

Introduction to the Era of Post-Quantum Cryptography

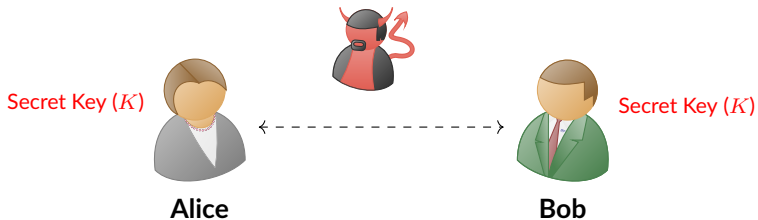
Dr. Essam Ghadafi

The lecture maps to the following CyBOK Knowledge Areas:

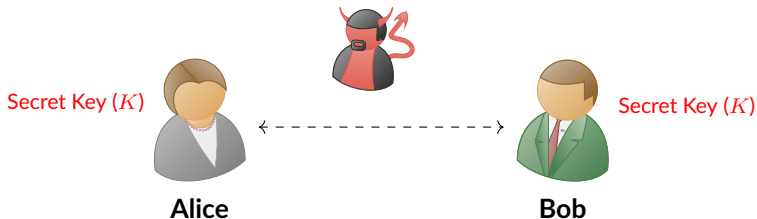
- Systems Security → Cryptography
- Infrastructure Security → Applied Cryptography

- **Classical Cryptography:** Overview of symmetric and asymmetric cryptography
- **Shor's and Grover's Algorithms:** Impact of quantum algorithms on cryptography
- **Introduction to Quantum Computing:** Qubits, superposition, and entanglement
- **The Issue(s) in Classical Cryptography:** Vulnerabilities of classical systems to quantum attacks
- **Post-Quantum Cryptography Approaches:** Brief overview of quantum-resistant cryptographic methods
- **Challenges in Post-Quantum Cryptography:** Efficiency trade-offs, key sizes, and security proofs

SECRET-KEY CRYPTOGRAPHY



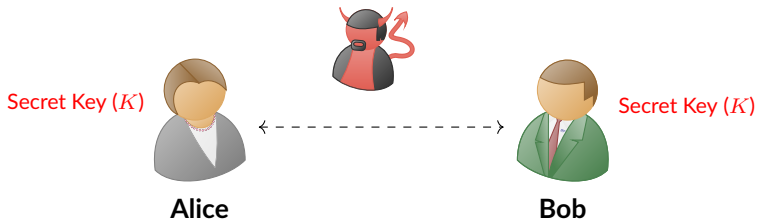
SECRET-KEY CRYPTOGRAPHY



Communicating parties must share a secret key

- Requires solution to key-distribution problem

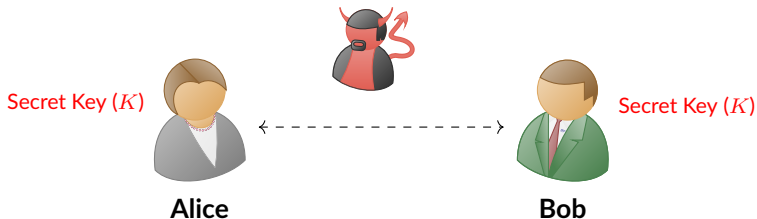
SECRET-KEY CRYPTOGRAPHY



Communicating parties must share a secret key

- Requires solution to key-distribution problem
- How can Alice send a message that only Bob can read?

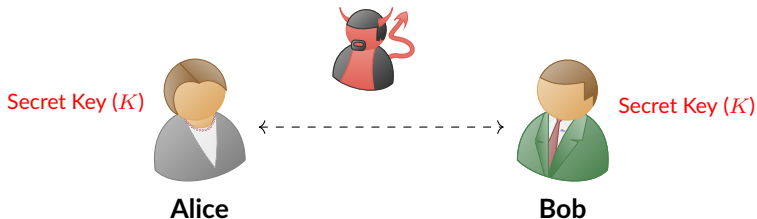
SECRET-KEY CRYPTOGRAPHY



Communicating parties must share a secret key

- Requires solution to key-distribution problem
- How can Alice send a message that only Bob can read? Use Secret-Key Encryption

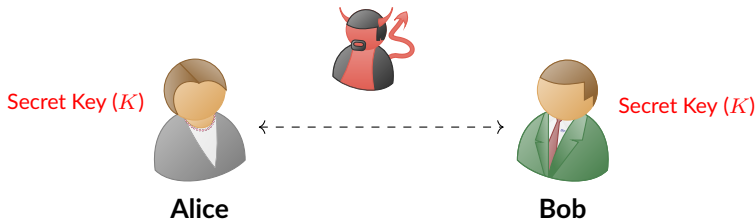
SECRET-KEY CRYPTOGRAPHY



Communicating parties must share a secret key

- Requires solution to key-distribution problem
- How can Bob send a message that only Bob could have sent?

SECRET-KEY CRYPTOGRAPHY

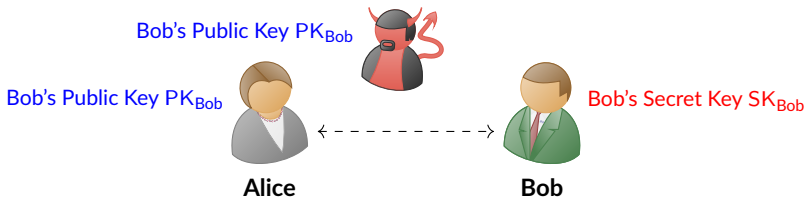


Communicating parties must share a secret key

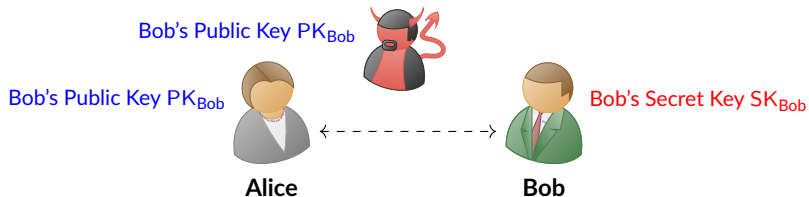
- Requires solution to key-distribution problem

- How can Bob send a message that only Bob could have sent?
Use MACs

PUBLIC-KEY CRYPTOGRAPHY



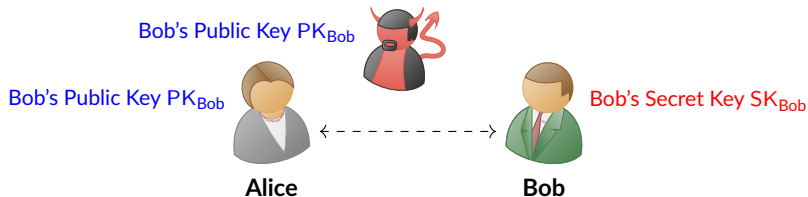
PUBLIC-KEY CRYPTOGRAPHY



Everyone knows PK_{Bob} but only Bob knows SK_{Bob}

- How is this done in practice?

PUBLIC-KEY CRYPTOGRAPHY



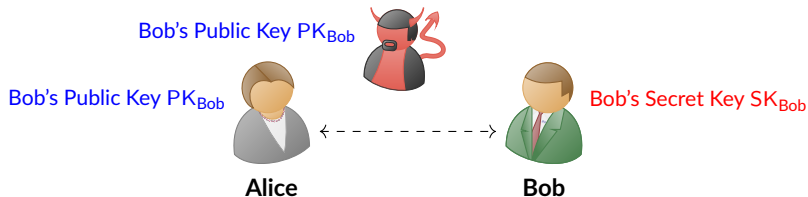
Everyone knows PK_{Bob} but only Bob knows SK_{Bob}

- How is this done in practice?

Goals:

- How can Alice send a message that only Bob can read?

PUBLIC-KEY CRYPTOGRAPHY



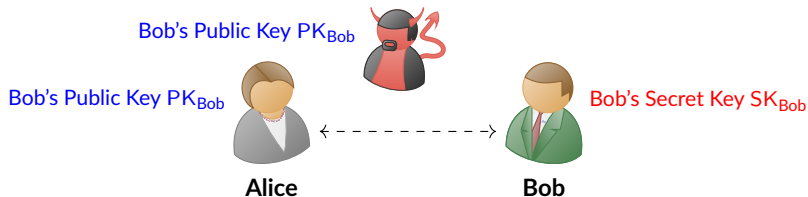
Everyone knows PK_{Bob} but only Bob knows SK_{Bob}

- How is this done in practice?

Goals:

- How can Alice send a message that only Bob can read?
Use Public-Key Encryption

PUBLIC-KEY CRYPTOGRAPHY



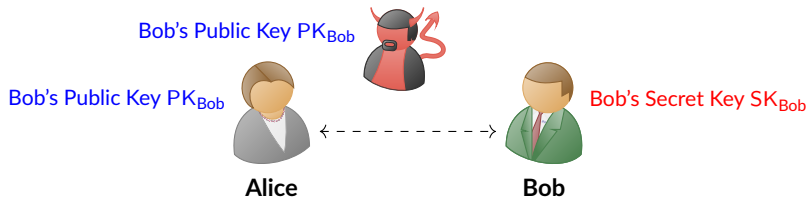
Everyone knows PK_{Bob} but only Bob knows SK_{Bob}

- How is this done in practice?

Goals:

- How can Bob send a message that only Bob could have sent?

PUBLIC-KEY CRYPTOGRAPHY



Everyone knows PK_{Bob} but only Bob knows SK_{Bob}

- How is this done in practice?

Goals:

- How can Bob send a message that only Bob could have sent? Use Digital Signatures

PROVABLE SECURITY (CLASSIC CRYPTO)

Security Proof

=

Security Requirement $\mathcal{R}$
e.g. EUF-CMA, IND-CCA

PROVABLE SECURITY (CLASSIC CRYPTO)

Security Proof

=

Security Requirement $\mathcal{R}$
e.g. EUF-CMA, IND-CCA

+

Hard Problem $\mathcal{P}$
e.g. Factoring, DLog

PROVABLE SECURITY (CLASSIC CRYPTO)

Security Proof

=

Security Requirement $\mathcal{R}$
e.g. EUF-CMA, IND-CCA

+

Hard Problem $\mathcal{P}$
e.g. Factoring, DLog

+

A reduction from $\mathcal{R}$ to $\mathcal{P}$

PROVABLE SECURITY (CLASSIC CRYPTO)

Security Proof

=

Security Requirement $\mathcal{R}$
e.g. EUF-CMA, IND-CCA

+

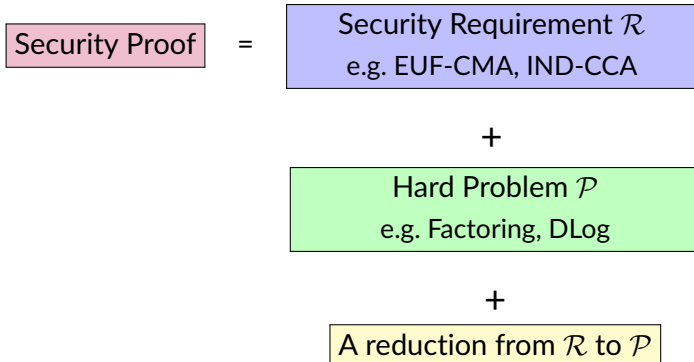
Hard Problem $\mathcal{P}$
e.g. Factoring, DLog

+

A reduction from $\mathcal{R}$ to $\mathcal{P}$

If attacker violates requirement $\mathcal{R}$, we solve problem $\mathcal{P}$

PROVABLE SECURITY (CLASSIC CRYPTO)



If attacker violates requirement $\mathcal{R}$, we solve problem $\mathcal{P}$

The Issue: Only *classic* attackers have been considered against $\mathcal{P}$

- How about Quantum attackers?

QUANTUM COMPUTERS

- Exploit quantum phenomena,
e.g. superposition &
entanglement



QUANTUM COMPUTERS

- Exploit quantum phenomena, e.g. superposition & entanglement
- Utilise different rules than classic computers, e.g. qubits



QUANTUM COMPUTERS

- Exploit quantum phenomena, e.g. superposition & entanglement
- Utilise different rules than classic computers, e.g. qubits
- A lot of attention and advancement in recent years



QUANTUM COMPUTERS

- Exploit quantum phenomena, e.g. superposition & entanglement
- Utilise different rules than classic computers, e.g. qubits
- A lot of attention and advancement in recent years
- A lot of applications: Genomic sequencing, finance, etc.
 - How do they affect used Cryptography?



CLASSIC BIT VS. QUANTUM BIT

The state of a classic bit can be either 0 or 1

CLASSIC BIT VS. QUANTUM BIT

The state of a quantum bit (qubit) is a complex unit vector

$$\alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$ and $\alpha, \beta \in \mathbb{C}$

CLASSIC BIT VS. QUANTUM BIT

The state of a quantum bit (qubit) is a complex unit vector

$$\alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$ and $\alpha, \beta \in \mathbb{C}$

We can think of a classic bit as additionally requiring that $|\alpha|^2 = 0$ or $|\alpha|^2 = 1$

CLASSIC BIT VS. QUANTUM BIT

The state of a quantum bit (qubit) is a complex unit vector

$$\alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$ and $\alpha, \beta \in \mathbb{C}$

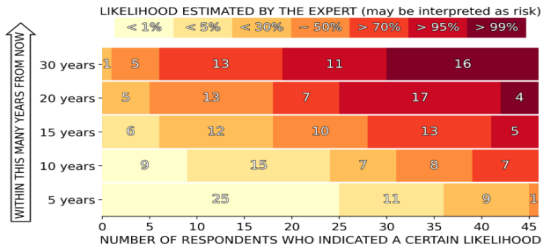
We can think of a classic bit as additionally requiring that $|\alpha|^2 = 0$ or $|\alpha|^2 = 1$

When measuring a qubit, it has probability $|\alpha|^2$ of being $|0\rangle$ and probability $|\beta|^2$ of being $|1\rangle$

- After the measurement, the system is in the measured state

EXPERT'S OPINIONS

In 2021, experts were asked about the likelihood of a quantum computer breaking 2048-bit RSA



[<https://globalriskinstitute.org/publications/2021-quantum-threat-timeline-report>]

WHY ACT NOW?

Why should we care, even if large-scale quantum computers are not yet a reality?

WHY ACT NOW?

Why should we care, even if large-scale quantum computers are not yet a reality?

- **Quantum Threats Loom:** Quantum computers will eventually break current cryptography
- **SNDL (Store Now, Decrypt Later):** Today's encrypted data may be stored now and decrypted in the future
- **Slow Deployment:** New cryptographic standards take years to be deploy at scale

- 1994: Shor's algorithm
 - Breaks DLog & factoring with poly many gates and depth

- 1996: Grover's algorithm
 - Quadratic speed up for search problems, i.e. from $\mathcal{O}(N)$ to $\mathcal{O}(\sqrt{N})$. Applicable to exhaustive key search (brute-forcing) and hash collisions

THE BASIC PROBLEM

Some of today's Cryptography relies mostly on *hard* problems which do not hold against quantum attackers

THE BASIC PROBLEM

Some of today's Cryptography relies mostly on *hard* problems which do not hold against quantum attackers, e.g.:

- Hardness of factoring large (e.g. 2048-bit) numbers, e.g. RSA encryption, RSA signatures

Some of today's Cryptography relies mostly on *hard* problems which do not hold against quantum attackers, e.g.:

- Hardness of computing discrete logarithms (and related assumptions), e.g DSA, Diffie-Hellman, ElGamal

CLASSICAL HARD PROBLEMS

Factoring Problem

Input: $n = p * q$ for some large primes p and q

Task: Factor n to find p and q

CLASSICAL HARD PROBLEMS

Factoring Problem

Input: $n = p * q$ for some large primes p and q

Task: Factor n to find p and q

Has implications for many other assumptions, e.g. RSA, Strong RSA, etc.

CLASSICAL HARD PROBLEMS

Factoring Problem

Input: $n = p * q$ for some large primes p and q

Task: Factor n to find p and q

Has implications for many other assumptions, e.g. RSA, Strong RSA, etc.

Quantum hardness of factoring is $\text{poly}(\log n)$

Discrete Logarithm (DLog) Problem

Input: Group $\mathbb{G} = \langle g \rangle$ of order p , elements g and $X = g^x$

Task: Find $x \in \mathbb{Z}_p$

Discrete Logarithm (DLog) Problem

Input: Group $\mathbb{G} = \langle g \rangle$ of order p , elements g and $X = g^x$

Task: Find $x \in \mathbb{Z}_p$

Has implications for many other assumptions, e.g. CDH, DDH, q -SDH, etc.

Discrete Logarithm (DLog) Problem

Input: Group $\mathbb{G} = \langle g \rangle$ of order p , elements g and $X = g^x$

Task: Find $x \in \mathbb{Z}_p$

Has implications for many other assumptions, e.g. CDH, DDH, q -SDH, etc.

Quantum hardness of DLog is $\text{poly}(\log p)$

SHOR'S ALGORITHM & FACTORING

- Factoring $\leq$ Order-Finding
 - Factoring reduces to order-finding
 - ▶ An efficient algorithm to find the order of elements modulo $N \Rightarrow$ we can factor N efficiently
- Order-Finding $\approx$ Period-Finding
 - Order-finding and period-finding are approximately equivalent

SHOR'S ALGORITHM

Steps of Shor's Algorithm:

- 1 Choose a random number a

Steps of Shor's Algorithm:

- 1 Choose a random number a
- 2 Compute $\gcd(a, N)$. If it's non-trivial, we are done

SHOR'S ALGORITHM

Steps of Shor's Algorithm:

- ① Choose a random number a
- ② Compute $\gcd(a, N)$. If it's non-trivial, we are done
- ③ Find the period r of $f(x) = a^x \bmod N$ using Quantum Fourier Transform (QFT)
 - r is the order of $a \bmod N$, i.e. smallest $r > 1$ s.t. $a^r \equiv 1 \bmod N$

This step can be efficiently computed in time $O((\log N)^2 \log \log N)$ on a quantum computer

SHOR'S ALGORITHM

Steps of Shor's Algorithm:

- ① Choose a random number a
- ② Compute $\gcd(a, N)$. If it's non-trivial, we are done
- ③ Find the period r of $f(x) = a^x \bmod N$ using Quantum Fourier Transform (QFT)
 - r is the order of $a \bmod N$, i.e. smallest $r > 1$ s.t. $a^r \equiv 1 \bmod N$
- ④ Compute $\gcd(a^{\frac{r}{2}} - 1, N)$ to get a non-trivial factor

This step can be efficiently computed in time $O((\log N)^2 \log \log N)$ on a quantum computer

SHOR'S ALGORITHM – EXAMPLE

Example: Factoring 21:

- Choose $a = 2$, compute powers: $2^x \bmod 21$
- Sequence: 2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, ... (Period $r = 6$)
- Compute $\gcd(2^{\frac{6}{2}} - 1, 21) = \gcd(7, 21) = 7$
- Factors: 3, 7, thus $21 = 3 \times 7$

Factoring is easy to solve if we can find the period r efficiently

What does it do?

- Quantum search algorithm
- Finds a marked item in an unsorted database of size N in $O(\sqrt{N})$
- Quadratic speedup over classical brute-force search ($O(\sqrt{N})$ vs. $O(N)$)

POST-QUANTUM (PQ) SECURITY

Security Proof

=

PQ Security Requirement $\mathcal{R}$
e.g. EUF-CMA-PQ

POST-QUANTUM (PQ) SECURITY

Security Proof

=

PQ Security Requirement $\mathcal{R}$
e.g. EUF-CMA-PQ

+

PQ Hard Problem $\mathcal{P}$
e.g. LWE, SIS

POST-QUANTUM (PQ) SECURITY

Security Proof

=

PQ Security Requirement $\mathcal{R}$
e.g. EUF-CMA-PQ

+

PQ Hard Problem $\mathcal{P}$
e.g. LWE, SIS

+

A reduction from $\mathcal{R}$ to $\mathcal{P}$

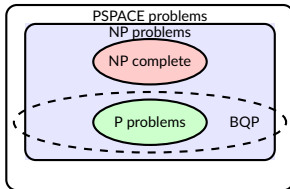
POST-QUANTUM (PQ) SECURITY

$$\begin{array}{c} \boxed{\text{Security Proof}} = \boxed{\begin{array}{c} \text{PQ Security Requirement } \mathcal{R} \\ \text{e.g. EUF-CMA-PQ} \end{array}} \\ \\ + \\ \\ \boxed{\begin{array}{c} \text{PQ Hard Problem } \mathcal{P} \\ \text{e.g. LWE, SIS} \end{array}} \\ \\ + \\ \\ \boxed{\text{A reduction from } \mathcal{R} \text{ to } \mathcal{P}} \end{array}$$

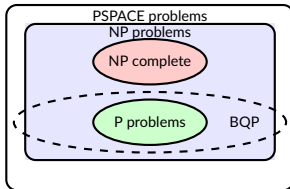
If attacker violates requirement $\mathcal{R}$, we solve problem $\mathcal{P}$.

Caters for quantum attackers

POST-QUANTUM HARD PROBLEMS

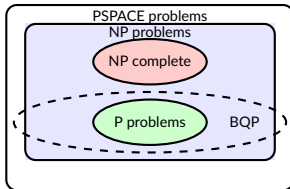


POST-QUANTUM HARD PROBLEMS



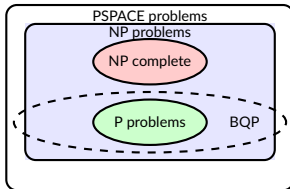
- **Class P:** Can be efficiently solved by a classical computer
Example: Primality Testing, Linear Programming, etc.

POST-QUANTUM HARD PROBLEMS



- **Class BQP:** Can be efficiently solved by a quantum computer
 - Factoring and DLog belong to this class

POST-QUANTUM HARD PROBLEMS



Some potential post-quantum candidates:

- Solving multivariate non-linear equations over a finite field
- Bounded distance decoding over finite fields
- Finding closest & shortest lattice vectors
- Breaking cryptographic hash functions

EXISTING/PREVIOUS STANDARDS

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

All of the above can be easily broken by a large scale quantum computer

A large scale quantum computer would also impact symmetric Cryptography , e.g. AES, SHA-3, but not by much

A large scale quantum computer would also impact symmetric Cryptography , e.g. AES, SHA-3, but not by much

- Quantum hardness of searching $\mathcal{X}$ is $\Theta(|\mathcal{X}|^{\frac{1}{2}})$ vs. $\mathcal{O}(|\mathcal{X}|)$ classic hardness

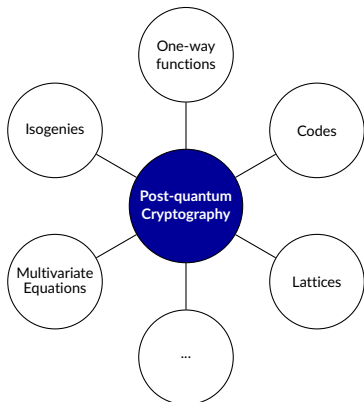
A large scale quantum computer would also impact symmetric Cryptography , e.g. AES, SHA-3, but not by much

- Quantum hardness of searching $\mathcal{X}$ is $\Theta(|\mathcal{X}|^{\frac{1}{2}})$ vs. $\mathcal{O}(|\mathcal{X}|)$ classic hardness
- Quantum hardness of finding collisions $\Theta(|\mathcal{X}|^{\frac{1}{3}})$ vs. $\mathcal{O}(|\mathcal{X}|^{\frac{1}{2}})$ classic hardness

A large scale quantum computer would also impact symmetric Cryptography , e.g. AES, SHA-3, but not by much

- Quantum hardness of searching $\mathcal{X}$ is $\Theta(|\mathcal{X}|^{\frac{1}{2}})$ vs. $\mathcal{O}(|\mathcal{X}|)$ classic hardness
- Quantum hardness of finding collisions $\Theta(|\mathcal{X}|^{\frac{1}{3}})$ vs. $\mathcal{O}(|\mathcal{X}|^{\frac{1}{2}})$ classic hardness

e.g. To get same security level of AES, double key size

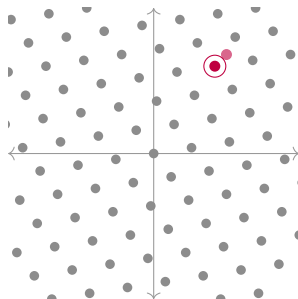
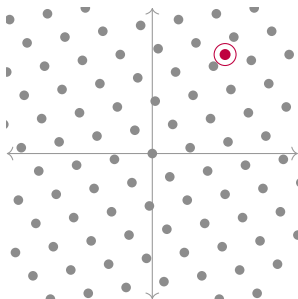


- Various intractability assumptions
- Sometimes not easy to compare like-for-like

A lattice is a set generated by integer linear combinations of the columns of a matrix

- **Key idea:** Use a **good** reduced basis as SK and a **bad** basis as PK
- Enabled the first realisation of fully homomorphic encryption

Example: Closest Vector Problem (CVP)



💡 **Intuition:** Find the closest lattice point to a given target

- Hard in high dimensions

MQ Problem

Input: Quadratic polynomials $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$ of degree ≤ 2

Task: Find $\mathbf{y} \in \mathbb{F}^n$ s.t. $f_i(\mathbf{y}) = 0$ for all $i = 1, \dots, m$

 **Intuition:** Solve a system of quadratic equations over a finite field

MQ Problem

Input: Quadratic polynomials $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$ of degree ≤ 2

Task: Find $\mathbf{y} \in \mathbb{F}^n$ s.t. $f_i(\mathbf{y}) = 0$ for all $i = 1, \dots, m$

 **Intuition:** Solve a system of quadratic equations over a finite field

- Even small systems are hard to solve efficiently
- Decisional MQ problem is NP-Complete
- Cryptosystems typically have large keys, but small signatures/ciphertexts (e.g., Rainbow signature scheme (**Broken!**))

Used mainly for digital signatures (e.g., Lamport scheme)

- Signature schemes based on hash functions
- Security reduces to finding collisions in the hash
- Large signatures and slower signing (SPHINCS+ signature scheme standardised by NIST)

Syndrome Decoding Problem

Input: A (binary) matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, syndrome $\mathbf{s} \in \mathbb{F}_2^n$

Task: Find $\mathbf{e} \in \mathbb{F}_2^m$ of small Hamming weight s.t. $\mathbf{A}\mathbf{e} = \mathbf{s}$

 **Intuition:** Solve a linear system over $\mathbb{F}_2$, constrained to a sparse (low-weight) solution

Syndrome Decoding Problem

Input: A (binary) matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, syndrome $\mathbf{s} \in \mathbb{F}_2^n$

Task: Find $\mathbf{e} \in \mathbb{F}_2^m$ of small Hamming weight s.t. $\mathbf{A}\mathbf{e} = \mathbf{s}$

 **Intuition:** Solve a linear system over $\mathbb{F}_2$, constrained to a sparse (low-weight) solution

- Large keys (e.g., 220kB for 128-bit security in McEliece Encryption)
- Some schemes broken recently, but still practical for high security
- Modern variant: HQC (Hamming Quasi-Cyclic encryption), selected by NIST in 2025 for standardization

Isogeny Problem

Input: Two isogenous elliptic curves E_1, E_2

Task: Compute a map (isogeny) $\psi : E_1 \rightarrow E_2$

💡 **Intuition:** Find a secret transformation between two elliptic curves

- Easy to verify once known, but hard to compute without the secret

Isogeny Problem

Input: Two isogenous elliptic curves E_1, E_2

Task: Compute a map (isogeny) $\psi : E_1 \rightarrow E_2$

💡 **Intuition:** Find a secret transformation between two elliptic curves

- Easy to verify once known, but hard to compute without the secret
- Yield compact cryptosystems
- Some schemes broken recently (SIDH/SIKE), but conceptually promising

Based on classical primitives (block ciphers, hash functions, etc.)
with quantum-safe parameters

Example:

- AES with 256-bit key (resistant to Grover's algorithm)
- Hash-based constructions for signatures

Based on classical primitives (block ciphers, hash functions, etc.) with quantum-safe parameters

Example:

- AES with 256-bit key (resistant to Grover's algorithm)
- Hash-based constructions for signatures

Advantages: Efficient, simple, well-studied

- 2016: NIST called for quantum-resistant cryptographic algorithms for new public-key cryptography standards (Digital signatures, Encryption & Key-establishment)

- 2016: NIST called for quantum-resistant cryptographic algorithms for new public-key cryptography standards (Digital signatures, Encryption & Key-establishment)
- 2022: NIST announced chosen candidates:

- 2016: NIST called for quantum-resistant cryptographic algorithms for new public-key cryptography standards (Digital signatures, Encryption & Key-establishment)
- 2022: NIST announced chosen candidates:
 - Encryption & Key Establishment:
 - ▶ CRYSTALS-KYBER (Lattice-based)

- 2016: NIST called for quantum-resistant cryptographic algorithms for new public-key cryptography standards (Digital signatures, Encryption & Key-establishment)

- 2022: NIST announced chosen candidates:
 - Encryption & Key Establishment:
 - ▶ CRYSTALS-KYBER (Lattice-based)

 - Digital Signatures:
 - ▶ CRYSTALS-DILITHIUM (Lattice-based)
 - ▶ FALCON (Lattice-based)
 - ▶ SPHINCS+ (Hash-based)

2024: Finalized NIST standards published:

- **FIPS 203:** Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) – **CRYSTALS-KYBER**
- **FIPS 204:** Module-Lattice-Based Digital Signature Algorithm (ML-DSA) – **CRYSTALS-DILITHIUM**
- **FIPS 205:** Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) – **SPHINCS+**

■ Key Dates

- NIST issues a call for Additional Signatures published: 6 Sep, 2022
- Submission deadline: 1 Jun, 2023
- Round 1 candidates announced (40): 17 Jul, 2023
- Round 2 candidates announced (14): 24 Oct, 2024
- Deadline for updated Round 2 packages: 17 Jan, 2025

■ Key Dates

- NIST issues a call for Additional Signatures published: 6 Sep, 2022
- Submission deadline: 1 Jun, 2023
- Round 1 candidates announced (40): 17 Jul, 2023
- Round 2 candidates announced (14): 24 Oct, 2024
- Deadline for updated Round 2 packages: 17 Jan, 2025

Round 2 Candidates by Approach (Total: 14)

Approach	Number
MPC-in-the-Head (Zero-Knowledge based)	5
Multivariate	4
Code-based	2
Isogeny-based	1
Lattice-based	1
Symmetric-based	1

- HQC (Hamming Quasi Cyclic) is a code-based Key Encapsulation Mechanism
- Selected by NIST on 11 Mar, 2025 as the 5th PQC standard and backup to ML KEM
 - Provides algorithmic diversity beyond lattice assumptions
- Security relies on the Quasi Cyclic Syndrome Decoding problem
- Draft standard expected in 2026 and final standard in 2027

EFFICIENCY COMPARISON (128-BIT SECURITY)

Scheme	Signature Size	PK Size
RSA-3072	384	384
ECDSA-256	64	64
CRYSTALS-DILITHIUM	2,420	1,312
FALCON	666	897
SPHINCS+	17,088	32

Sizes are in bytes

SOME CHALLENGES IN PQC

- Larger keys, signatures, and ciphertexts
- Higher resource demands (time, hardware, memory)
- New operations and assumptions
 - Some assumptions (e.g., SIDH) have been broken
- Side-channel attacks
- More complex implementations
 - Non-uniform sampling, sampling rejection, decryption failures, etc.

KEY TAKEAWAYS

- **Shor's and Grover's Algorithms** threaten classical cryptosystems by efficiently solving hard problems
- **Post-Quantum Cryptography** aims to develop quantum-resistant cryptosystems to secure future communications
- **Challenges in PQC** include balancing efficiency and security, and some approaches are not as mature as their classical counterparts

ADDITIONAL RESOURCES & READING

- **NIST PQC Standardization Project:** <https://csrc.nist.gov/projects/post-quantum-cryptography>
- **Open Quantum Safe Project:**
<https://github.com/open-quantum-safe>
- **Shor's Paper:**
<https://arxiv.org/abs/quant-ph/9508027>