Foundations of Post-Quantum Cryptography

A 3-Hour Introduction for Quantum Technologies Students

Dr. José Ramón Martínez Saavedra

Quside Technologies SL

March 30, 2026

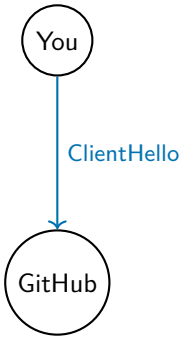
The Invisible Infrastructure

Live Demo

1. Open `https://github.com`
2. Click padlock icon
3. View certificate details

In 400 milliseconds...

- 3 digital signatures verified
- ECDHE key exchange performed
- 6 cryptographic keys derived
- AES-256-GCM encryption started



All of this will be broken by quantum computers!

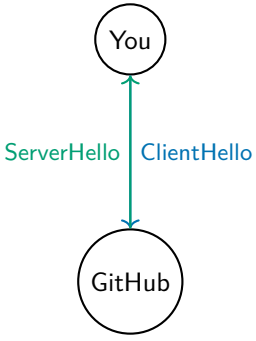
The Invisible Infrastructure

Live Demo

1. Open `https://github.com`
2. Click padlock icon
3. View certificate details

In 400 milliseconds...

- 3 digital signatures verified
- ECDHE key exchange performed
- 6 cryptographic keys derived
- AES-256-GCM encryption started



All of this will be broken by quantum computers!

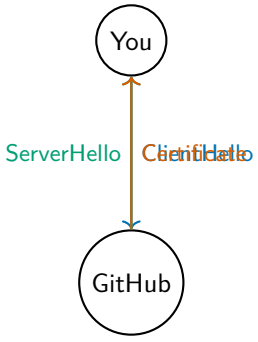
The Invisible Infrastructure

Live Demo

1. Open `https://github.com`
2. Click padlock icon
3. View certificate details

In 400 milliseconds...

- 3 digital signatures verified
- ECDHE key exchange performed
- 6 cryptographic keys derived
- AES-256-GCM encryption started



All of this will be broken by quantum computers!

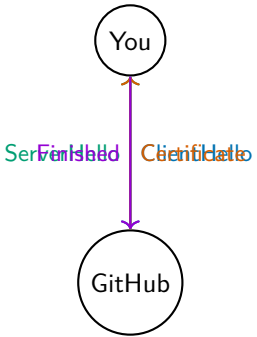
The Invisible Infrastructure

Live Demo

1. Open `https://github.com`
2. Click padlock icon
3. View certificate details

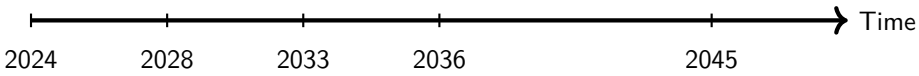
In 400 milliseconds...

- 3 digital signatures verified
- ECDHE key exchange performed
- 6 cryptographic keys derived
- AES-256-GCM encryption started

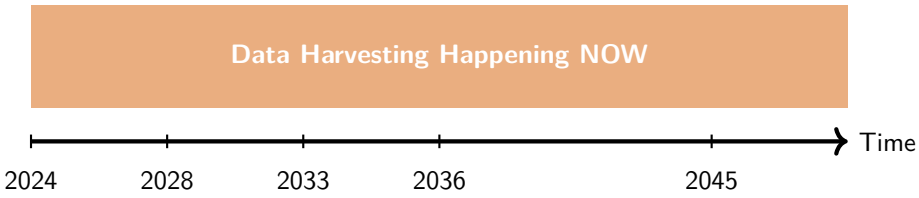


All of this will be broken by quantum computers!

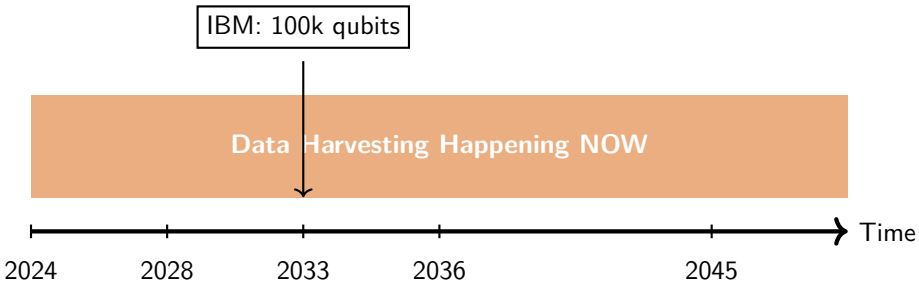
The Quantum Timeline



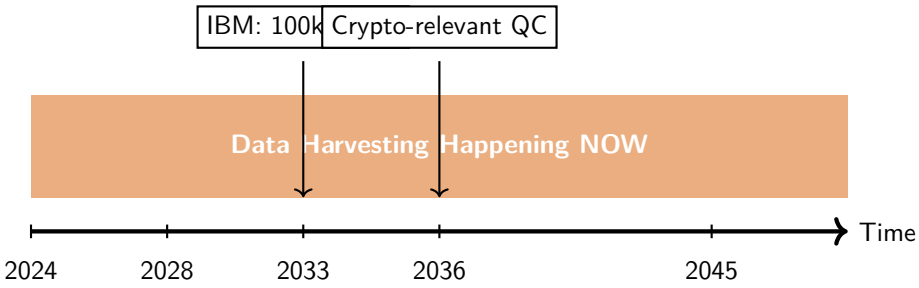
The Quantum Timeline



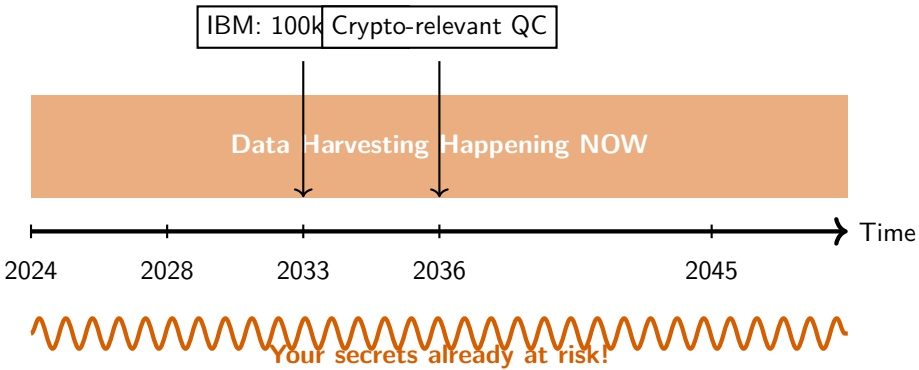
The Quantum Timeline



The Quantum Timeline



The Quantum Timeline



How do we protect information from computers that don't exist yet?

How do we protect information from computers that don't exist yet?

1. Foundation

How does current cryptography work?

How do we protect information from computers that don't exist yet?

1. Foundation

How does current cryptography work?

2. Threat

Why do quantum computers break it?

How do we protect information from computers that don't exist yet?

1. Foundation

How does current cryptography work?

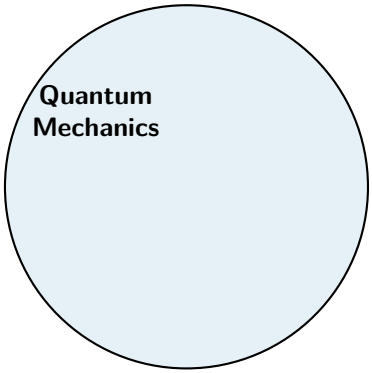
2. Threat

Why do quantum computers break it?

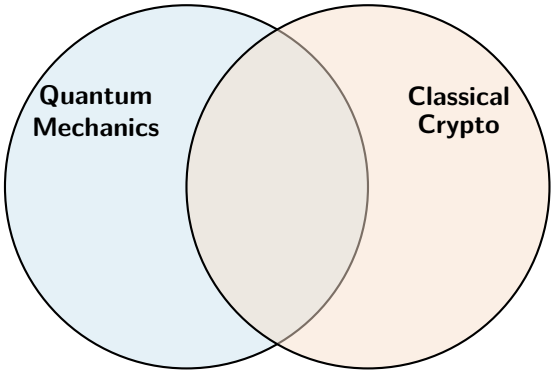
3. Solution

What mathematical structures survive?

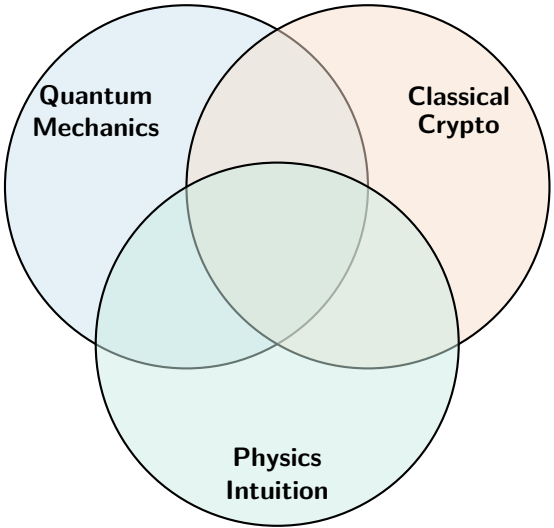
Why YOU Are Uniquely Positioned



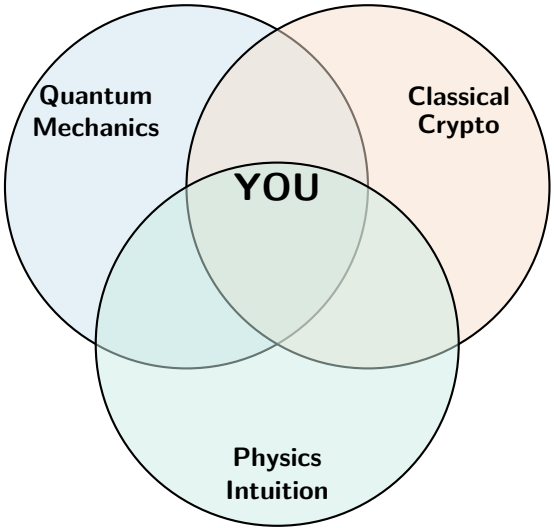
Why YOU Are Uniquely Positioned



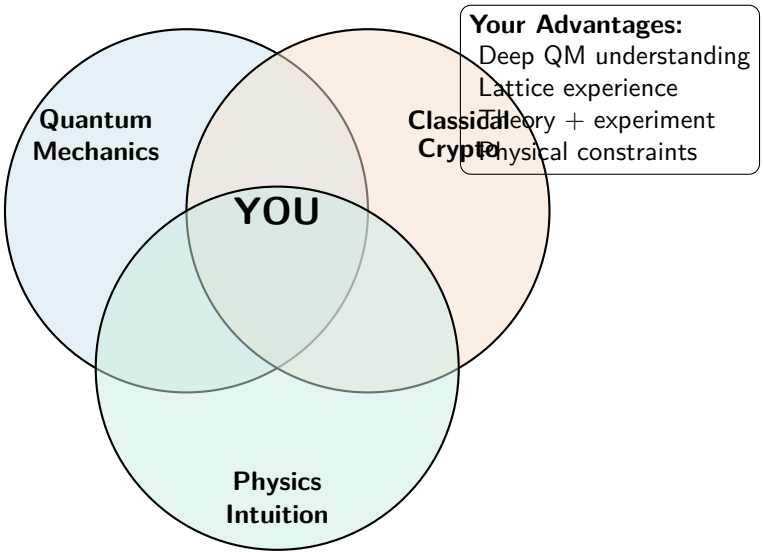
Why YOU Are Uniquely Positioned



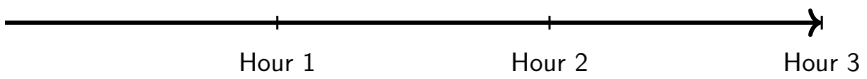
Why YOU Are Uniquely Positioned



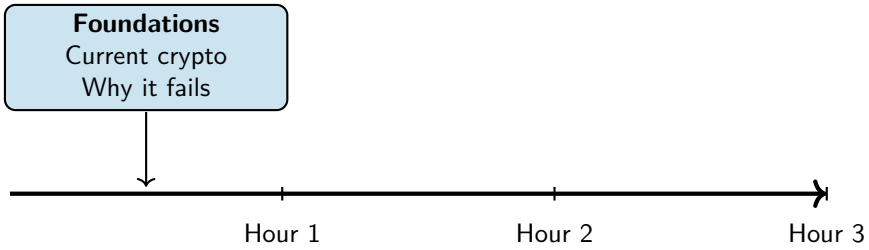
Why YOU Are Uniquely Positioned



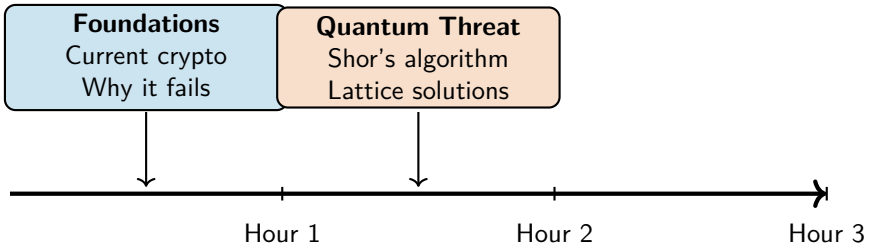
Our Three-Hour Journey



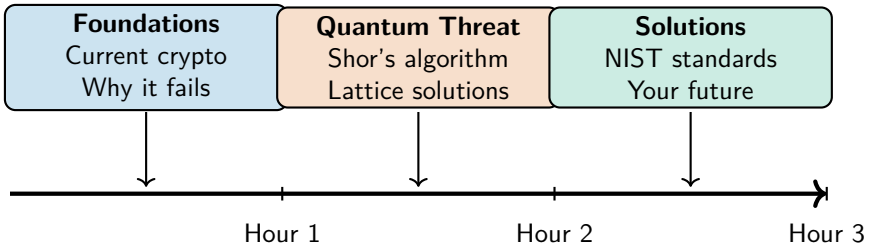
Our Three-Hour Journey



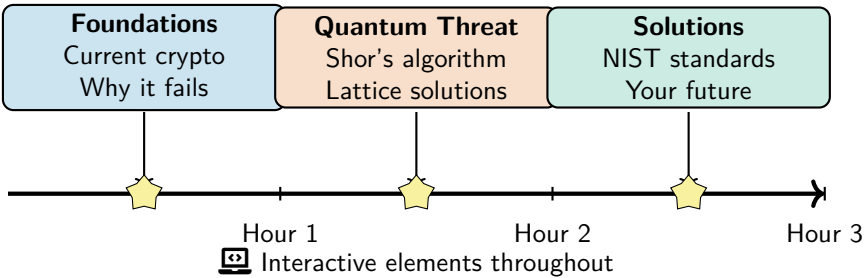
Our Three-Hour Journey



Our Three-Hour Journey



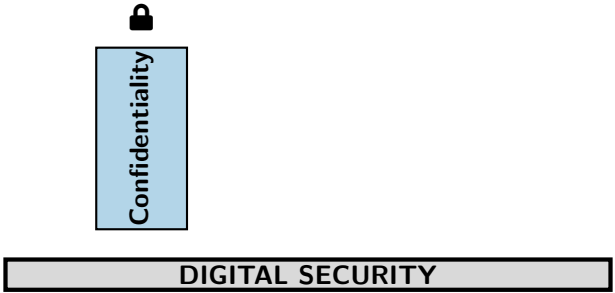
Our Three-Hour Journey



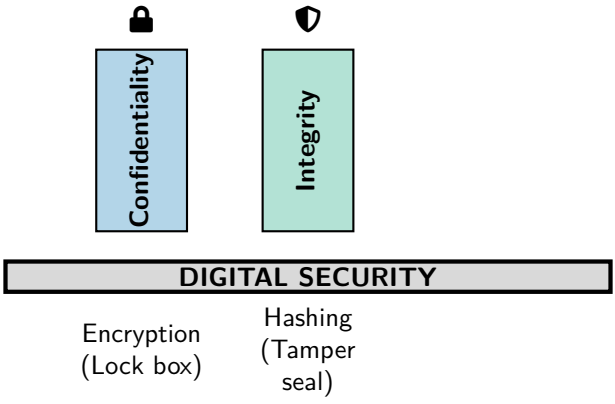
The Three Pillars of Security

DIGITAL SECURITY

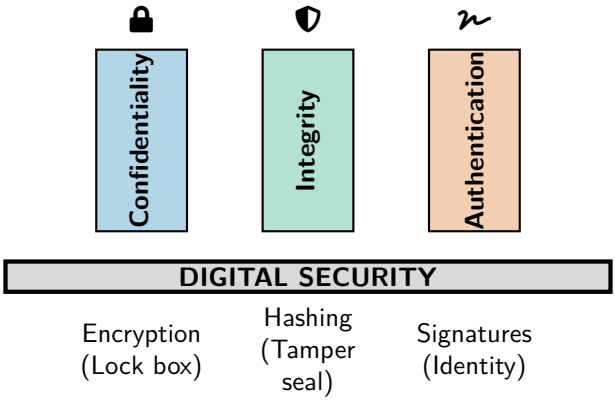
The Three Pillars of Security



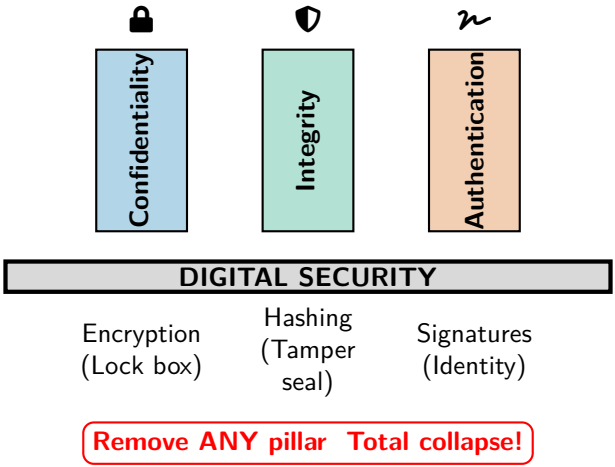
The Three Pillars of Security



The Three Pillars of Security



The Three Pillars of Security



Pillar 1: Confidentiality

`ATTACK AT DAWN`

What Encryption Provides

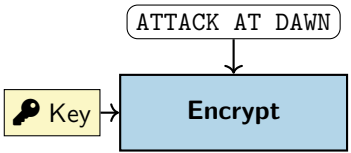
- Transforms readable unreadable
- Only key holder can reverse
- Examples: AES, RSA, ChaCha20

Critical Limitation

Encryption alone is NOT enough!

- No integrity guarantee
- No sender verification
- Vulnerable to tampering

Pillar 1: Confidentiality



What Encryption Provides

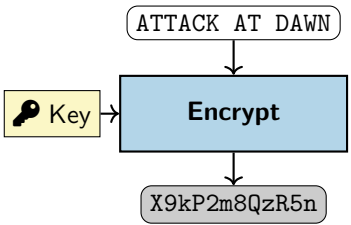
- Transforms readable unreadable
- Only key holder can reverse
- Examples: AES, RSA, ChaCha20

Critical Limitation

Encryption alone is NOT enough!

- No integrity guarantee
- No sender verification
- Vulnerable to tampering

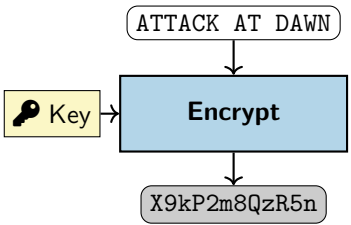
Pillar 1: Confidentiality



What Encryption Provides

- Transforms readable unreadable
- Only key holder can reverse
- Examples: AES, RSA, ChaCha20

Pillar 1: Confidentiality



What Encryption Provides

- Transforms readable unreadable
- Only key holder can reverse
- Examples: AES, RSA, ChaCha20

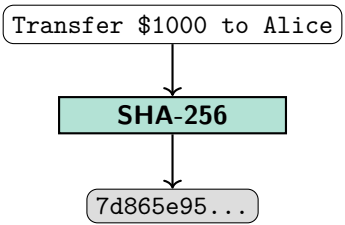
Critical Limitation

Encryption alone is NOT enough!

- No integrity guarantee
- No sender verification
- Vulnerable to tampering

2013: NSA vs Google Internal Traffic

Pillar 2: Integrity



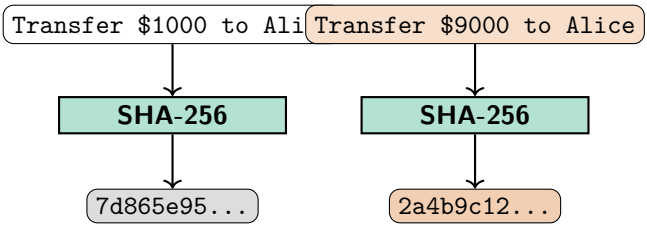
Hash Properties

- Fixed size output
- Deterministic
- Avalanche effect
- One-way function

Use Cases

- File verification
- Password storage
- Digital fingerprints
- Blockchain

Pillar 2: Integrity



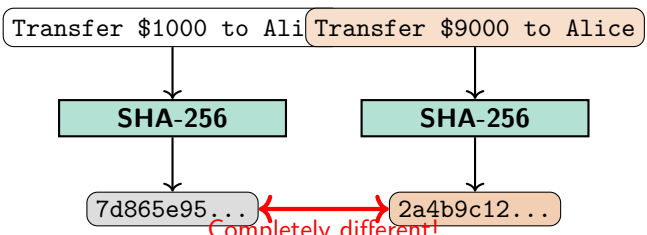
Hash Properties

- Fixed size output
- Deterministic
- Avalanche effect
- One-way function

Use Cases

- File verification
- Password storage
- Digital fingerprints
- Blockchain

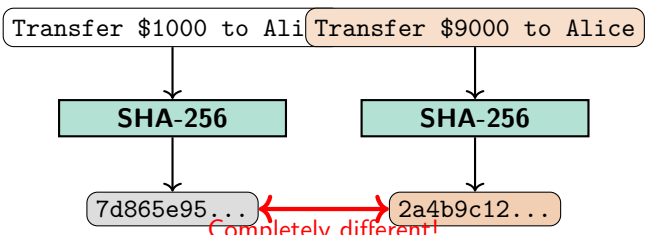
Pillar 2: Integrity



Hash Properties

- Fixed size output
- Deterministic
- Avalanche effect
- One-way function

Pillar 2: Integrity



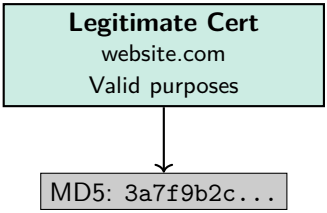
Hash Properties

- Fixed size output
- Deterministic
- Avalanche effect
- One-way function

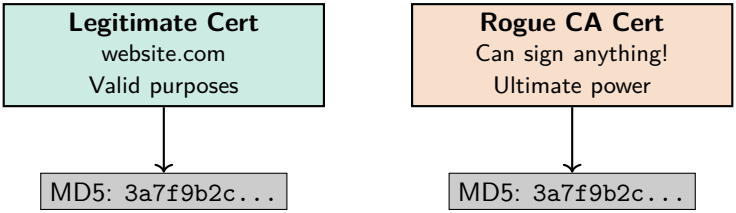
Use Cases

- File verification
- Password storage
- Digital fingerprints
- Blockchain

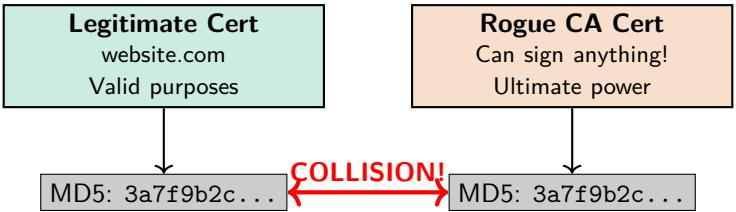
Case Study: MD5 Collision Disaster (2008)



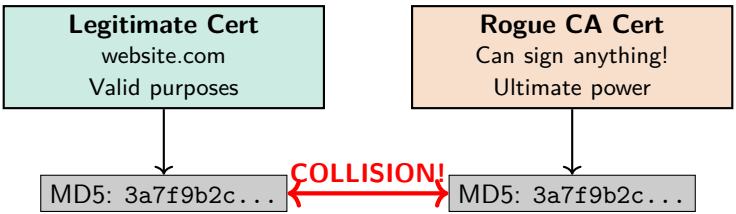
Case Study: MD5 Collision Disaster (2008)



Case Study: MD5 Collision Disaster (2008)

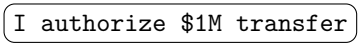


Case Study: MD5 Collision Disaster (2008)



Impact: Attackers can impersonate ANY website
Complete breakdown of web security

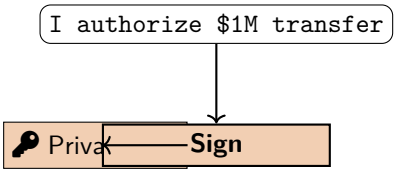
Pillar 3: Authentication



Properties:
Non-repudiation
Public verifiability
Unforgeable

Examples:
RSA signatures
ECDSA
EdDSA

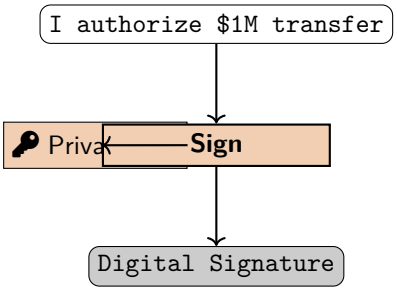
Pillar 3: Authentication



Properties:
Non-repudiation
Public verifiability
Unforgeable

Examples:
RSA signatures
ECDSA
EdDSA

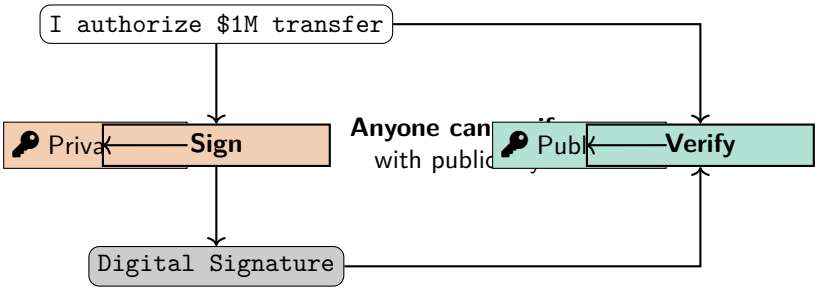
Pillar 3: Authentication



Properties:
Non-repudiation
Public verifiability
Unforgeable

Examples:
RSA signatures
ECDSA
EdDSA

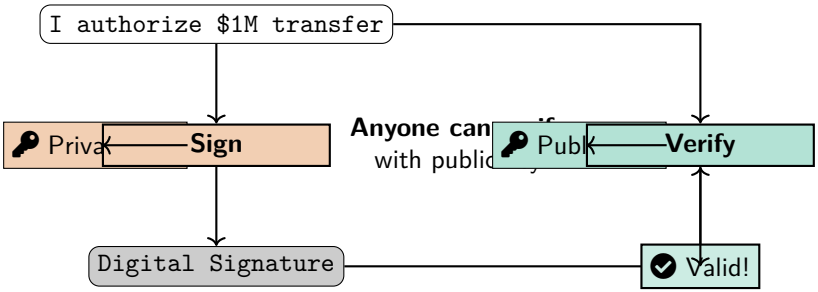
Pillar 3: Authentication



Properties:
Non-repudiation
Public verifiability
Unforgeable

Examples:
RSA signatures
ECDSA
EdDSA

Pillar 3: Authentication



Properties:
Non-repudiation
Public verifiability
Unforgeable

Examples:
RSA signatures
ECDSA
EdDSA

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Lesson

Authentication failure affects everyone who trusts the system

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Lesson

Authentication failure affects everyone who trusts the system

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Lesson

Authentication failure affects everyone who trusts the system

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Lesson

Authentication failure affects everyone who trusts the system

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Case Study: DigiNotar Catastrophe (2011)

Timeline of Disaster

June: Hackers breach DigiNotar

July: Issue *.google.com cert

August: Iran uses for MITM

Sept: Company bankrupt

Result: Dutch gov crisis

Impact Cascade

1. Fraudulent certificates trusted globally
2. Iranian citizens' Gmail compromised
3. All browsers revoke DigiNotar
4. Dutch government sites fail
5. Complete infrastructure rebuild

Lesson

Authentication failure affects everyone who trusts the system

The Circular Paradox

To communicate securely, you must first communicate securely!

The Circular Paradox

To communicate securely, you must first communicate securely!

499,500

keys! ;

The Circular Paradox

To communicate securely, you must first communicate securely!

499,500

keys! ;

The Revolutionary Insight (1976)

"We stand today on the brink of a revolution in cryptography."

– Diffie & Hellman

The Revolutionary Insight (1976)

> *"We stand today on the brink of a revolution in cryptography."*
>
> *– Diffie & Hellman*

The Breakthrough:
Separate the key into two parts with different capabilities!

The Revolutionary Insight (1976)

"We stand today on the brink of a revolution in cryptography."

– Diffie & Hellman

Private: x

The Breakthrough:
Separate the key into two parts with different capabilities!

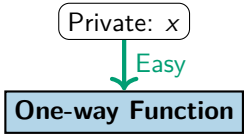
- Public key: Can encrypt
- Private key: Can decrypt

The Revolutionary Insight (1976)

> *"We stand today on the brink of a revolution in cryptography."*
>
> *– Diffie & Hellman*

The Breakthrough:
Separate the key into two parts with different capabilities!

- Public key: Can encrypt
- Private key: Can decrypt

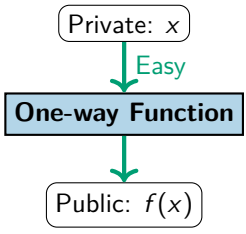


The Revolutionary Insight (1976)

> *"We stand today on the brink of a revolution in cryptography."*
>
> *– Diffie & Hellman*

The Breakthrough:
Separate the key into two parts with different capabilities!

- Public key: Can encrypt
- Private key: Can decrypt



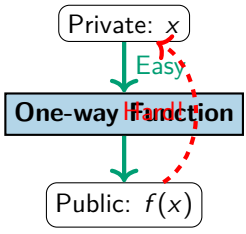
The Revolutionary Insight (1976)

"We stand today on the brink of a revolution in cryptography."

– Diffie & Hellman

The Breakthrough:
Separate the key into two parts with different capabilities!

- Public key: Can encrypt
- Private key: Can decrypt



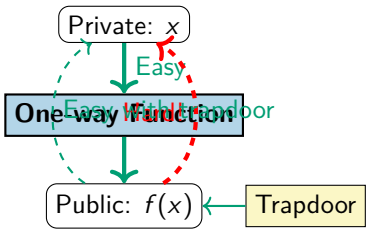
The Revolutionary Insight (1976)

"We stand today on the brink of a revolution in cryptography."

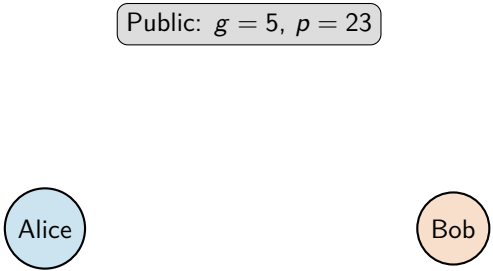
– Diffie & Hellman

The Breakthrough:
Separate the key into two parts with different capabilities!

- Public key: Can encrypt
- Private key: Can decrypt



Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange

Public: $g = 5$, $p = 23$

Alice

Secret: $a = 6$

Bob

Secret: $b = 15$

Diffie-Hellman Key Exchange

Public: $g = 5$, $p = 23$

Alice

Secret: $a = 6$

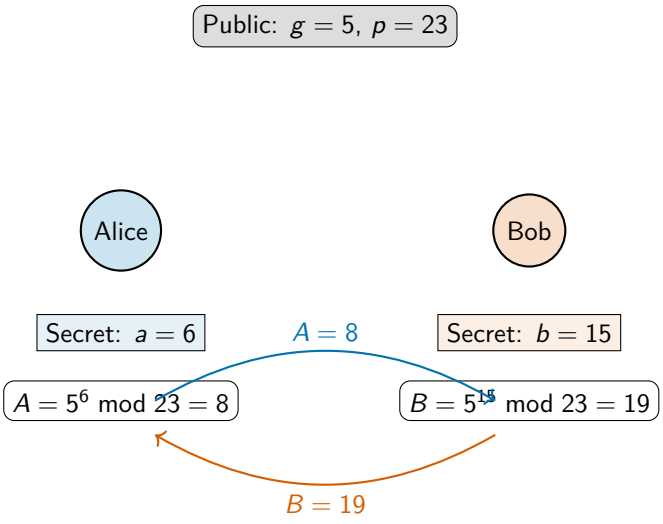
$A = 5^6 \bmod 23 = 8$

Bob

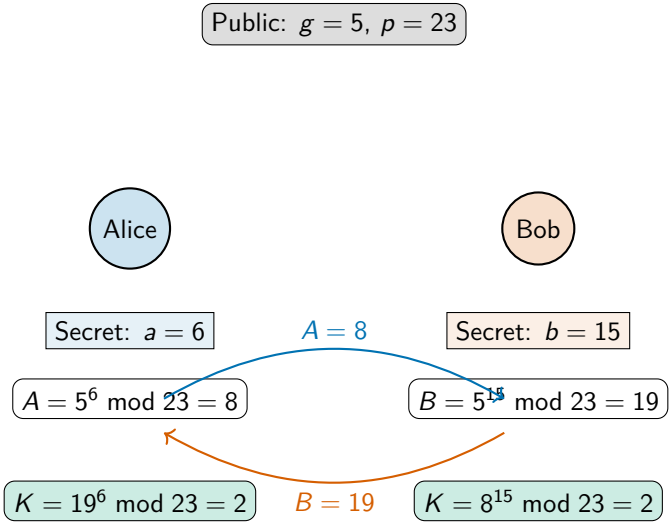
Secret: $b = 15$

$B = 5^{15} \bmod 23 = 19$

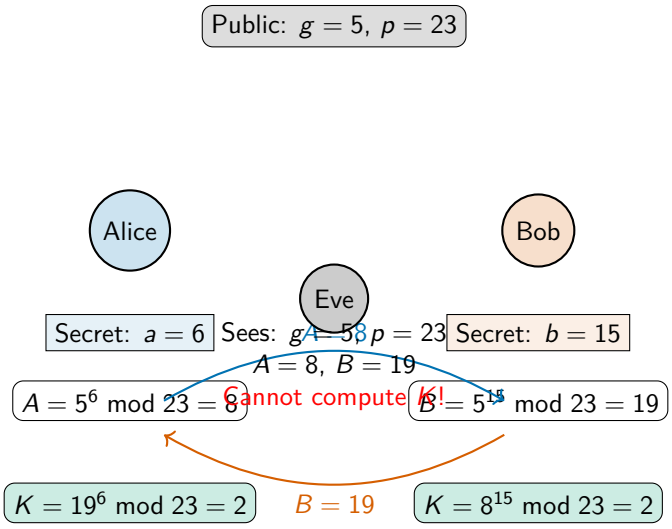
Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange



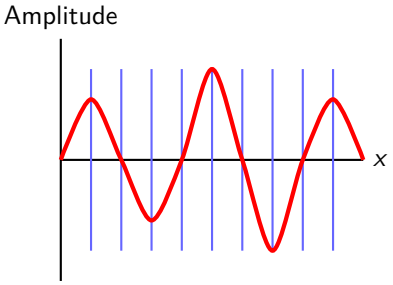
Dispelling the Parallel Search Myth

Common Misconception:

- "QC tries all keys at once"
- "Quantum = massive parallelism"
- "Like having 2^n classical computers"

Reality:

- Quantum interference patterns
- Amplitude cancellation/reinforcement
- Extracts global properties



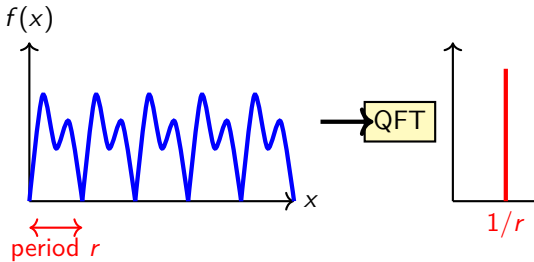
Interference reveals structure

Better Analogy:
Like a diffraction grating revealing hidden wavelengths

Shor's Algorithm - The Real Magic

Not brute force - finds hidden periods!

1. **Transform problem:**
   Factoring $\rightarrow$ Period finding
2. **Create superposition:**
   $\sum_x |x\rangle|f(x)\rangle$ over all values
3. **Extract period:**
   QFT reveals through interference



Physics Analogy:
Diffraction grating reveals light wavelength λ from interference pattern

Mathematical Connection

Goal: Factor $n = pq$ (RSA modulus)

1. Choose random a coprime to n
2. Define function: $f(x) = a^x \bmod n$
3. **Key insight:** This function is periodic!
   Find period r such that $a^r \equiv 1 \pmod{n}$
4. If r is even and $a^{r/2} \not\equiv -1 \pmod{n}$:

$$a^r - 1 \equiv 0 \pmod{n} \tag{1}$$

$$(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod{n} \tag{2}$$

5. Then $\gcd(a^{r/2} - 1, n)$ gives a factor!

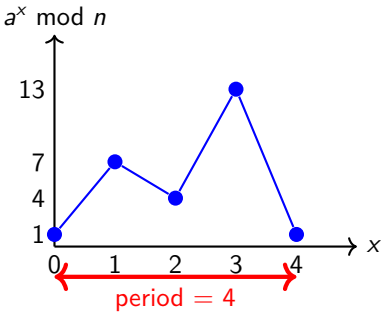
Success probability $\geq 1/2$ for random a

Concrete Example - Factoring 15

Let's factor $n = 15$ using $a = 7$:

$7^0 \bmod 15 = 1$
$7^1 \bmod 15 = 7$
$7^2 \bmod 15 = 49 \bmod 15 = 4$
$7^3 \bmod 15 = 28 \bmod 15 = 13$
$7^4 \bmod 15 = 91 \bmod 15 = \mathbf{1}$

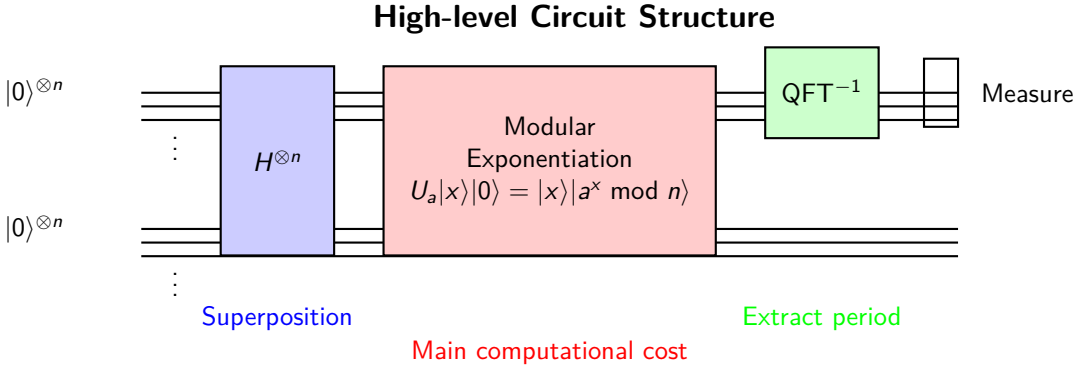
Period found: $r = 4$



Since $r = 4$ is even:
$a^{r/2} - 1 = 7^2 - 1 = 48$
$\gcd(48, 15) = \gcd(48, 15) = 3$

$15 = 3 \times 5$ ✓

Quantum Circuit for Shor



Resource Requirements:

- $\sim 3n$ logical qubits for n-bit RSA modulus
- $\sim n^3$ gates (mostly in modular exponentiation)

The Devastating Efficiency

Classical vs Quantum Time Complexity

RSA Size	Classical Time	Quantum Time	Speedup
RSA-1024	10^6 years	3 hours	$10^{10}\times$
RSA-2048	10^{20} years	8 hours	$10^{24}\times$
RSA-4096	10^{40} years	15 hours	$10^{44}\times$
RSA-8192	10^{80} years	30 hours	$10^{84}\times$

No escape through bigger keys!

- Classical: Exponential in key size
- Quantum: Polynomial in key size
- Doubling key size: Classical $\times 10^{20}$, Quantum $\times 2$

Time

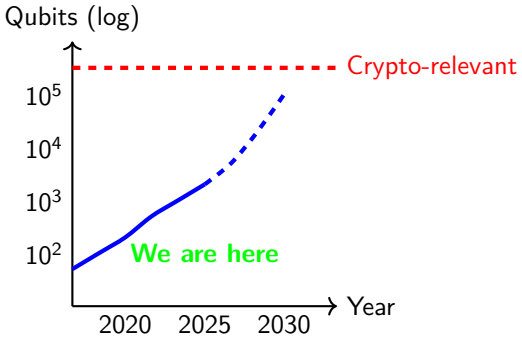
Current Quantum Hardware Reality Check

Requirements for Breaking RSA-2048:

- **Logical qubits:** $\sim$4,000
- **Error rate:** $< 10^{-10}$
- **Gate depth:** $\sim 10^9$

Physical Reality:

- Logical $\rightarrow$ Physical: $\sim$10,000:1
- **Total needed:** 20-40 million physical qubits
- **Current record:** $\sim$1,000 qubits
- **Gap:** 4-5 orders of magnitude



Timeline Estimate:
10-20 years for cryptographically relevant quantum computers

Breaking Signatures Too!

Same Quantum Algorithms Break All Classical Public Key Crypto

Signature Scheme	Underlying Problem	Quantum Algorithm
RSA Signatures	Integer Factorization	Shor's Algorithm
DSA	Discrete Logarithm	Modified Shor
ECDSA	Elliptic Curve DLog	Modified Shor
EdDSA	Elliptic Curve DLog	Modified Shor

Impact of Signature Breaking:

- Forge ANY digital signature
- Impersonate ANY website (fake certificates)
- Falsify software updates
- Compromise blockchain transactions
- Destroy digital trust infrastructure

Grover's Algorithm - The Symmetric Threat

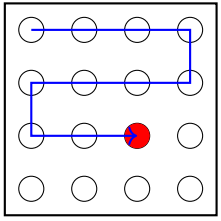
Different Quantum Approach:

- Amplitude amplification
- Search N items in $\sqrt{N}$ time
- Quadratic speedup (not exponential)

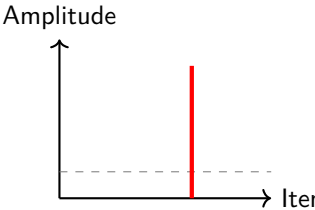
Impact on Symmetric Crypto:

Algorithm	Classical	Quantum
AES-128	128-bit	64-bit
AES-256	256-bit	128-bit
SHA-256	128-bit coll.	85-bit coll.
SHA-512	256-bit coll.	170-bit coll.

Solution: Double key sizes
AES-256 remains secure
SHA-384/512 remain secure



Classical: $O(N)$ steps



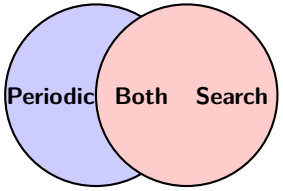
Quantum: $O(\sqrt{N})$ steps

Why These Quantum Advantages?

What Makes Some Problems Quantum-Easy?

Shor Exploits:

- Hidden periodic structure
- Fourier analysis natural for QC
- Interference extracts global property



Grover Exploits:

- Superposition over search space
- Amplitude amplification
- Quadratic speedup only

The Search for Quantum-Hard Problems

Requirements for Post-Quantum Security:

- × No hidden periodicity (Shor doesn't apply)
- × No efficient quantum algorithm known
- ✓ Still usable for cryptography
- ✓ Efficient classical algorithms for legitimate users
- ✓ Decades of cryptanalysis attempts

The Search for Quantum-Hard Problems

Requirements for Post-Quantum Security:

- × No hidden periodicity (Shor doesn't apply)
- × No efficient quantum algorithm known
- ✓ Still usable for cryptography
- ✓ Efficient classical algorithms for legitimate users
- ✓ Decades of cryptanalysis attempts

Enter: LATTICE PROBLEMS

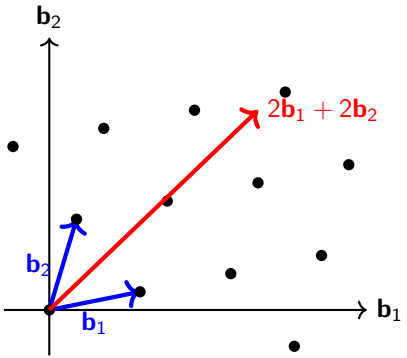
Lattices - Your Physics Background Helps!

Mathematical Definition:

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}_i : a_i \in \mathbb{Z} \right\} \quad (3)$$

You Already Know This From:

- Crystal structures (Bravais lattices)
- Reciprocal lattices
- Brillouin zones
- Miller indices
- X-ray diffraction



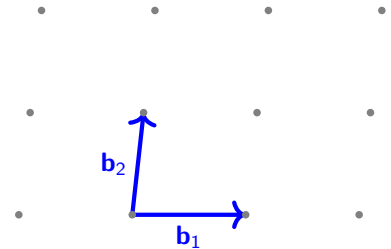
Crystal Structure Analogy:
FCC, BCC, hexagonal - all lattices!

Your Advantage:
Geometric intuition from condensed matter physics!

Good Basis vs Bad Basis

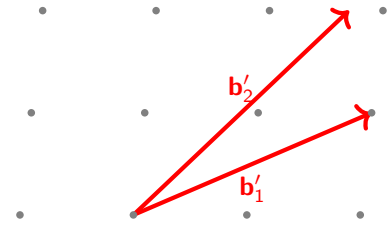
Same Lattice, Different Bases

Good Basis



Nearly orthogonal
Short vectors
Easy to work with

Bad Basis



Highly skewed
Long vectors
Hard to analyze

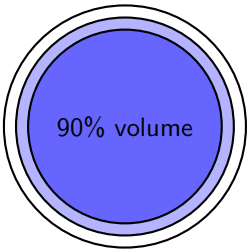
The Dimension Explosion

Your 3D Intuition:

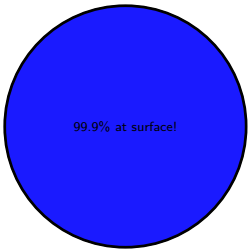
- Vectors have clear "direction"
- Can visualize shortest vector
- Volume is... volume
- Few "nearly shortest" vectors

In 500+ Dimensions:

- Volume concentrates at surface
- All vectors nearly orthogonal
- Exponentially many short vectors
- Geometry becomes counterintuitive



$n = 3$



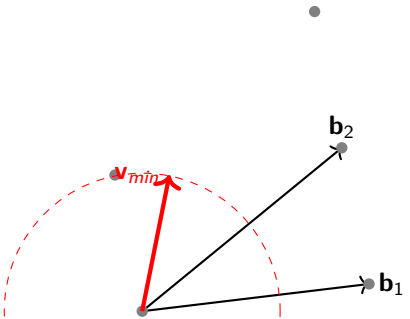
$n = 500$

Curse of Dimensionality:
Algorithms that work in 3D fail spectacularly in 500D

The Shortest Vector Problem (SVP)

Core Lattice Problem #1

Given: A basis $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ of lattice $\mathcal{L}$
Find: The shortest non-zero vector $\mathbf{v} \in \mathcal{L}$



- **Complexity:**
  - NP-hard to solve exactly
  - NP-hard to approximate within factor n^c
  - Best algorithms: $2^{O(n)}$ time

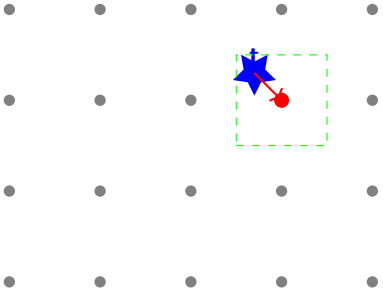
Why it's hard:

- Can't just try basis vectors
- Need integer combinations
- Exponentially many to check

The Closest Vector Problem (CVP)

Core Lattice Problem #2

Given: Lattice $\mathcal{L}$ and target point $\mathbf{t} \notin \mathcal{L}$
Find: Closest lattice point $\mathbf{v} \in \mathcal{L}$ to $\mathbf{t}$



Connection to Coding:

- Lattice = Error correcting code
- Target = Received noisy codeword
- CVP = Decode to nearest codeword

Also NP-hard!

- Even harder than SVP
- Used directly in crypto

The Algorithm Zoo

Algorithm	Year	Time	Quality
LLL	1982	Polynomial	Exponential approx
BKZ-k	1994	$2^{O(k)}$	$(2k)^{n/2k}$ approx
Sieving	2008	$2^{0.292n+o(n)}$	Exact SVP
Enumeration	2010	$2^{O(n \log n)}$	Exact SVP

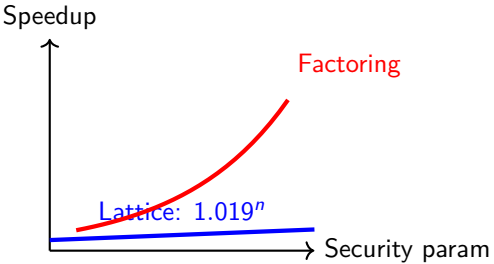
- **LLL:** Fast but poor quality (sufficient for some attacks)
- **BKZ:** Time-quality tradeoff via blocksize k
- **Sieving:** Best asymptotic complexity
- **Key point:** All exponential for good approximations!

The Quantum Speedup That Wasn't

Algorithm	Classical	Quantum	Speedup
Best SVP solver	$2^{0.292n}$	$2^{0.265n}$	$\approx 1.019^n$
Compare: Factoring	$2^{n^{1/3}}$	poly(n)	Exponential!

Tiny Quantum Advantage:

- Classical: $2^{0.292n}$
- Quantum: $2^{0.265n}$
- Speedup: $2^{0.027n} \approx 1.019^n$
- For $n = 500$: $1.019^{500} \approx 13,000\times$



Compare to Shor:

Why No Quantum Breakthrough?

What Makes Lattices Quantum-Resistant?

No Hidden Periodicity:

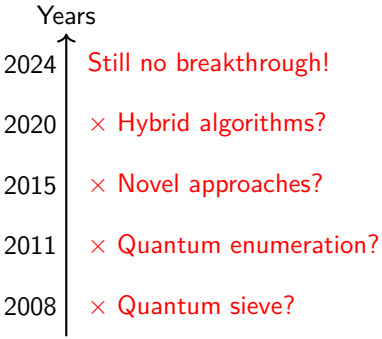
- Shor needs periodic structure
- Lattice problems are geometric
- No algebraic pattern to exploit

No Algebraic Structure:

- Can't use quantum Fourier transform
- No group structure to leverage
- Pure geometric optimization

High-Dimensional Geometry:

- Quantum walks don't help much
- No efficient amplitude amplification



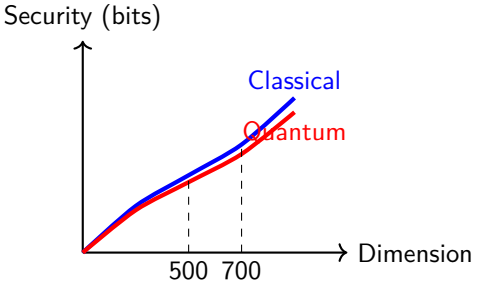
Concrete Security Numbers

Dimension vs Security Level

Dimension	Classical	Quantum	Equivalent
$n = 300$	2^{88}	2^{80}	-
$n = 500$	2^{146}	2^{133}	AES-128
$n = 700$	2^{204}	2^{186}	RSA-2048
$n = 1000$	2^{292}	2^{265}	AES-256

Key Takeaways:

- Still exponentially hard for QC
- Modest dimension increase compensates
- Well-understood security



The Lattice Advantage Summary

Why Lattices Win for Post-Quantum Crypto

Security Advantages:

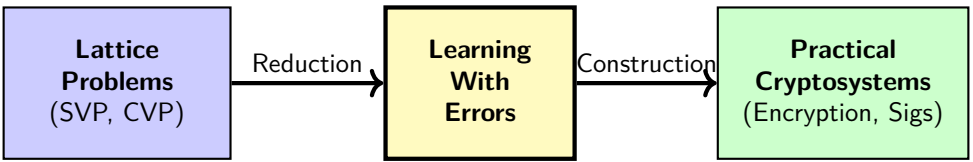
- ✓ No quantum algorithm breakthrough
- ✓ 15+ years of quantum cryptanalysis
- ✓ Worst-case hardness guarantees
- ✓ Well-understood security

Practical Advantages:

- ✓ Efficient implementations
- ✓ Multiple crypto primitives
- ✓ Parallelizable operations

Lattices: The Winner of Post-Quantum Cryptography

The Challenge: How to Use Geometric Hardness?



LWE: The crucial bridge between abstract lattice problems and usable cryptography

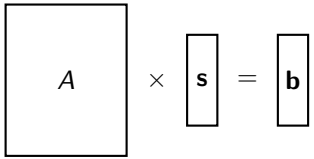
The LWE Insight - Linear Algebra Plus Noise

Without Noise:

$$\mathbf{b} = A\mathbf{s} \tag{4}$$

- Linear system of equations
- Solve by Gaussian elimination
- Polynomial time: $O(n^3)$
- Completely insecure!

Clean System

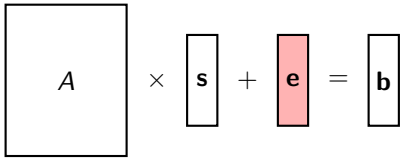


Noisy System

With Small Noise:

$$\mathbf{b} = A\mathbf{s} + \mathbf{e} \tag{5}$$

- Noisy linear system
- Gaussian elimination fails!
- Best algorithms: exponential
- Secure foundation



Physics Analogy:

LWE Formal Definition

Learning With Errors Problem

LWE Distribution:
Given secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution χ over $\mathbb{Z}_q$:

- Sample $\mathbf{a}_i \leftarrow \mathbb{Z}_q^n$ uniformly
- Sample $e_i \leftarrow \chi$ (small error)
- Output $(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q)$

The LWE Problem:
Given many samples $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ from LWE distribution:
$\rightarrow$ Recover the secret $\mathbf{s}$

Parameters:

Hardness:

Concrete LWE Example

Let's See LWE with Small Numbers

Parameters: $n = 4$, $q = 97$, secret $\mathbf{s} = (3, 7, 2, 5)$

$\mathbf{a}_i$	$\langle \mathbf{a}_i, \mathbf{s} \rangle$	e_i	b_i
$(14, 28, 91, 45)$	$42 + 196 + 182 + 225 = 66 \bmod 97$	$+1$	67
$(73, 12, 56, 89)$	$219 + 84 + 112 + 445 = 82 \bmod 97$	-1	81
$(31, 65, 22, 18)$	$93 + 455 + 44 + 90 = 4 \bmod 97$	0	4
$(88, 47, 39, 71)$	$264 + 329 + 78 + 355 = 54 \bmod 97$	$+2$	56

Try to solve this!

- Without errors: Easy linear algebra
- With errors: Which equation has $e = 0$? ± 1? ± 2?
- Can't use standard techniques!

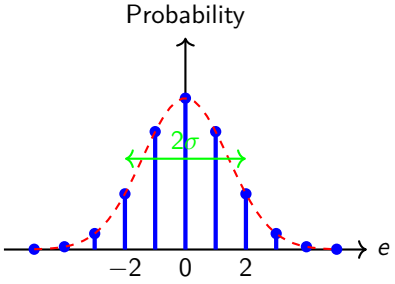
The Error Distribution

The Critical Component: Discrete Gaussian Noise

Discrete Gaussian Distribution:

$$\Pr[e = x] \propto \exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad (6)$$

for $x \in \mathbb{Z}$



Key Properties:

- Center: $\mu = 0$
- Width: σ (standard deviation)
- Support: Integers only
- Tails decay exponentially

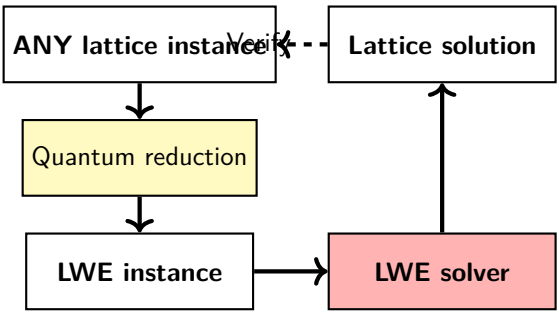
The Balancing Act:

- Too small σ: Insecure

Security comes from:
Many possible error patterns
$\rightarrow$ Many possible secrets

The Theoretical Foundation (2005)

Regev's Theorem:
If you can solve LWE on average, you can solve worst-case lattice problems



From Hard Problem to Encryption Scheme

Key Generation:

- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$
- Public key: $(A, \mathbf{b} = A\mathbf{s} + \mathbf{e})$

Encrypt bit $\mu \in \{0, 1\}$:

1. Sample random subset $S \subseteq \{1, \ldots, m\}$
2. Compute:

$$\mathbf{a}' = \sum_{i \in S} \mathbf{a}_i \tag{7}$$

$$b' = \sum_{i \in S} b_i + \mu \cdot \lfloor q/2 \rfloor \tag{8}$$

Why Decryption Works

Mathematical Analysis of Correctness

Let's trace through decryption:

$$v = b' - \langle \mathbf{a}', \mathbf{s} \rangle \tag{9}$$

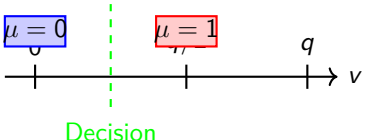
$$= \sum_{i \in S} b_i + \mu \cdot \lfloor q/2 \rfloor - \left\langle \sum_{i \in S} \mathbf{a}_i, \mathbf{s} \right\rangle \tag{10}$$

$$= \sum_{i \in S} (\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) + \mu \cdot \lfloor q/2 \rfloor - \sum_{i \in S} \langle \mathbf{a}_i, \mathbf{s} \rangle \tag{11}$$

$$= \sum_{i \in S} e_i + \mu \cdot \lfloor q/2 \rfloor \tag{12}$$

Key Insight:

- Error term: $\sum_{i \in S} e_i$ is small



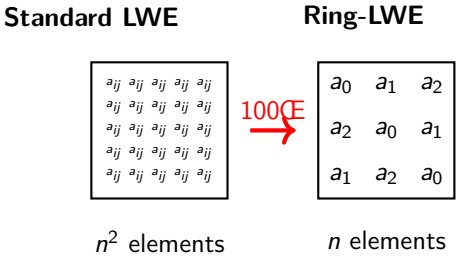
The Efficiency Revolution

Problem with Basic LWE:

- Public key: $n \times m$ matrix
- Size: $O(n^2)$ elements
- For $n = 1024$: Several MB!
- Too large for practice

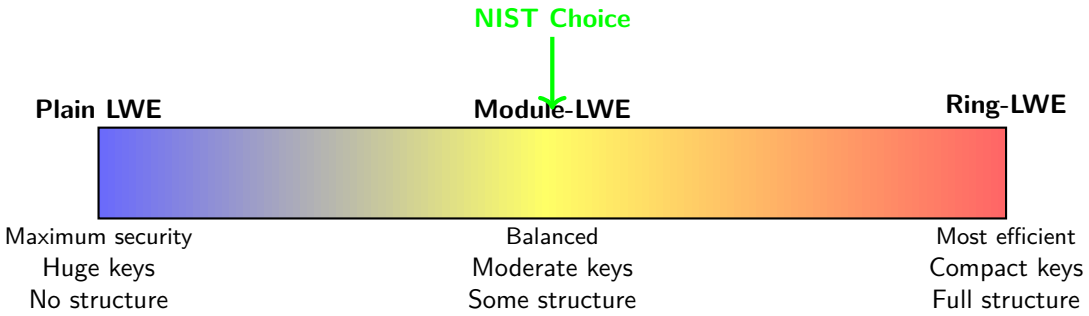
Solution: Add Structure

- Work in polynomial ring
- $R_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Structured matrices



Module-LWE - The Sweet Spot

The Spectrum of LWE Variants



Plain LWE:
Work with $\mathbb{Z}_q^n$
Full flexibility

Module-LWE:
Work with R_q^k
k small (2-4)

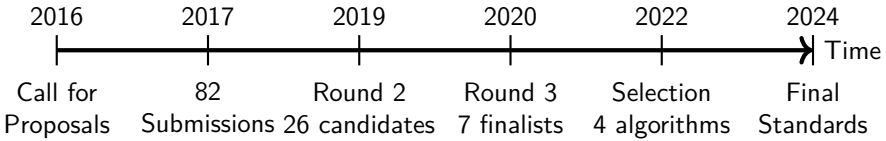
Ring-LWE:
Work with R_q
Maximum efficiency

The NIST Competition - Global Effort

The Largest Cryptographic Competition in History

82 submissions
25 countries
1000+ researchers

5 years
3 rounds
4 winners



Why This Matters:

- Global consensus on best algorithms
- Extensive cryptanalysis by top experts

The Winners - Lattice Domination

NIST Post-Quantum Cryptography Standards

Algorithm	Type	Based On	Purpose
green!30ML-KEM (Kyber)	KEM	Module-LWE	Encryption
green!30ML-DSA (Dilithium)	Signature	Module-LWE	Digital Signatures
green!30Falcon	Signature	NTRU Lattices	Signatures (alt.)
yellow!30SPHINCS+	Signature	Hash-based	Backup option

Key Takeaway:
3 of 4 winners are lattice-based!

Lattices dominate PQC

Why Lattices Won: Best balance of security, efficiency, and flexibility

ML-KEM (Kyber) Overview

The New Encryption Standard

Purpose: Replace ECDH/RSA for key exchange

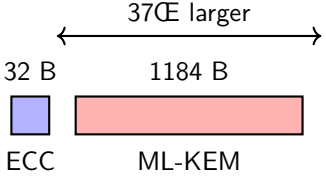
Key Features:

- Based on Module-LWE
- IND-CCA2 secure
- Three security levels
- Often faster than ECC!

Performance:

- KeyGen: 12-38 s
- Encaps: 16-48 s
- Decaps: 17-55 s

Level	PK	CT	Security
ML-KEM-512	800 B	768 B	128-bit
ML-KEM-768	1184 B	1088 B	192-bit
ML-KEM-1024	1568 B	1568 B	256-bit



Already deployed: Chrome, Cloudflare, AWS, Signal...

ML-DSA (Dilithium) Overview

The New Signature Standard

Purpose: Replace ECDSA/RSA signatures

Key Features:

- "Fiat-Shamir with Aborts"
- Rejection sampling for security
- Deterministic signatures
- Very fast verification

How it Works:

1. Create candidate signature
2. Check if it leaks secret
3. If yes: reject and retry
4. If no: output signature

Level	PK	Sig	Security
ML-DSA-44	1312 B	2420 B	128-bit
ML-DSA-65	1952 B	3309 B	192-bit
ML-DSA-87	2592 B	4627 B	256-bit

Performance:

- Sign: 100-300 s
- Verify: 50-150 s
- Faster than RSA verify!

Challenge:
Signatures 50-100Œ larger than ECDSA

Real-World Deployment - Cloudflare

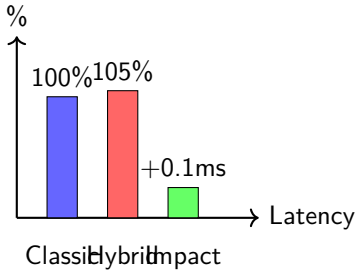
Case Study: 20+ Million Websites Protected

Cloudflare's PQC Timeline:

- 2019: Initial experiments
- 2022: Beta testing with Chrome
- 2023: Hybrid mode default
- 2024: Full production

Deployment Strategy:

- Hybrid: X25519 + Kyber768
- Gradual rollout
- Extensive monitoring
- Fallback mechanisms



Results:
<0.1ms latency increase
Zero user complaints
Quantum-safe today!

The Size Challenge

The Elephant in the Room: Key and Signature Sizes

Algorithm	Public Key	Signature/CT	vs Classic
ECDSA P-256	64 B	64 B	1Œ
RSA-2048	256 B	256 B	4Œ
ML-KEM-768	1,184 B	1,088 B	18Œ
ML-DSA-65	1,952 B	3,309 B	52Œ
Falcon-512	897 B	666 B	10Œ

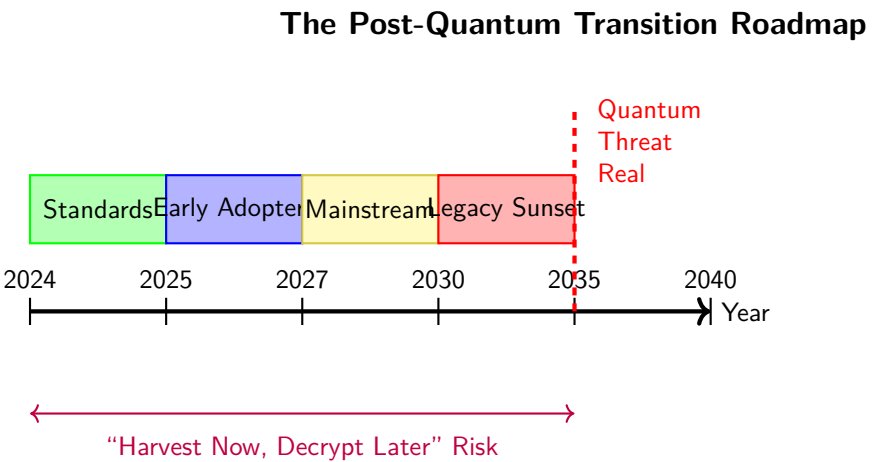
Impact Areas:

- TLS handshakes
- Certificate chains
- IoT devices
- Blockchain systems
- Storage requirements

Mitigation Strategies:

- Caching public keys
- Compression techniques
- Protocol optimizations
- Hybrid modes (transitional)
- Hardware acceleration

Migration Timeline



NOW (2024-2025):

URGENT:

Recommended during transition: Best of both worlds - no security regression

Safe Transition Strategy: Use Both!

Your Unique Advantages as Physicists

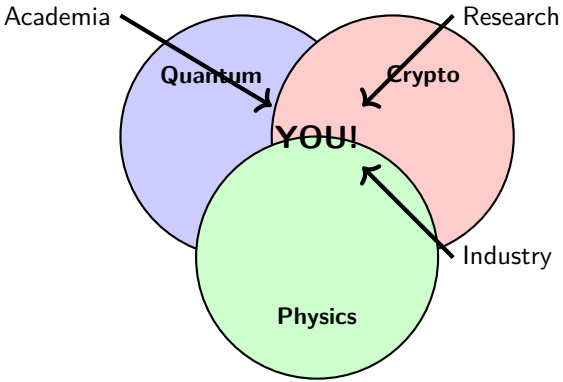
Why YOU Are Perfectly Positioned

Technical Advantages:

- ✓ Real quantum intuition
- ✓ Lattice experience (crystals)
- ✓ Noise/error understanding
- ✓ Hardware knowledge
- ✓ Mathematical maturity

Perspective Advantages:

- ✓ Bridge theory & experiment
- ✓ Systems thinking



Open Problems Where Physics Meets PQC

Theoretical Directions:

- Quantum algorithms for lattices
- New hardness assumptions
- Connections to condensed matter
- Quantum reduction improvements
- Novel cryptographic constructions

Physical Perspectives:

- Thermodynamic crypto bounds
- Photonic implementations
- Quantum error analogies
- Phase transition insights

Practical Challenges:

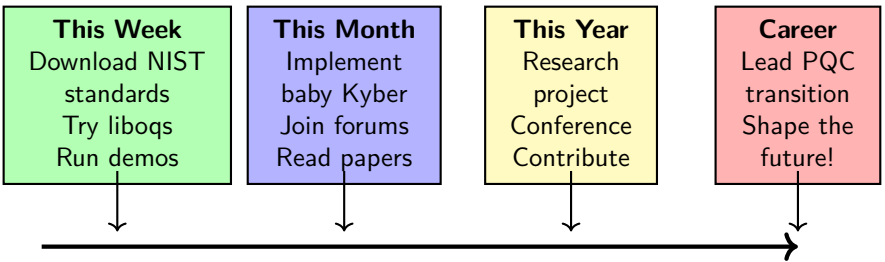
- Side-channel resistance
- Hardware acceleration
- Quantum RNG design
- Hybrid protocol optimization
- IoT implementations

Interdisciplinary:

- Quantum-classical interfaces
- Error correction crossover
- Physical unclonable functions
- Quantum key distribution + PQC

Practical Next Steps

Your Action Plan



Start Today: `https://github.com/open-quantum-safe/liboqs`

Resources for Deep Dive

Your Learning Toolkit

Essential Papers:

- Regev (2005) - LWE introduction
- Peikert (2016) - "A Decade of Lattice"
- NIST standards (2024)
- Micciancio-Regev survey

Code Libraries:

- `liboqs` - All NIST algorithms
- `PQClean` - Reference implementations
- `CRYSTALS` - Kyber/Dilithium
- `FrodoKEM` - Learning project

Online Courses:

- Coursera - Post-Quantum Crypto
- MIT OCW - Lattice Cryptography
- YouTube - Simons Institute

Communities:

- NIST PQC Forum
- IACR ePrint Archive
- PQC Standardization list
- Stack Exchange - Crypto

Remember These Four Things

1. **Current crypto WILL fail against quantum computers**
   RSA, ECDSA, DH - all broken by Shor's algorithm

2. **Lattice-based solutions are ready NOW**
   NIST standards published, real-world deployments working

3. **Migration must start IMMEDIATELY**
   "Harvest now, decrypt later" means secrets already at risk

Your Mission

"In 2074, historians will write about the post-quantum transition.

Will they write:

'Scientists acted in time'
or
'If only they had started sooner'?"

That story is being written by YOUR actions.

The future needs you. Act now.