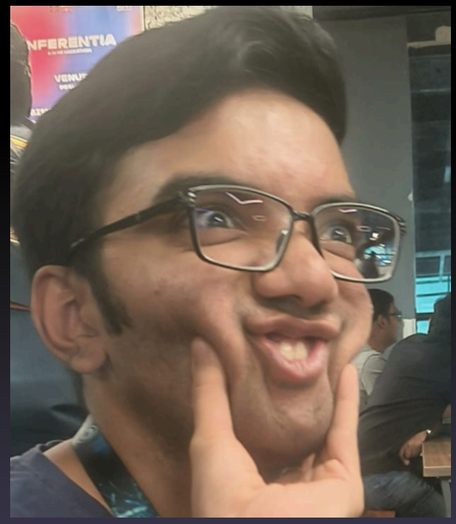
Introduction to

Quantum Post Quantum

Intro to Quantum Computing

Machiraju Karthikeya
Quantum computing student
researcher (Physics modeling)
Teaching assistant
6-foot feminist
Also IBM QISKIT advocate and
QForest club head



Krishna Sujith
Quantum computing student
researcher (hardware)
Local Mallu
Echo chamber keys
professional notice period
server

Intro to Quantum Crypto

Pranav Hemanth

Cybersecurity & AI student researcher

Teaching assistant

Professional tryhard :)

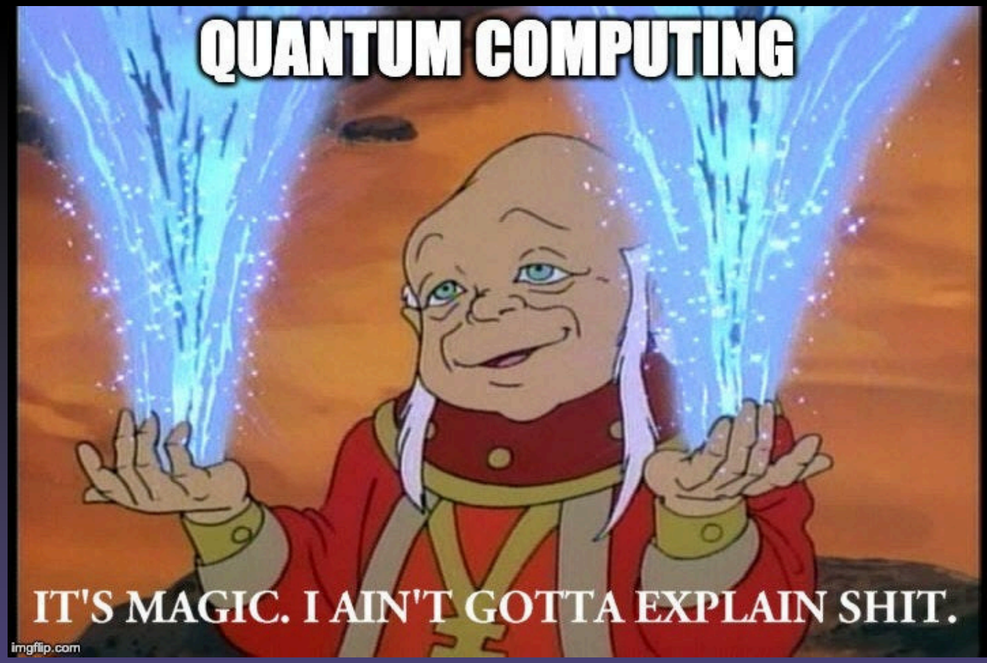
I chain some blocks sometimes

Pranavjeet Naidu

Student (maybe)

quantum shiiii

Classical Computer: “I’ll factor RSA-2048 in 300 trillion years!”
Quantum Computer: “Done. What’s next?”
Classical Computer: “...But you didn’t even show your work!”
Quantum Computer: “I showed ALL my work. Simultaneously

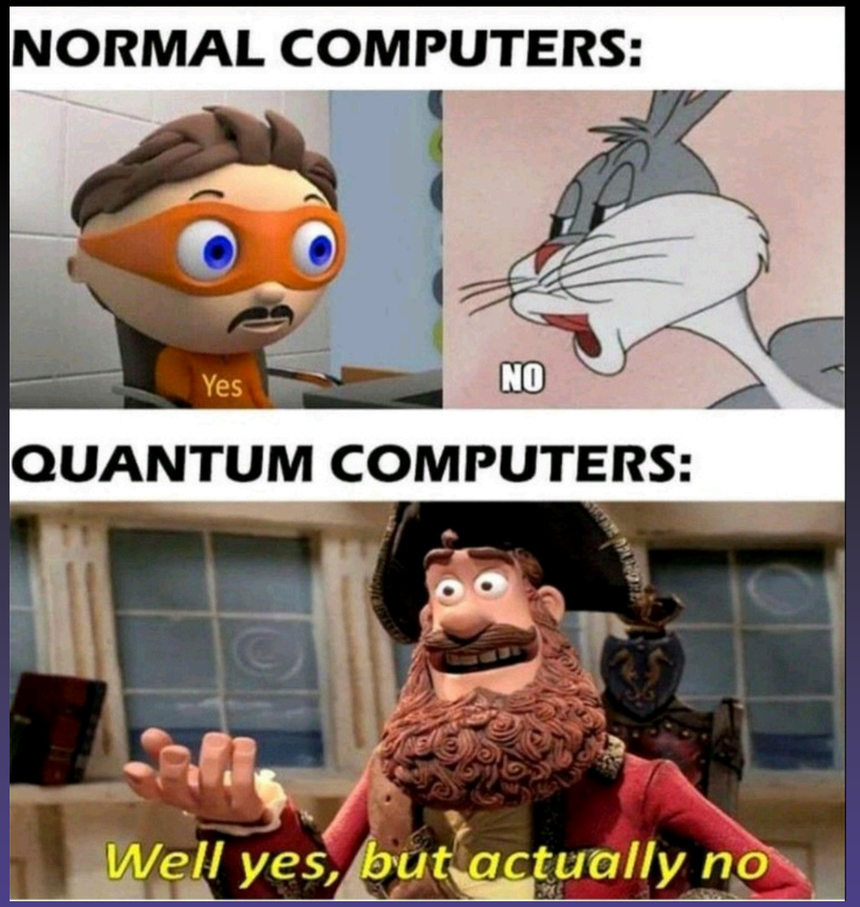


Classical: $O(2^{(\log N)^{1/3}})$ time — basically forever
Quantum: $O((\log N)^3)$ time — coffee break

Quantum spins, the cosmos quakes.

What is Prime Factorization?

- The Problem
- Given: A composite number N = p × q (where p, q are unknown primes)
- Find: The prime factors p and q
- Easy to verify: 15 = 3 × 5
- Hard to compute: Given 15, find 3 and 5 (trivial). Given RSA-2048, find factors
- (impossible classically).



Quantum computing = math in disguise

Classical Algorithm 1: Trial Division

- The Naive Approach
- Try dividing N by every number starting from 2:
- Check if N mod d = 0 for d = 2, 3, 4, 5, . . . , $\sqrt{N}$
- Example: Factoring 143
- 1. Is 143 mod 2 = 0? No (143 is odd)
- 2. Is 143 mod 3 = 0? No (1 + 4 + 3 = 8, not divisible by 3)
- 3. Is 143 mod 5 = 0? No (doesn’t end in 0 or 5)
- 4. Is 143 mod 7 = 0? No (143 = 7 × 20 + 3)
- 5. Is 143 mod 11 = 0? Yes! (143 = 11 × 13)
- Found factors: 11 and 13t

Pollard's Rho

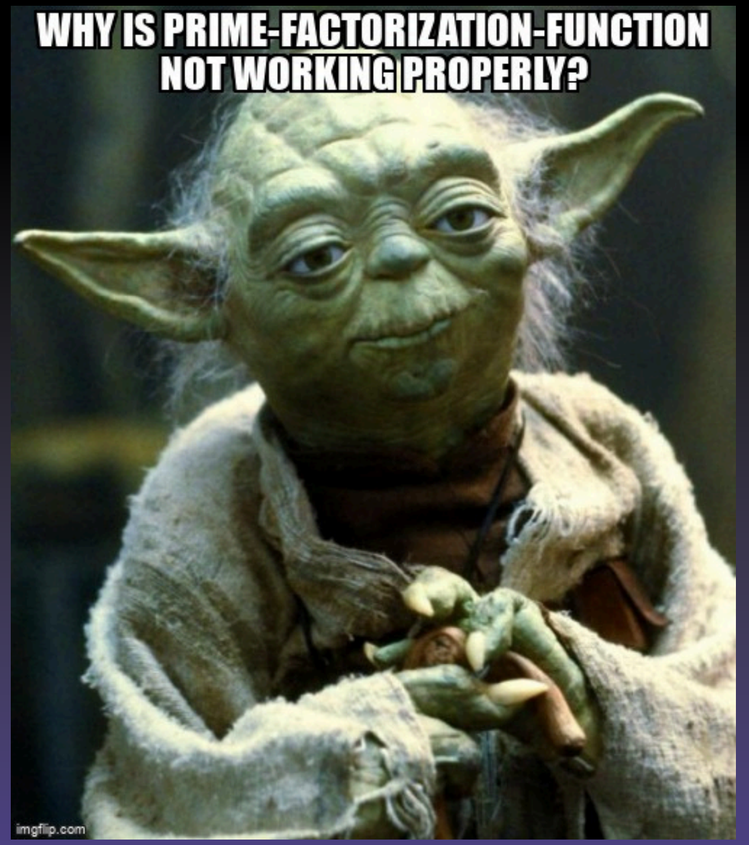
- A Better Classical Approach
- Don't check every divisor. Use randomization and cycle-finding:

1. Start with $x_0 = 2$ and iterate: $x_i = (x_{i-1}^2 + 1) \bmod N$
2. Compute $\gcd(|x_i - x_j|, N)$ for various i, j
3. If $\gcd > 1$ and $\gcd < N$: Found a non-trivial factor!
4. If stuck: restart with different random seed

- Why It Works (Intuition)
  - By the pigeonhole principle, xi mod p will repeat before xi mod N repeats
  - Where p is a factor of N
  - When the sequences diverge, gcd(xi − xj , N) catches the factor
  - Much faster than trial division!her classical loser.

Setting Up Your Quantum Workspace

Algorithm	Time Complexity	Practical for RSA-2048?
Trial Division	$O(\sqrt{N})$	No (impossible)
Pollard's Rho	$O(N^{1/4})$	No (impossible)
Quadratic Sieve	$O(e^{\sqrt{\log N \log\log N}})$	Maybe (2 weeks)
Number Field Sieve	$O(e^{(\log N)^{1/3}})$	Hardest known (16 years)



Quantum spins, the cosmos quakes.

Shor’s Insight

- Don’t factor directly. Instead:

1. Reduce factorization to the period-finding problem

- Given N, pick random $1 < a < N$, find the smallest r where:

$$a^r \equiv 1 \pmod{N}$$

This r is called the order or period.

2. Extract factors from the period

- Once you have r, use algebra to compute the factors.

Why This Helps

- Classical: Finding period needs O(√r) time, where r can be O(N) →
- exponential
- Quantum: Finding period needs O((log N)3) time → polynomial!
- This is the exponential speedup
- By shifting the problem, quantum gets its advantage

Quantum spins, the cosmos quakes.

Why This Helps

Why Period $\Rightarrow$ Factors? (The Magic Trick)

Once you have period r, use this algebra:

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$$

Since $a^r \equiv 1 \pmod{N}$, we know N divides $a^r - 1$.

Therefore:

$$\gcd(a^{r/2} \pm 1, N) = \text{factors of } N$$

Plot twist: This math works classically too! Finding r is the bottleneck.

Quantum spins, the cosmos quakes.

Why This Helps

Step 1: Quantum Superposition — The Grand Entrance

$$|\psi_0\rangle = |0\rangle \xrightarrow{\text{Hadamard}^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

Apply Hadamard gates to create superposition of ALL possible inputs at once.
It's like opening every door simultaneously instead of trying each key one by one.

Classical: "Let me try $x = 0$, then $x = 1$, then $x = 2$..."
Quantum: "I'm already in all states. Catch up."

Step 2: Modular Exponentiation — The Quantum Flex

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|a^x \bmod N\rangle$$

Compute $f(x) = a^x \bmod N$ for ALL values of x simultaneously using quantum gates.
No loops. No waiting. No sequential computation. Just pure quantum showing off.

This is called **quantum parallelism** and it's why quantum computers are unfairly good at this.

Why This Helps

Step 3: Measure Ancilla Register — The Collapse

Measure the second register (the $|a^x \bmod N\rangle$ part).

Let's say we measure some value $y_0 = a^{x_0} \bmod N$.

The first register collapses to periodic superposition:

$$|\psi_2\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{\lfloor n/r \rfloor} |x_0 + jr\rangle$$

Now the state *encodes* the period r in the spacing between terms!

But we can't just read r directly (quantum mechanics is passive-aggressive like that).

Step 4: Quantum Fourier Transform — The Secret Weapon

Apply QFT to the periodic state:

$$\text{QFT}\,|\psi_2\rangle \approx \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |k \cdot 2^n / r\rangle$$

Creates sharp probability peaks at multiples of $2^n/r$.

It's like the quantum computer is playing "Hot and Cold" and suddenly shouts **"BURNING HOT!"**

The QFT transforms *periodicity in position* into *peaks in frequency*. Pure magic (actually Fourier analysis).

Why This Helps

Step 3: Measure Ancilla Register — The Collapse

Measure the second register (the $|a^x \bmod N\rangle$ part).

Let's say we measure some value $y_0 = a^{x_0} \bmod N$.

The first register collapses to periodic superposition:

$$|\psi_2\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{\lfloor n/r \rfloor} |x_0 + jr\rangle$$

Now the state *encodes* the period r in the spacing between terms!

But we can't just read r directly (quantum mechanics is passive-aggressive like that).

Step 4: Quantum Fourier Transform — The Secret Weapon

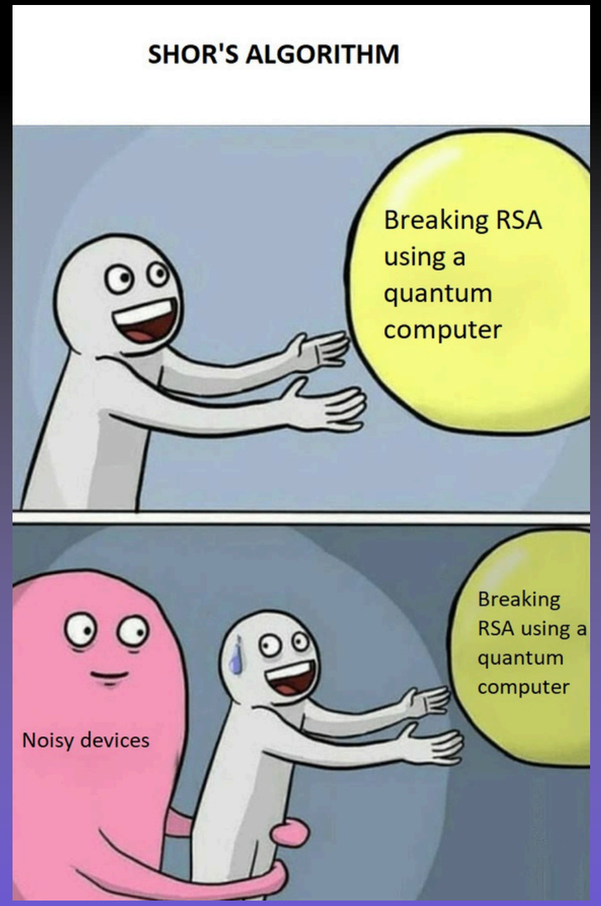
Apply QFT to the periodic state:

$$\text{QFT}\,|\psi_2\rangle \approx \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |k \cdot 2^n / r\rangle$$

Creates sharp probability peaks at multiples of $2^n/r$.

It's like the quantum computer is playing "Hot and Cold" and suddenly shouts **"BURNING HOT!"**

The QFT transforms *periodicity in position* into *peaks in frequency*. Pure magic (actually Fourier analysis).



Why This Helps

Step 5: Measurement & Classical Post-Processing

Measure the first register after QFT.

You'll get some value $\approx k \cdot 2^n/r$ for some integer k.

Classical computation:

1. Use continued fractions to extract r from measured value
2. Compute $\gcd(a^{r/2} \pm 1, N)$ to get factors
3. If you get 1 or N, try again with different a (rare failure case)
4. Otherwise: Congrats! You factored N. RSA is dead.

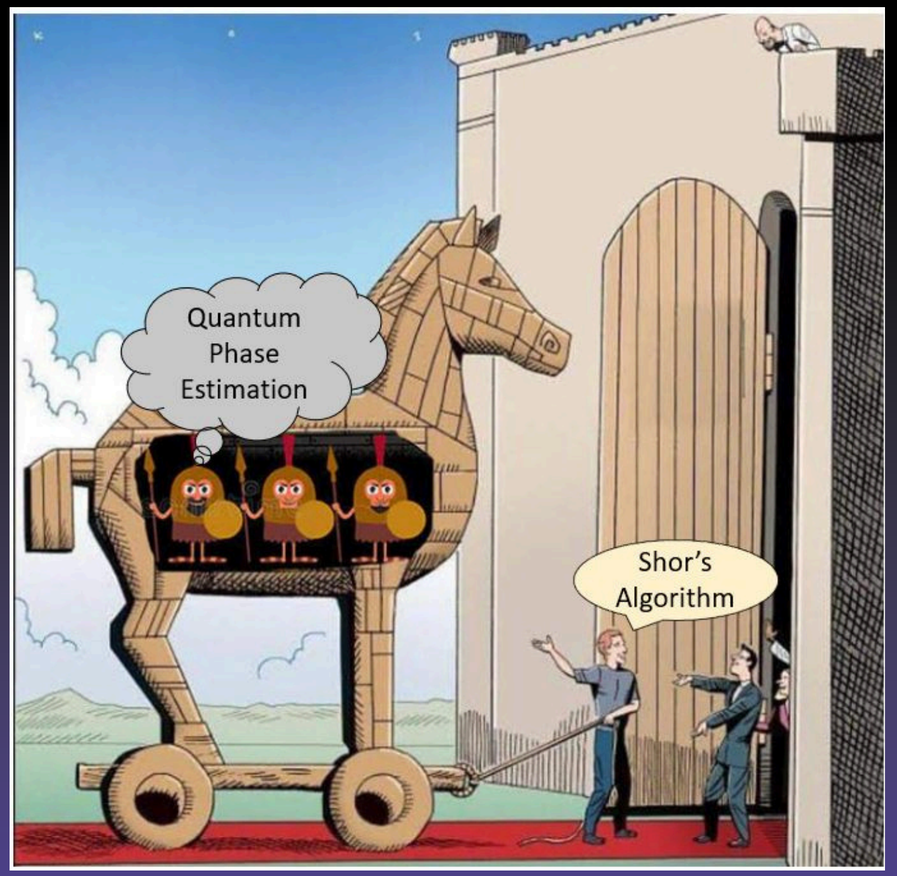
Success Probability

Each run has good probability of success ($> 50\%$).

If it fails, just run again with different random a.

Expected number of runs: $O(1)$ (constant!)

Bottom line: Polynomial time with high probability. Game over for RSA.



Quantum spins, the cosmos quakes.

Why This Helps

What QFT Does Mathematically

The Quantum Fourier Transform on n qubits transforms:

$$|x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i x y / 2^n} |y\rangle$$

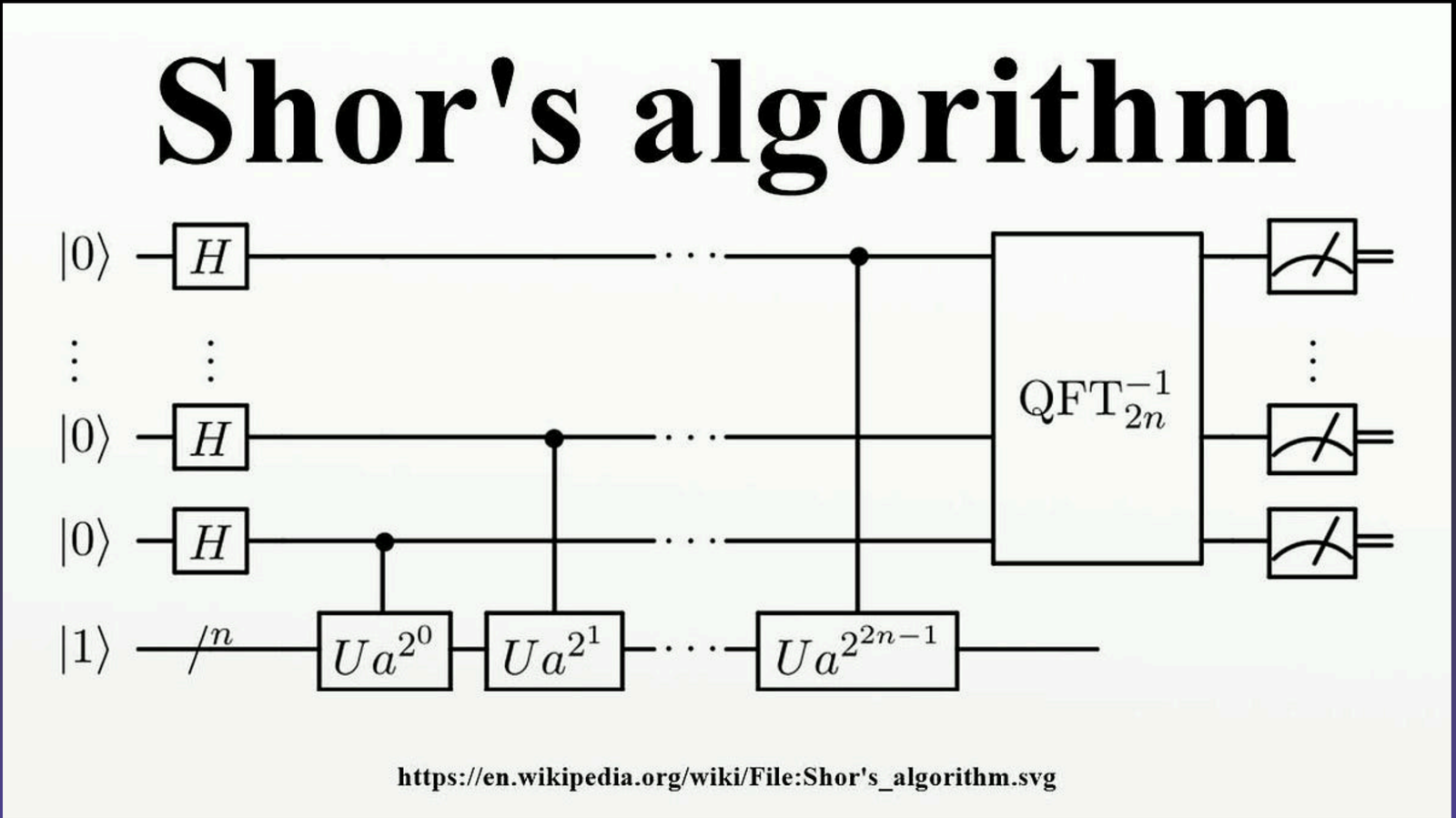
For a periodic state with period r:

$$\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |x_0 + jr\rangle \xrightarrow{\text{QFT}} \text{peaks at } |k \cdot 2^n / r\rangle$$

The periodicity in the input becomes peaks in the output!

Quantum spins, the cosmos quakes.

Why This Helps

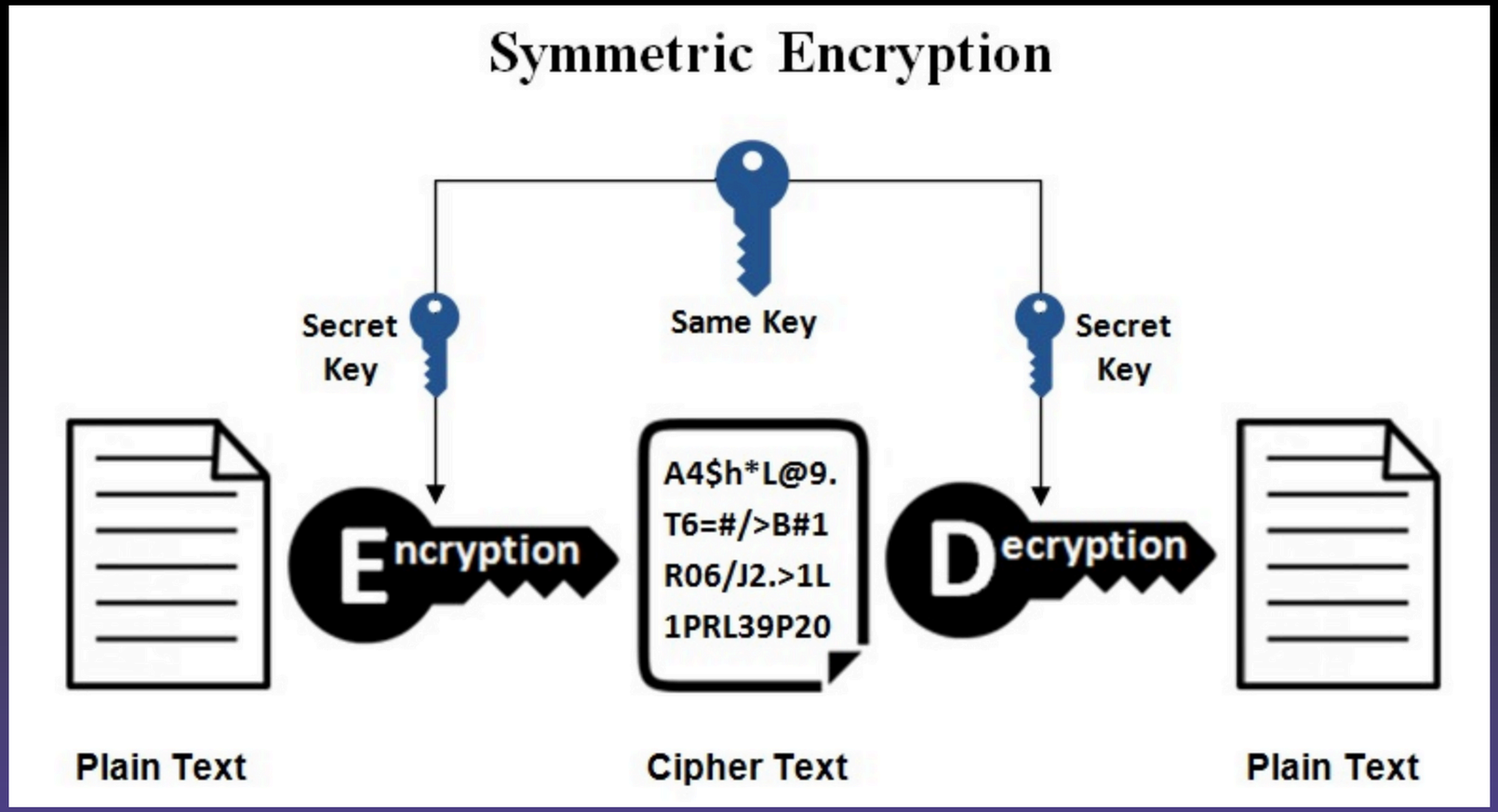


Quantum spins, the cosmos quakes.

Cryptography time

Quantum spins, the cosmos quakes.

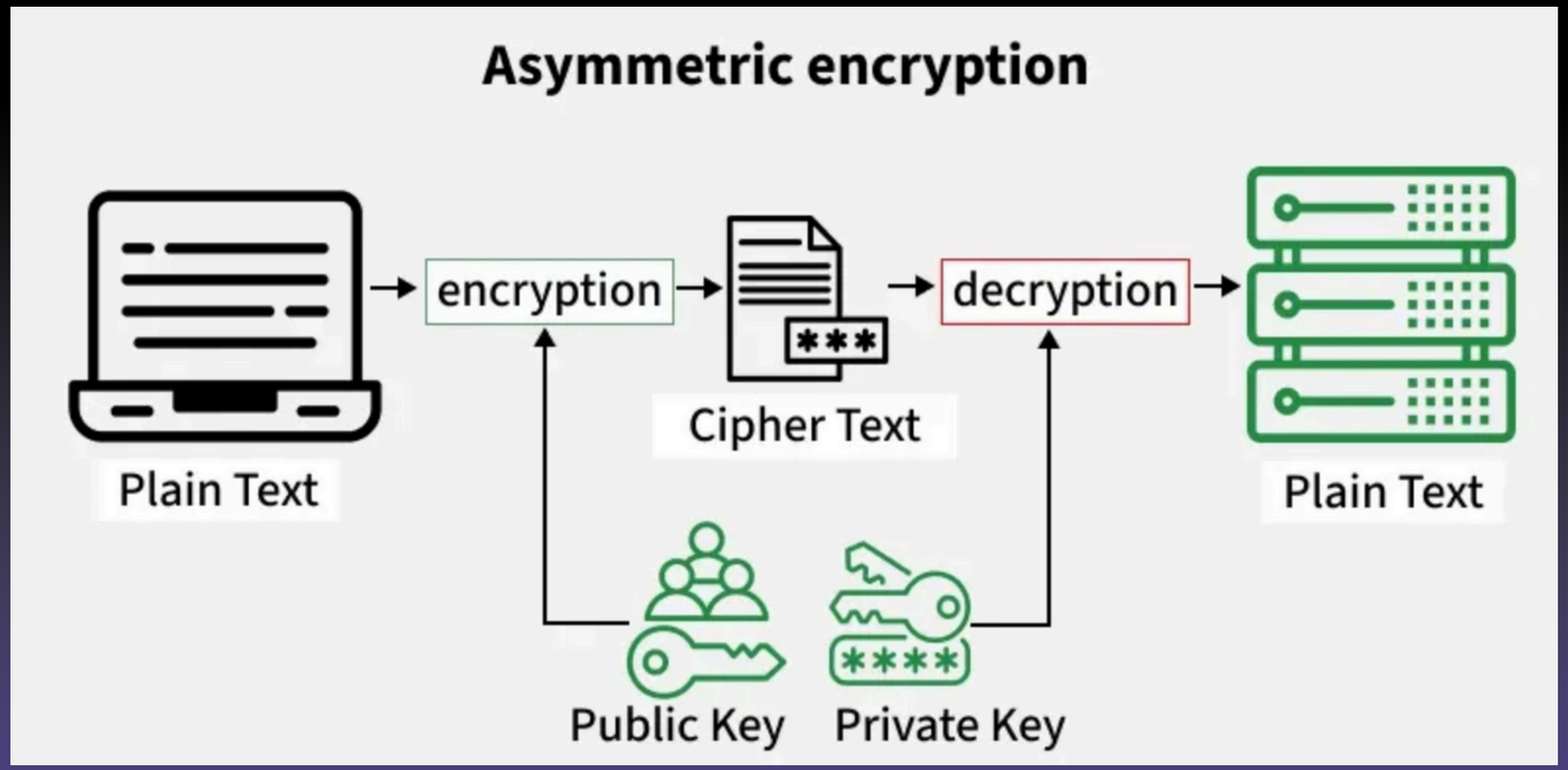
Symmetric Encryption



Grovers vs Symmetric

Grover’s algo lets a quantum computer brute-force symmetric keys in roughly $O(\sqrt{n})$ instead of $O(n)$ time, giving a quadratic speedup.

Asymmetric Encryption



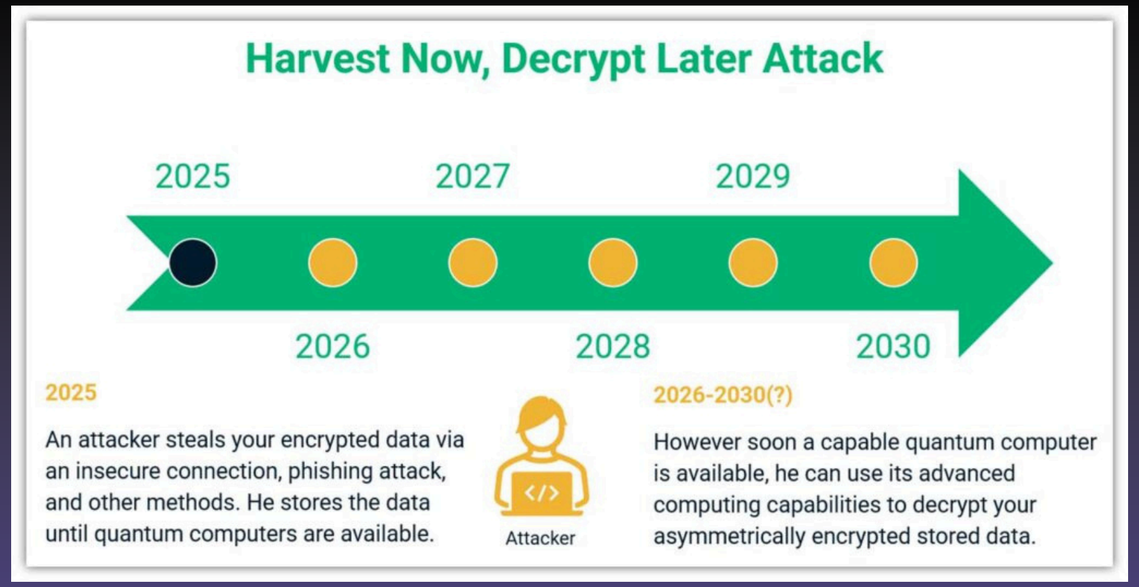
how do they break

Asymmetric - Shor’s

Shor’s algorithm efficiently factors large integers and solves discrete logarithms using quantum period finding, breaking the mathematical foundations of RSA, Diffie–Hellman, and ECC.

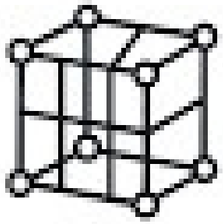
harvest now decrypt later

1. Stages of the attack
2. Why is it a serious threat
3. Who is at risk
4. How do we defend

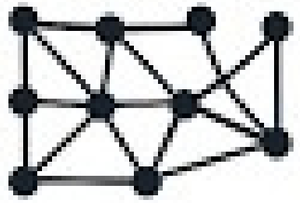


Quantum Proof Cryptography

Families of Post-Quantum Cryptography Algorithms



Lattice-Based



Code-Based

$x_1 - y_2$

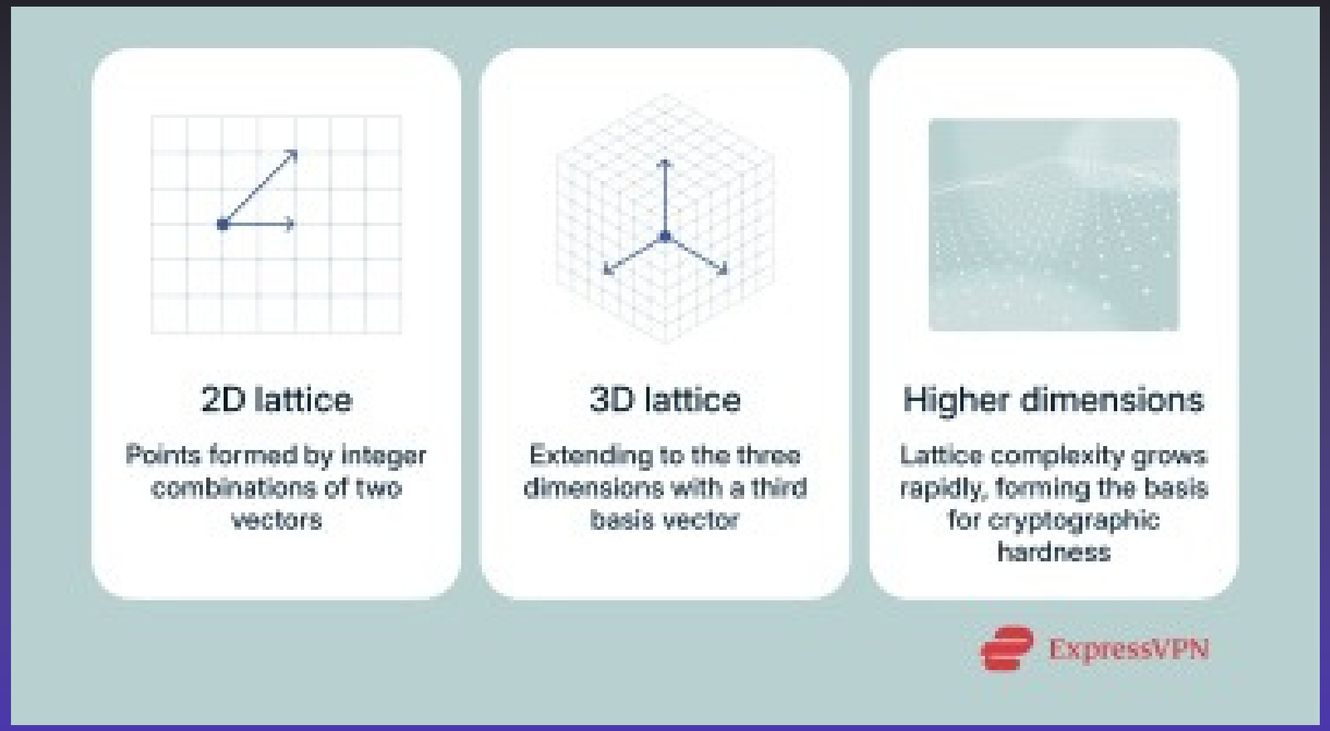
Multivariate Polynomial

Hash-Based

Quantum Proof Cryptography

Lattice-based (Kyber, Dilithium)

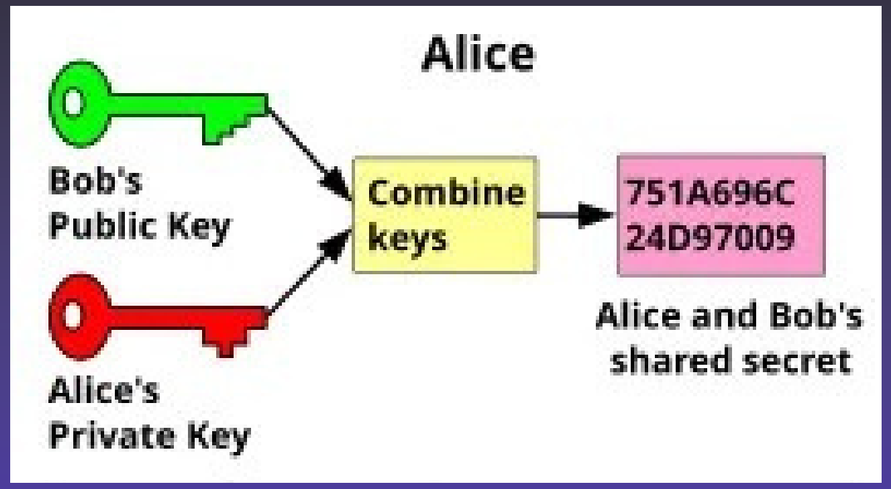
Security from the hardness of solving noisy high dimensional linear equations.

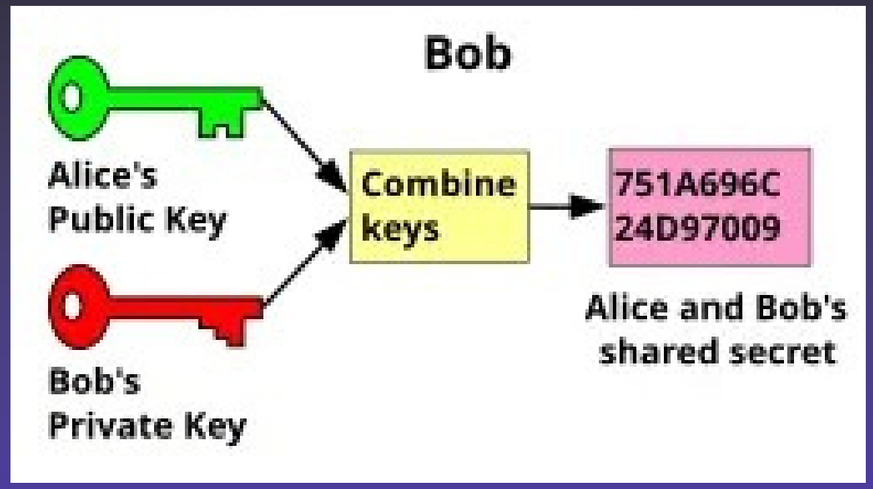


Quantum Proof Cryptography

Code-based (Classic McEliece)

Use error-correcting codes, but hide their structure.

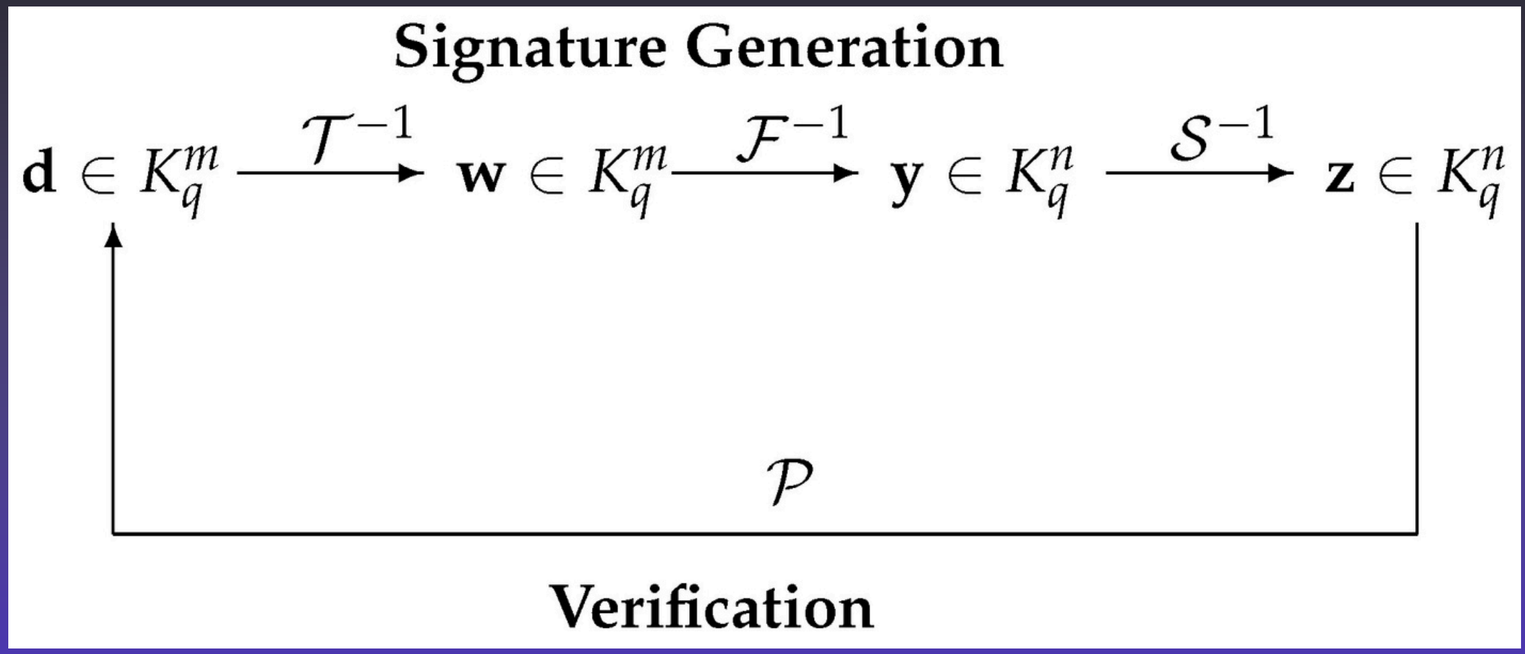




Quantum Proof Cryptography

Multivariate polynomial-based (Rainbow, Gemss)

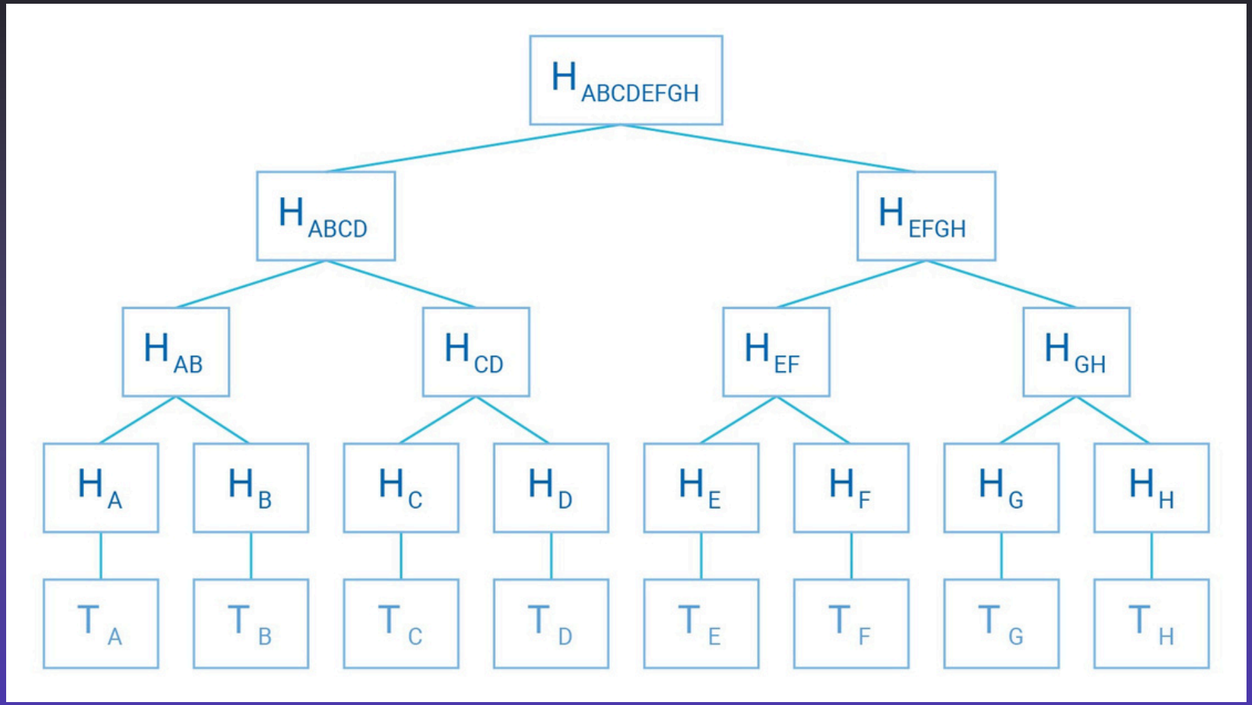
Security from solving systems of multivariate quadratic equations (MQ problem).



Quantum Proof Cryptography

Hash-based (SPHINCS+)

Security entirely built on the one-way property of cryptographic hash functions.



Thank You

-