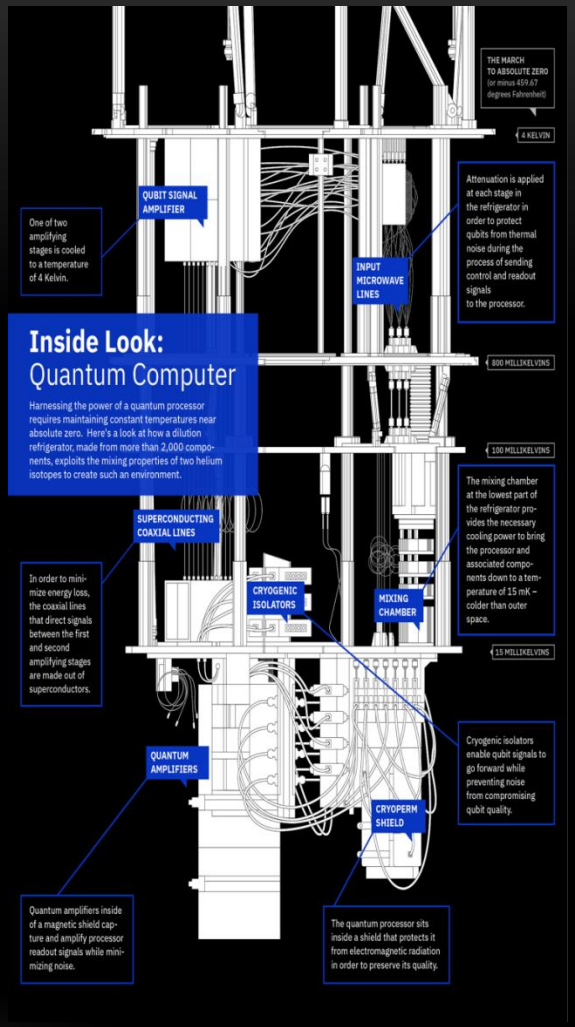
Richard Tychansky, CISSP-ISSEP, GRCP, CSSLP, CCSP, PMP

LEARNING OBJECTIVES

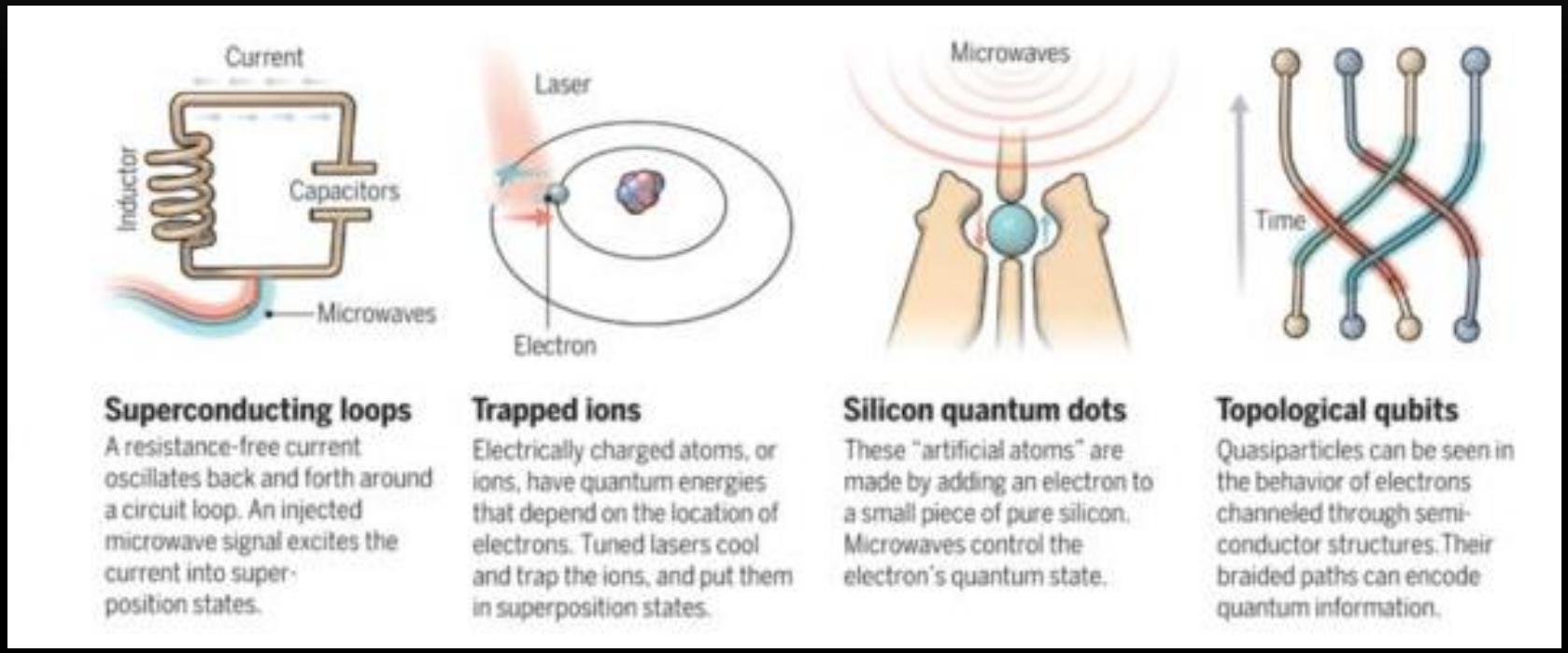
- Describe the benefits and risks to security that quantum computing presents.
- Be able to recall the NIST PQC candidates.
- Assess organizational cryptographic exposure and have the ability to plan the implementation of quantum-resistant cryptography.

ASSUMPTIONS

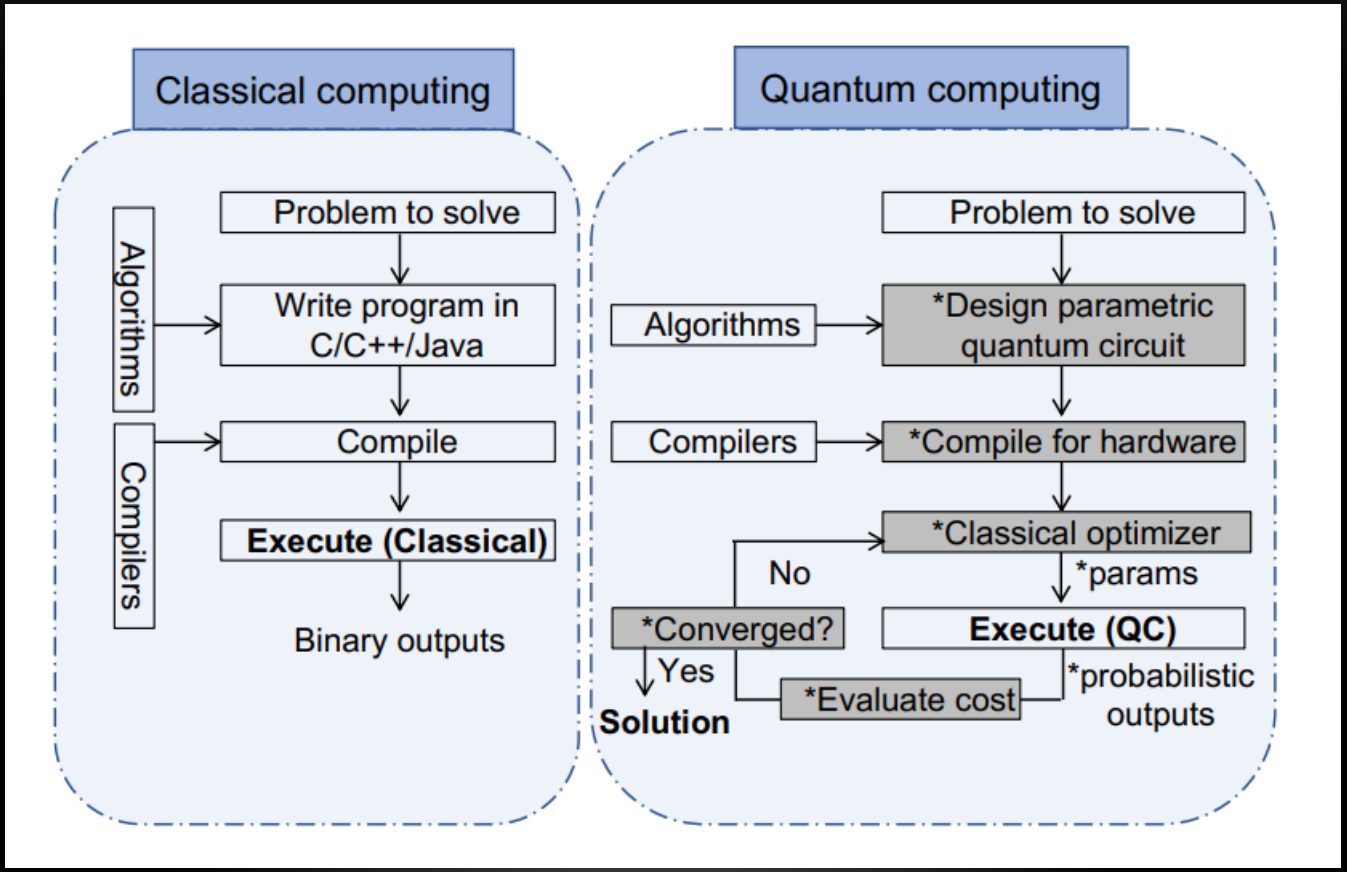
- Some familiarity with the quantum computing paradigm.
- Knowledge of prime numbers and factoring algorithms such as Shor and Grovers.
- Some knowledge of how asymmetric cryptography works.
- Some familiarity with NIST standards.



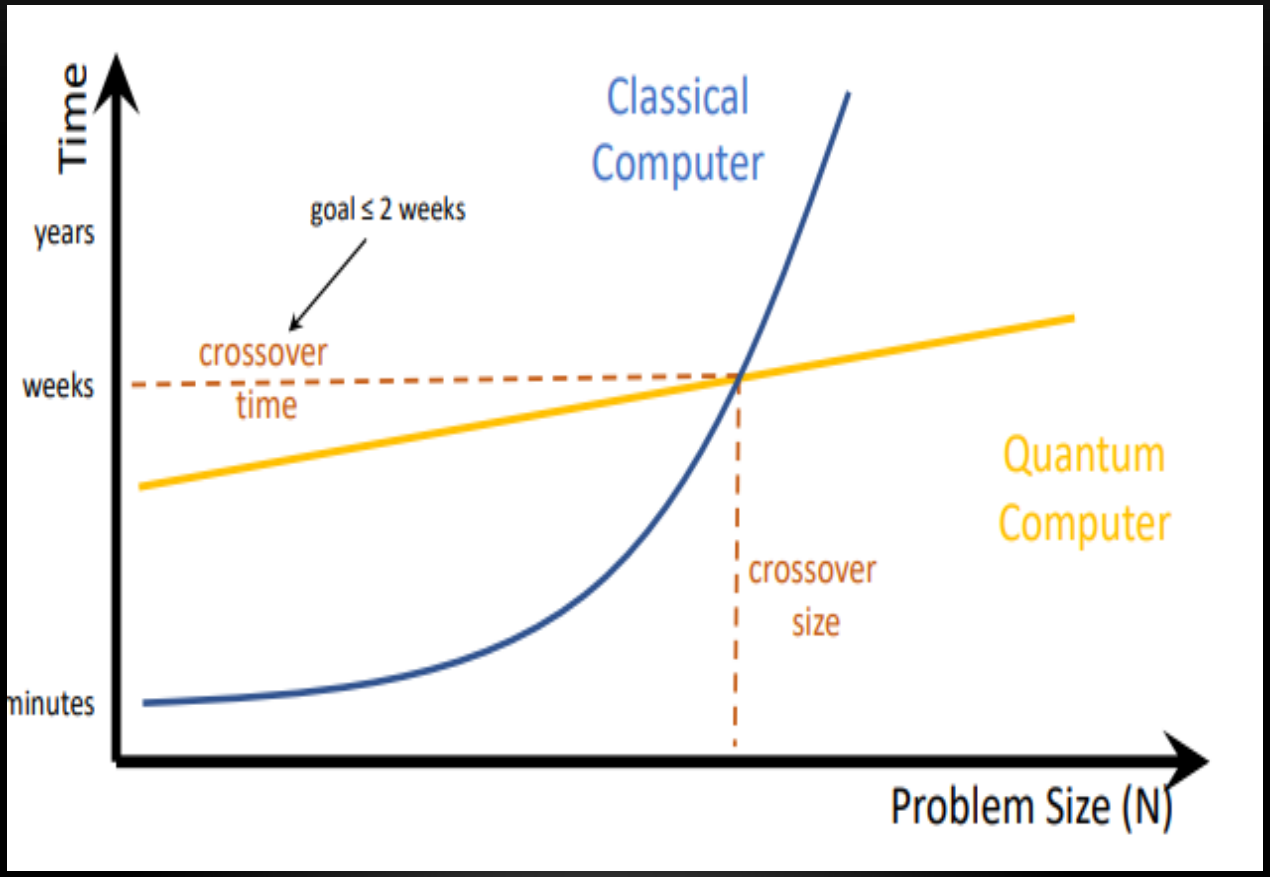
TECHNOLOGIES USED FOR MAKING QUBITS



CLASSICAL VS QUANTUM COMPUTING



QUANTUM SPEEDUP



NSA ANNOUNCEMENT

- Aug 2015 - NSA's Information Assurance Directorate updated its list of Suite B cryptographic algorithms

- "IAD will initiate a transition to **quantum resistant algorithms** in the **not too distant future**. Based on experience in deploying Suite B, we have determined to start planning and communicating early about the upcoming transition to quantum resistant algorithms."

NIST SP 1800-38A, MIGRATION TO POST-QUANTUM CRYPTOGRAPHY: PREPARATION FOR CONSIDERING THE IMPLEMENTATION AND ADOPTION OF QUANTUM SAFE CRYPTOGRAPHY

- Many of the cryptographic products, protocols, and services used today, in particular those using public-key algorithms like Rivest-Shamir-Adleman algorithm (RSA), Elliptic Curve Diffie Hellman (ECDH), and Elliptic Curve Digital Signature Algorithm (ECDSA), need to be updated, replaced, or significantly altered to use quantum-resistant algorithms.

- Many public-key algorithms and the protocols that use them will be vulnerable to attacks. A majority of today's information and communication technology systems are not designed to support rapid adaptations of new cryptographic algorithms without making significant changes to the systems' components.

THE INTERNET E-COMMERCE CRYPTOGRAPHIC APOCALYPSE

- How quantum computation might be used to decode public-keys based on large prime numbers.
- The goal is the transmission of “secret” messages, which for computational purposes are taken to be strings of bits.
- For maximum security, the key should consist of a random bit string of same length as the message, and the key should be used for one time only.
- The difficulty with this scheme is that the key must be shared between the sender and receiver via a communication channel that is subject to “eavesdropping”, i.e., a public channel.

WHY QUANTUM COMPUTERS REPRESENT A CYBER THREAT TO FINANCIAL DATA

- Quantum computers, should they reach sufficient size and power, may be able to break the cryptographic schemes widely used today to ensure secure financial transactions and data.

- This makes quantum computing one of the most important cybersecurity threats facing the financial system, potentially exposing all financial transactions and much of our existing stored financial data to attack.

- Implementing quantum-resistant communication is already feasible today, but it is crucial to understand how these new standards will affect banking operations.

- The operational characteristics of these new protocols are for the most part unknown since real-time processes have not been adequately tested…

HARVEST NOW, DECRYPT LATER? THE TRUTH BEHIND THIS COMMON QUANTUM THEORY

- To many, the term “quantum computing” equates to a world of new possibilities. What started as a theoretical curiosity is now touted as the future of IT.
- To others in cybersecurity they think the sky is about to fall.
- With quantum, there will be lightning fast processing of information and computers will be able to answer problems previously thought of as unsolvable.
- The foundation behind this fear, uncertainty, and doubt (FUD) revolves around the future threat that quantum computers have for existing data. Commonly referred to as Harvest Now, Decrypt Later (HNDL), this theory centers on concerns that a nation-state will gain access to currently encrypted data and then decrypt it at a later time using a quantum computer.
- Regardless of when these breakthroughs occur, it is undisputed that the overall value of quantum computing is driving research at an accelerated pace.

BREAKING TODAY'S CRYPTOGRAPHY

- Once effective quantum computers are available, they will essentially eliminate the cryptographic strength of public-key (e.g., RSA) cryptosystems.
- More traditional cryptosystems (e.g., AES) will also be affected, reducing their effective security strength to roughly half of what we would consider it to be today.
- This will have a devastating effect on the systems used to protect electronic communications and digital transactions.
- Most secure internet processes rely on protocols that employ public-key cryptography, including those used to secure web sites, for banking transactions, secure email and digital signatures.

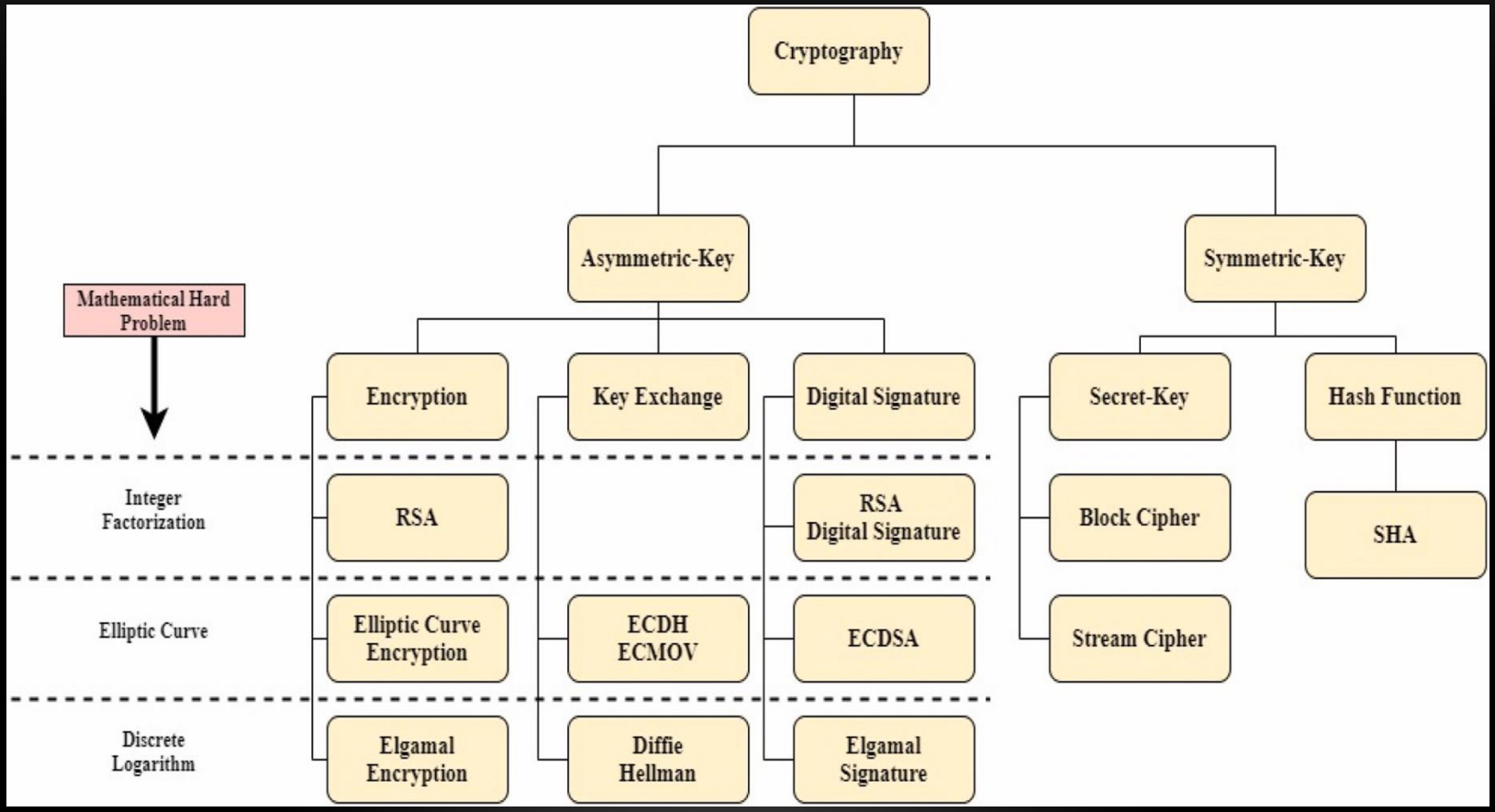
ALGORITHMS FOR BREAKING TODAY'S CRYPTO

- The most well known algorithms are Shor's algorithm for factoring, and Grover's algorithm for searching an unstructured database or an unordered list.
- **Shor's** algorithm runs **exponentially** faster than the best known classical algorithm for factoring,
- **Grover's** algorithm runs **quadratically** faster than the best possible classical algorithm for the same task, a linear search.
- Quantum Computation and Shor's Factoring Algorithm
- The challenge is implementing them to run on a quantum computer.

THE IMPACT OF QUANTUM COMPUTING ON COMMON CRYPTOGRAPHIC ALGORITHMS

Cryptographic Algorithm	Type	Purpose	Impact from Large Scale Quantum Computer
AES	Symmetric Key	Encryption	Larger key sizes needed
SHA-2, SHA-3	—————	Hash Functions	Larger output needed
RSA	Public Key	Signatures, Key establishment	No longer secure
ECDSA, ECDH (Elliptical Curve Cryptography)	Public Key	Signatures, Key Exchange	No longer secure
DSA (Finite Field Cryptography)	Public Key	Signatures, Key Exchange	No longer secure

THE IMPACT OF SHOR AND GROVERS ALGORITHMS ON TODAY'S CRYPTOSYSTEMS



$2^{82,589,933} - 1$

PETER SHOR

Recognition that a functional quantum computer could break current cryptography

Availability of a quantum computer

Data are exposed to "harvest now, decrypt later" attacks

1994 Today

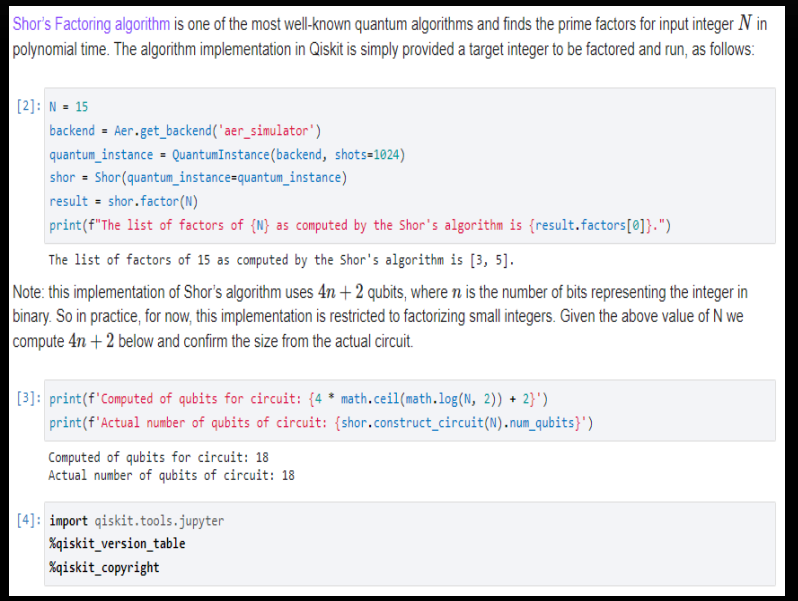
Migration to post-quantum cryptography

Data are protected

Standardisation process of post-quantum cryptography

SHOR'S FACTORING ALGORITHM (IN QISKIT)

- The PQC research field has flourished over the past decade, leading to the creation of a large variety of algorithms that are expected to be resistant to quantum attacks.
- Quantum computers are expected to break modern public-key cryptography owing to Shor's algorithm. As a result, these cryptosystems need to be replaced by quantum-resistant algorithms, also known as post-quantum cryptography (PQC) algorithms.
- These PQC algorithms are being selected and standardized by several standardization bodies.
- However, even with the guidance from these important efforts, the danger is not gone: there are **billions of old and new devices that need to transition to the PQC suite of algorithms**, leading to a **multidecade transition process** that has to account for aspects such as security, algorithm performance, ease of secure implementation, compliance and more.



Shor's Factoring algorithm is one of the most well-known quantum algorithms and finds the prime factors for input integer N in polynomial time. The algorithm implementation in Qiskit is simply provided a target integer to be factored and run, as follows:

[2]:
```
N = 15
backend = Aer.get_backend('aer_simulator')
quantum_instance = QuantumInstance(backend, shots=1024)
shor = Shor(quantum_instance=quantum_instance)
result = shor.factor(N)
print(f"The list of factors of {N} as computed by the Shor's algorithm is {result.factors[0]}.")
```

```
The list of factors of 15 as computed by the Shor's algorithm is [3, 5].
```

Note: this implementation of Shor's algorithm uses $4n + 2$ qubits, where n is the number of bits representing the integer in binary. So in practice, for now, this implementation is restricted to factorizing small integers. Given the above value of N we compute $4n + 2$ below and confirm the size from the actual circuit.

[3]:
```
print(f'Computed of qubits for circuit: {4 * math.ceil(math.log(N, 2)) + 2}')
print(f'Actual number of qubits of circuit: {shor.construct_circuit(N).num_qubits}')
```

```
Computed of qubits for circuit: 18
Actual number of qubits of circuit: 18
```

[4]:
```
import qiskit.tools.jupyter
%qiskit_version_table
%qiskit_copyright
```

BUT WAIT IT GETS BETTER…

Oded Regev developed a multidimensional version of Shor's algorithm that runs even faster.

"We show that n-bit integers can be factorized by independently running a quantum circuit with O˜(n 3/2) gates for √ n + 4 times, and then using polynomial-time classical post-processing."

https://arxiv.org/pdf/2308.06572.pdf

A NEW PAPER EXTENDS SHOR'S ALGORITHM TO MULTIPLE DIMENSIONS

- In the past 30 years, computer scientists have streamlined Shor's algorithm in preparation for the day that quantum technology matures enough to run it.
- A new variant, from the New York University computer scientist Oded Regev, is faster in a fundamentally new sense.
- **It's the first to improve the relationship between the size of the number being factored and the number of quantum operations required to factor it.**
- Regev's paper is interesting but cautioned that beating the state of the art in practice will require further optimization. "Shor's original algorithms are already surprisingly efficient, so it is not trivial to make major improvement
- Regev developed his new algorithm by augmenting Shor's algorithm with techniques from a branch of cryptography dealing with **high-dimensional geometry.**

NIST: POST-QUANTUM CRYPTOGRAPHY (PQC) "COMPETITION"

QUANTUM SECURITY – HOW TO ASSESS ALGORITHMS?

- No clear consensus on best way to measure quantum attacks.
- Uncertainties
  - The possibility that new quantum algorithms will be discovered, leading to new attacks.
  - The performance characteristics of future quantum computers, such as their cost, speed and memory size.
  - Currently, NIST crypto standards specify parameters for classical security levels at 112, 128, 192, 256 bits.
  - For PQC standardization, the need is to specify concrete parameters with security estimates.

	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)

NIST POST-QUANTUM CRYPTOGRAPHY STANDARDIZATION

- Post-Quantum Cryptography Lounge – the most concise source on candidate algorithms:
  - Signature Performance test results
  - Encryption Performance test results
  - KEM (key generation, encryption/decryption)
- Automated testing using pqbench
- Lots of discussion in the crypto community over testing, algorithm security assertions and mathematical proofs.

CRYPTANALYSIS TOOLS: FEATHERDUSTER

- FeatherDuster - An automated, modular cryptanalysis tool; i.e., a **Weapon of Math Destruction**.

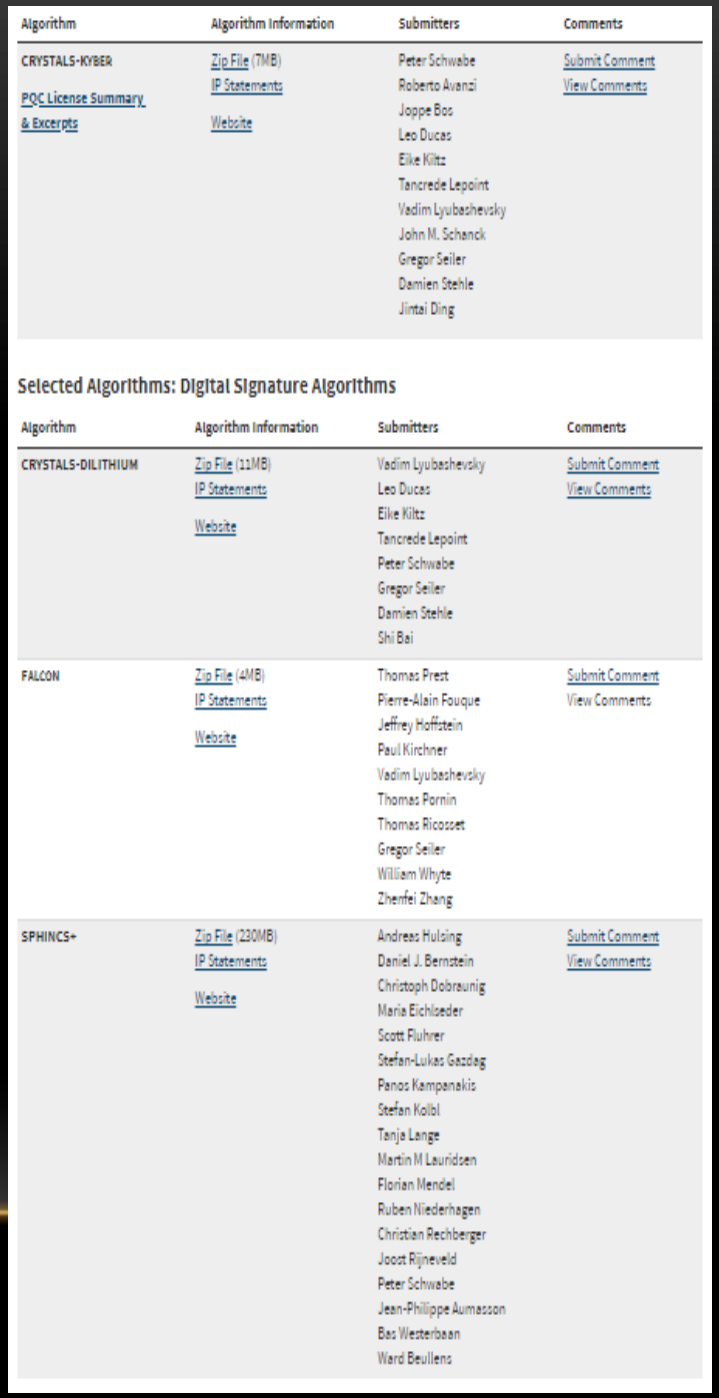
- Written by "unicornfurnace" for breaking crypto which tries to make the process of identifying and exploiting weak cryptosystems as easy as possible.
  - **Cryptanalib** is the moving part behind FeatherDuster, and can be used independently of FeatherDuster.

THE POST-QUANTUM STACK ACCORDING TO NIST

Applications	Web browsers, certificates, Tor, Signal, software updates, secure boot, etc.
Protocols	SSL, TLS, IPSec (IKE)
Cryptography	Post-Quantum: • Encryption • Signatures • Key exchange Hybrid Modes (PQC+classical)
Lower level	Libraries (GMP, NTL, etc.) Block ciphers (AES) Hash functions (SHA-2/3) RNGs Hardware support for maintain state for hash-based signatures

SELECTED PQC ALGORITHMS

- **Public-key Encryption Algorithm**
  - CRYSTALS-KYBER
    - Kyber-1024 aims at security roughly equivalent to AES-256.
- **Digital Signature Algorithms**
  - CRYSTALS-DILITHIUM
  - FALCON
  - SPHINCS+



Algorithm	Algorithm Information	Submitters	Comments
CRYSTALS-KYBER PQC License Summary & Excerpts	Zip File (7MB) IP Statements Website	Peter Schwabe Roberto Avanzi Joppe Bos Leo Ducas Eike Kiltz Tancrede Lepoint Vadim Lyubashevsky John M. Schanck Gregor Seiler Damien Stehle Jintai Ding	Submit Comment View Comments

Selected Algorithms: Digital Signature Algorithms

Algorithm	Algorithm Information	Submitters	Comments
CRYSTALS-DILITHIUM	Zip File (11MB) IP Statements Website	Vadim Lyubashevsky Leo Ducas Eike Kiltz Tancrede Lepoint Peter Schwabe Gregor Seiler Damien Stehle Shi Bai	Submit Comment View Comments
FALCON	Zip File (4MB) IP Statements Website	Thomas Prest Pierre-Alain Fouque Jeffrey Hoffstein Paul Kirchner Vadim Lyubashevsky Thomas Pornin Thomas Ricosset Gregor Seiler William Whyte Zhenfei Zhang	Submit Comment View Comments
SPHINCS+	Zip File (230MB) IP Statements Website	Andreas Hulsing Daniel J. Bernstein Christoph Dobraunig Maria Eichlseder Scott Fluhrer Stefan-Lukas Gazdag Panos Kampanakis Stefan Kolbl Tanja Lange Martin M Lauridsen Florian Mendel Ruben Niederhagen Christian Rechberger Joost Rijneveld Peter Schwabe Jean-Philippe Aumasson Bas Westerbaan Ward Beullens	Submit Comment View Comments

FIPS 203, MODULE-LATTICE-BASED KEY-ENCAPSULATION MECHANISM STANDARD (DRAFT)

- A key-Encapsulation Mechanism (or KEM) is a set of algorithms that, under certain conditions, can be used by two parties to establish a shared secret key over a public channel. A shared secret key that is securely established using a KEM can then be used with symmetric-key cryptographic algorithms to perform basic tasks in secure communications, such as encryption and authentication.

- This standard specifies a key-encapsulation mechanism called ML-KEM. The security of ML-KEM is related to the computational difficulty of the so-called **Module Learning with Errors** problem.
  - At present, ML-KEM is believed to be secure even against adversaries who possess a quantum computer.

- This standard specifies three parameter sets for ML-KEM. In order of increasing security strength (and decreasing performance), these parameter sets are ML-KEM-512, ML-KEM-768, and ML-KEM-1024.

FIPS 204, MODULE-LATTICE-BASED DIGITAL SIGNATURE STANDARD (DRAFT)

- Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory.
- In addition, the recipient of signed data can use a digital signature as evidence in demonstrating to a third party that the signature was, in fact, generated by the claimed signatory.
- This is known as non-repudiation since the signatory cannot easily repudiate the signature at a later time.
- This standard specifies ML-DSA, a set of algorithms that can be used to generate and verify digital signatures. ML-DSA is believed to be secure even against adversaries in possession of a large-scale quantum computer.

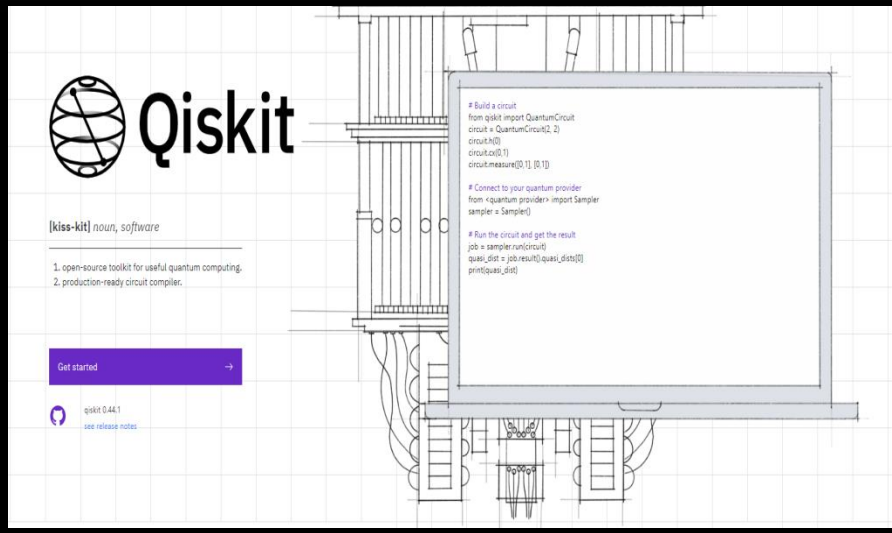
FIPS 205, STATELESS HASH-BASED DIGITAL SIGNATURE STANDARD (DRAFT)

- This standard specifies the stateless hash-based digital signature algorithm (SLH-DSA).
- Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory.
- In addition, the recipient of signed data can use a digital signature as evidence in demonstrating to a third party that the signature was, in fact, generated by the claimed signatory.
- This is known as non-repudiation since the signatory cannot easily repudiate the signature at a later time. SLH-DSA is based on SPHINCS$^+$, which was selected for standardization as part of the NIST Post-Quantum Cryptography Standardization process.

TRANSITIONING ORGANIZATIONS TO POST-QUANTUM CRYPTOGRAPHY. ARE YOU READY?

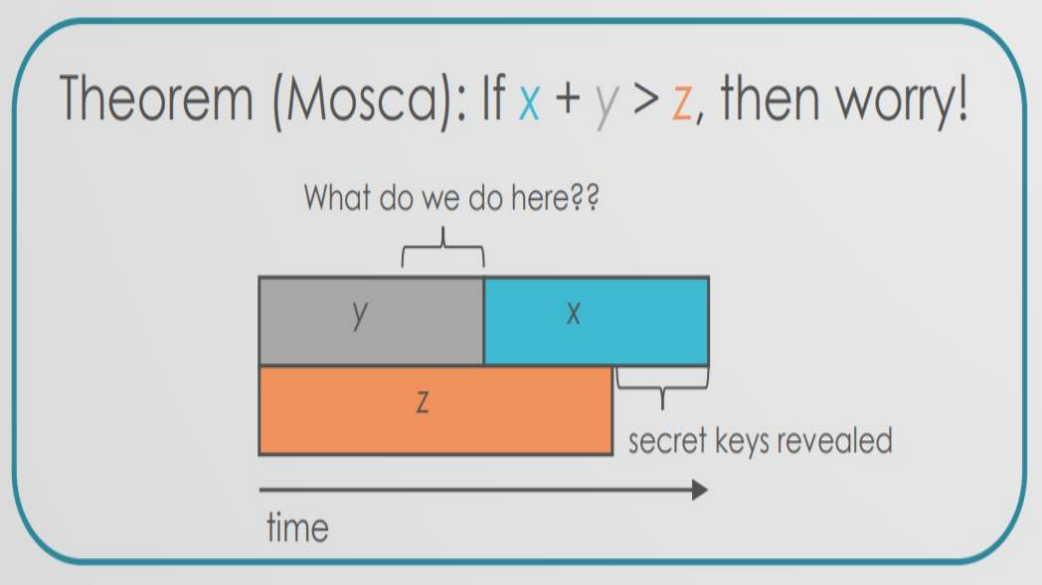
Steps:

1. Prepare a cryptographic inventory
2. Discuss Post-Quantum Roadmaps with Vendors
3. Assess Supply Chain Readiness
4. Profile your adversaries and their capabilities.



QUANTUM RISK ASSESSMENT

- Assess organizational cryptographic exposure and have the ability to plan the implementation of quantum-resistant cryptography.



QUANTUM RISK ASSESSMENT: PHASE 1

- Identify and document information assets, and their current cryptographic protection.
  - Inventory information assets which require cryptographic protection, in accordance with the organization's security policy.
  - Identify the nature of the cryptography being used, how encryption keys are generated, stored and applied, and the origin of tools or appliances employed in these processes.
  - Document business value, access controls, backup strategies, and data sharing arrangements

QUANTUM RISK ASSESSMENT: PHASE 2

- Research the state of emerging quantum computers and quantum-safe cryptography.
- Understand the relevance and impact of specific research developments.
- Develop partnerships with the cryptographic community.
- Estimate the timelines for availability of the technologies and start planning
- Work to influence the development and validation of quantum-safe cryptography within your organization.

QUANTUM RISK ASSESSMENT: PHASE 3

- Identify threat actors, and estimate their time to access quantum technology "**z**" (i.e., the "collapse time")
- Security conscious organizations should be aware of their most significant threat actors.
- Focus on the likelihood that a threat actor will gain access to a quantum computer.

QUANTUM RISK ASSESSMENT: PHASE 4

- Identify the lifetime of your assets "**x**", and the time required to transform the organization's technical infrastructure to a quantum-safe state "**y**".

- Determining the useful lifetime of your business information is critical to understanding your organization's quantum vulnerability. If an adversary can capture and archive your encrypted information, how long will it remain useful? This will be governed by the nature of your business, your products and your clients, as well as by regulatory requirements that may apply to your organization.

- Consider the tools available to combat a quantum-powered threat actor.

- Examine the strength of the existing cryptography, and how effectively it is being applied and used.

- Review available quantum-safe cryptographic methods, to determine whether they might be appropriate replacements for existing capabilities.

- Having this information we can compute the remaining values of the risk model - the Shelf Life of an organization's data (**x**) and the infrastructure Migration Time (**y**).

QUANTUM RISK ASSESSMENT: PHASE 5

- Determine quantum risk by calculating whether business assets will become vulnerable before the organization can move to protect them. **(x + y > z ?)**
- Next, assess the risk the organization faces as quantum computers emerge.
- Consider the lifetime of sensitive data is considered, including its likelihood of exposure; and combine that information with the time required to migrate existing processes and infrastructure.
- This is compared to the timeframe in which quantum technology will be available to relevant threat actors. Taken together, this provides a reasonable estimate of when the organization needs to be taking active steps to **mitigate quantum risk**.
- Assess the business process impact that results from anticipated changes in products, protocols, and procedures.
- Consider whether or not quantum-safe technologies introduce latencies, reliability or performance issues that need to be addressed.

QUANTUM RISK ASSESSMENT: PHASE 6

- Identify and prioritize the activities required to maintain awareness, and to migrate the organization's technology to a quantum-safe state.

- The quantum risk assessment provides information and guidance towards a quantum-safe status.

- Create migration plans to respond to changes as vendors incorporate quantum capabilities into their products and tools. It is important to track all these, and most organizations should develop a roadmap that addresses immediate concerns while permitting the incorporation of new quantum technologies as they become available.

- Any cyber risk assessment must be periodically updated to account for emerging threats and to take advantage of improved security solutions. This is particularly true for quantum technologies, which are rapidly evolving.

WHY WAIT FOR STANDARDS?

BUILD FOR TOMORROW. TODAY.

OPEN QUANTUM SAFE (OQS)

- **liboqs** - C library for prototyping and experimenting with quantum-resistant cryptography

  - Part of the **Open Quantum Safe (OQS)** project led by Douglas Stebila and Michele Mosca, which aims to develop and integrate into applications quantum-safe cryptography to facilitate deployment and testing in real world contexts.

  - OQS provides prototype integrations of liboqs into TLS and SSH, through OpenSSL and OpenSSH.

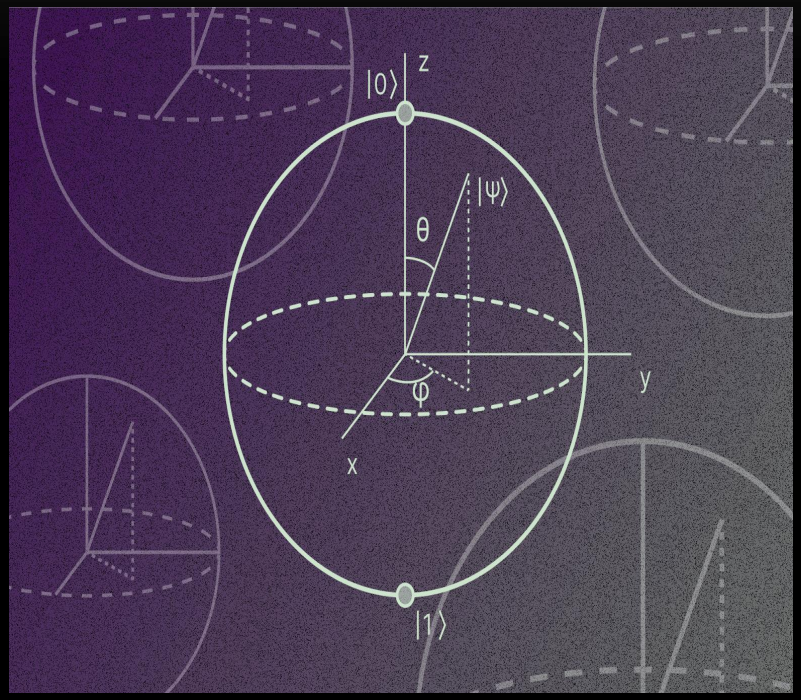
BUT WAIT THERE IS LOTS OF COMPETITION

- Key Encapsulation Mechanisms
  - **BIKE**: BIKE-L1, BIKE-L3, BIKE-L5
  - **Classic McEliece**: Classic-McEliece-348864†, Classic-McEliece-348864f†, Classic-McEliece-460896†, Classic-McEliece-460896f†, Classic-McEliece-6688128†, Classic-McEliece-6688128f†, Classic-McEliece-6960119†, Classic-McEliece-6960119f†, Classic-McEliece-8192128†, Classic-McEliece-8192128f†
  - **FrodoKEM**: FrodoKEM-640-AES, FrodoKEM-640-SHAKE, FrodoKEM-976-AES, FrodoKEM-976-SHAKE, FrodoKEM-1344-AES, FrodoKEM-1344-SHAKE
  - **HQC**: HQC-128, HQC-192, HQC-256†
  - **Kyber**: Kyber512, Kyber768, Kyber1024
  - **NTRU-Prime**: sntrup761

- Signature Schemes
  - **CRYSTALS-Dilithium**: Dilithium2, Dilithium3, Dilithium5
  - **Falcon**: Falcon-512, Falcon-1024
  - **SPHINCS+-SHA2**: SPHINCS+-SHA2-128f-simple, SPHINCS+-SHA2-128s-simple, SPHINCS+-SHA2-192f-simple, SPHINCS+-SHA2-192s-simple, SPHINCS+-SHA2-256f-simple, SPHINCS+-SHA2-256s-simple
  - **SPHINCS+-SHAKE**: SPHINCS+-SHAKE-128f-simple, SPHINCS+-SHAKE-128s-simple, SPHINCS+-SHAKE-192f-simple, SPHINCS+-SHAKE-192s-simple, SPHINCS+-SHAKE-256f-simple, SPHINCS+-SHAKE-256s-simple

QUANTUM RESISTANCE AND THE SIGNAL PROTOCOL "POST-QUANTUM EXTENDED DIFFIE-HELLMAN" (PQXDH)

- The Signal Protocol is a set of cryptographic specifications that provides end-to-end encryption for private communications exchanged daily by billions of people around the world.
- The first step in advancing quantum resistance for the Signal Protocol is an upgrade to the X3DH specification which we are calling PQXDH. With this upgrade, we are adding a layer of protection against the threat of a quantum computer being built in the future that is powerful enough to break current encryption standards.
- PQXDH establishes a shared secret key between two parties who mutually authenticate each other based on public keys. PQXDH provides post-quantum forward secrecy and a form of cryptographic deniability but still relies on the hardness of the discrete log problem for mutual authentication in this revision of the protocol.
- PQXDH is designed for asynchronous settings where one user ("Bob") is offline but has published some information to a server. Another user ("Alice") wants to use that information to send encrypted data to Bob, and also establish a shared secret key for future communication.



- Source: https://signal.org/docs/specifications/pqxdh/

SIGNAL PQXDH - SECURITY CONSIDERATIONS

- **Authentication**
  - Before or after a PQXDH key agreement, the parties may compare their identity public keys through some authenticated channel.
  - **Authentication in PQXDH is not quantum-secure.**
  - In the presence of an active quantum adversary, the parties receive no cryptographic guarantees as to who they are communicating with.
  - Post-quantum secure deniable mutual authentication is an open research problem.
- **Protocol Replay**
  - If Alice's initial message doesn't use a one-time prekey, it may be replayed to Bob and he will accept it. This could cause Bob to think Alice had sent him the same message (or messages) repeatedly.
- **Replay and Key Reuse**
  - a successfully replayed initial message would cause Bob to derive the same *SK* in different protocol runs.
- **Deniability**
  - Introduce a notion of 1-out-of-2 deniability for semi-honest parties and a "big brother" judge with access to all parties' secret keys. Since either Alice or Bob can create a fake transcript using only their own secret keys, PQXDH has this deniability property.

SIGNAL PQXDH - SECURITY CONSIDERATIONS

- **Signatures**
  - It might be tempting to omit the prekey signature after observing that mutual authentication and forward secrecy are achieved by the *DH* calculations. However, this would allow a "weak forward secrecy" attack:
- **Key Compromise**
  - Compromise of a party's private keys has a disastrous effect on security, though the use of ephemeral keys and prekeys provides some mitigation.
- **Identity Binding**
  - Authentication does not necessarily prevent an "identity misbinding" or "unknown key share" attack.
- **Risks of weak randomness sources**
  - In addition to concerns about the generation of the keys themselves, the security of the PQKEM shared secret relies on the random source available to Alice's machine

SIGNAL PQXDH - QUANTUM ADVERSARIES

Passive

- PQXDH is designed to prevent "harvest now, decrypt later" attacks by adversaries with access to a quantum computer capable of computing discrete logarithms in *curve*.
- If an attacker has recorded the public information and the message from Alice to Bob, even access to a quantum computer will not compromise.
- If a post-quantum key encapsulation one-time prekey is used for a protocol run and deleted as specified then compromise after deletion and access to a quantum computer at some future time will not compromise the older *key*
- If post-quantum one-time prekeys were not used for a protocol run, then access to a quantum computer and a compromise of the private key for $PQSPK_B$ from that protocol run would compromise the *SK* that was calculated earlier. Frequent replacement of signed prekeys mitigates this, as does using a post-PQXDH ratcheting protocol which rapidly replaces with new keys to provide fresh forward secrecy.

Active

- PQXDH is not designed to provide protection against active quantum attackers.
- An active attacker with access to a quantum computer capable of computing discrete logarithms in *curve* can compute $DH(PK_1, PK_2)$ and $Sig(PK, M, Z)$ for all elliptic *curve* keys PK_1, PK_2, and PK. This allows an attacker to impersonate Alice by using the quantum computer to compute the secret key corresponding to PK_A then continuing with the protocol. A malicious server with access to such a quantum computer could impersonate Bob by generating new key pairs $PQSPK'_B$ and $PQOPK'_B$, computing the secret key corresponding to PK_B, then using PK_B to sign the newly generated post-quantum KEM keys and delivering these attacker-generated keys in place of Bob's post-quantum KEM key when Alice requests a prekey bundle.

CHROME SUPPORT

- Google has been working with the security community for over a decade to explore options for PQC algorithms beyond theoretical implementations.
- In a 2016 experiment in Chrome where a small fraction of connections between desktop Chrome and Google's servers used a post-quantum key-exchange algorithm, in addition to the elliptic-curve key-exchange algorithm that would typically be used. By adding a post-quantum algorithm in a hybrid mode with the existing key-exchange, we were able to test its implementation without affecting user security.
- Google then took this work further in 2019 and announced a wide-scale post-quantum experiment with **Cloudflare**. They worked together to implement two post-quantum key exchanges, integrated them into Cloudflare's TLS stack, and deployed the implementation on edge servers and in Chrome Canary clients. Through this work, they learned more about the performance and feasibility of deployment in TLS of two post-quantum key agreements, and have continued to integrate these learnings into our technology roadmap.
- In 2021, they tested broader deployment of post-quantum confidentiality in TLS and discovered a range of network products that were incompatible with post-quantum TLS. We were able to work with the vendor so that the issue was fixed in future firmware updates. By experimenting early, we resolved this issue for future deployments.
- In 2023 Cloudflare announced general availability of PQC support.

FINAL THOUGHTS

- Conduct a quantum risk assessment and plan to implement quantum-resistant cryptography.
- Don’t wait for perfect cryptographic implementations – they are better than what we are using today.
- Don’t wait for commercial availability of quantum computers – the emerging model is to use a shared “Quantum-as-a-Service” model with manufacturers.
- Build for tomorrow.

REFERENCES

- A Methodology for Quantum Risk Assessment
- CSA: The State of Post-Quantum Cryptography
- GitHub Quantum Resources
- NIST: Post-Quantum Cryptography Standardization
- Post-Quantum Cryptography Lounge
- Quantum Computation and Shor's Factoring Algorithm
- The Sounds of IBM: IBM Q
- Thirty Years Later, a Speed Boost for Quantum Factoring
- Post-quantum cryptography Algorithm's standardization and performance analysis

THANK YOU!