

## **An open-source Post-Quantum blockchain layer 1 project**

Open to anyone who want to join and contribute based on his/ her passion in quantum-resistant blockchain ecosystem.

## **Vision**

Unlocking the Power of Post-Quantum Technology;

Immerse yourself in a revolutionary project that aims to redefine the blockchain landscape. Our mission is to create a decentralized, secure, and transparent network that remains impervious to both classical and quantum computing attacks.

## **Key Objective**

Harness the power of blockchain to create immutable records, fortifying defenses against malicious attacks aiming to tamper with or delete crucial data. By accomplishing these pivotal objectives, our open-source post-quantum blockchain layer 1 project will contribute to the establishment of a more secure and resilient digital world.

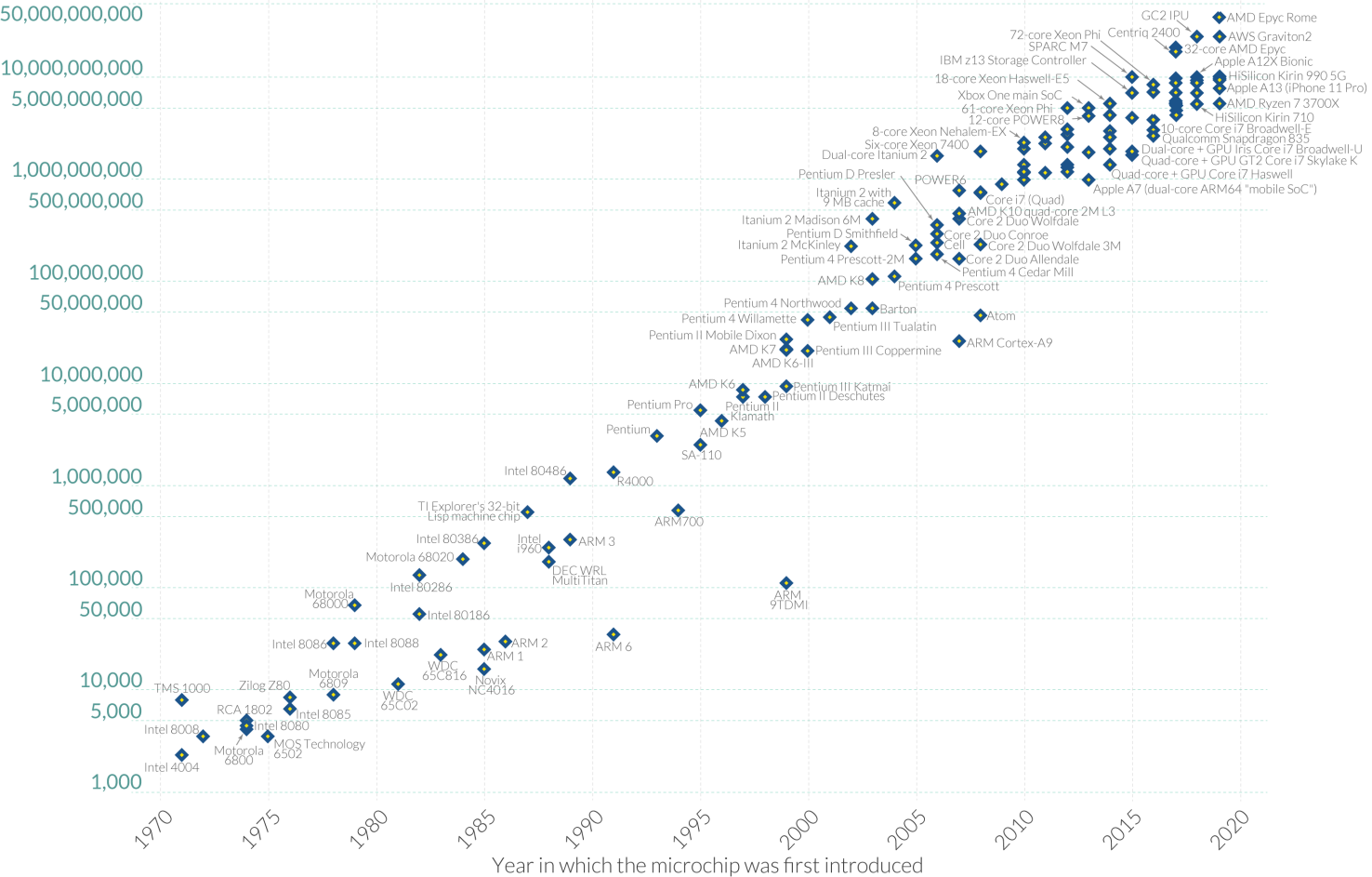
# THE PROBLEM

## Moore's Law: The number of transistors on microchips has doubled every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



### Transistor count



## Moore's Law

## IBM Roadmap to Scaling Quantum Technology

1. In 2023 IBM quantum scientists are building a quantum computer with a 1,121-qubit processor, called Condor.
2. Condor lays the groundwork for scaling to fully error-corrected, interconnected, 1-million-plus-qubit quantum computers.
3. In 2021, IBM will debut the 127-qubit "Eagle" chip
4. Eagle will be followed by the 433-qubit "Osprey" processor in 2022.



**Figure 1**

Quantum computing's potential for significant speedup over classical computers, according to IBM internal analysis

Type of scaling	Time to solve problem				
Classical algorithm with exponential runtime	10 secs	2 mins	330 years	3300 years	Age of the universe
Quantum algorithm with polynomial runtime	1 min	2 mins	10 mins	11 mins	~24 mins

# Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer\*

Peter W. Shor<sup>†</sup>

## Abstract

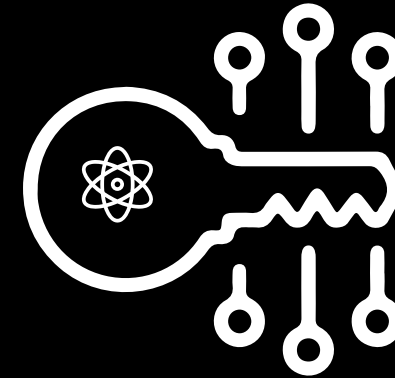
A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

**Keywords:** algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

**AMS subject classifications:** 81P10, 11Y05, 68Q10, 03D10

## How fast can a quantum computer crack encryption?

Researchers at the University of Sussex estimated in February 2022, that a quantum computer with 1.9 billion qubits could essentially crack the encryption safeguarding Bitcoin within a mere **10 minutes**. Just 13 million qubits could do the job in about a day.



## Encryption & KEM's

Well, Shor's algorithm really addresses this problem:

Given a function  $F$  where the identity  $F(a+x) = F(a)$  holds for all  $a$ , find  $x$ .

It turns out that, by selecting  $F$  properly, we can use it either to factor or to compute discrete logarithms. However, we select different  $F$  functions for solving those two problems.

Now, in terms of elliptic curves, the discrete logarithm problem is "given two points  $P, Q = xP$ , find  $x$ ".

So, to apply Shor's to this problem, we define  $F(u, v) = uP + vQ$  (note that this  $F$  has two inputs; actually, Shor's doesn't really care about that); by defining the  $+$  operation properly, we can get Shor's to give us  $x, y$  values [1] such that  $F(a, b) = F(a + x, b + y)$ ; that is  $aP + bQ = (a + x)P + (b + y)Q$  or  $xP + yQ = 0$ . Assuming  $y$  doesn't happen to be 0, we get  $Q = -xy^{-1}P$  [2], which is our solution.

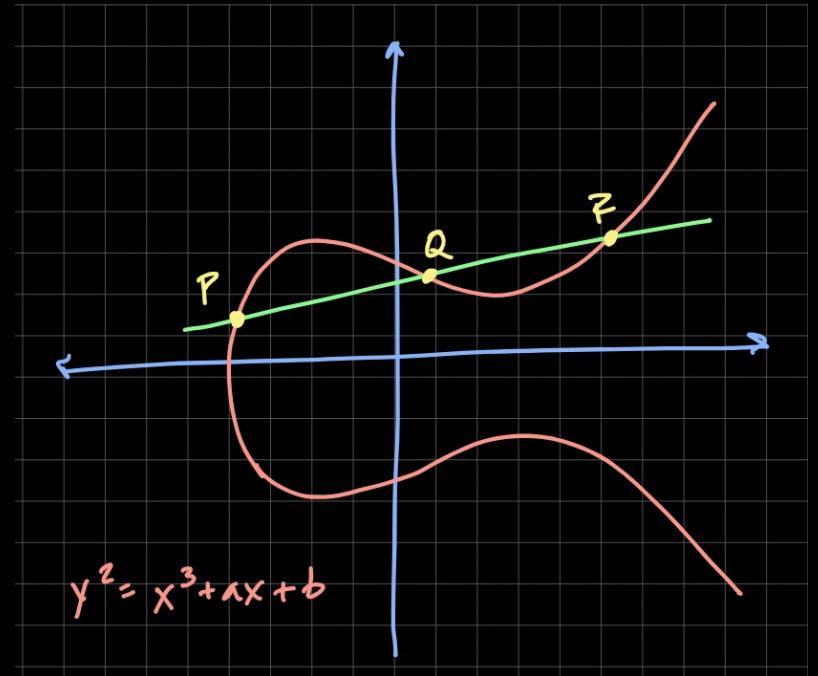
And, tying this back to ECDSA, once we can compute the discrete log of the public key, that's the private key, and that allows the attacker to sign any message he wants.

[1]: Actually, there are lots of such  $(x, y)$  pairs - we don't care which one Shor's gives us as long as  $y \neq 0$

[2]:  $-xy^{-1}$  is mathematical notation;  $y^{-1}$  is that value  $z$  such that  $y \times z \equiv 1 \pmod{n}$ , where  $n$  is the number of points on the curve, and  $-xy^{-1}$  is the value  $u$  such that  $u + xy^{-1} \equiv 0 \pmod{n}$ . This is easily computed if we know  $x, y$  and  $n$  (which we do)

[source](#)

## EC(DSA) VS Shor's Algorithm





Grover's Algorithm can reverse a black-box function implemented as a quantum oracle in  $O(\sqrt{N})$  iterations with  $O(\log_2 N)$  qubits, with  $N$  being the number of possible input combinations to the function. In this context, the quantum oracle phase-flips a target qubit when the desired output is produced (e.g., when the password is correct). Grover's Algorithm searches for the input ( $s$ ) that cause the phase-flip to occur.

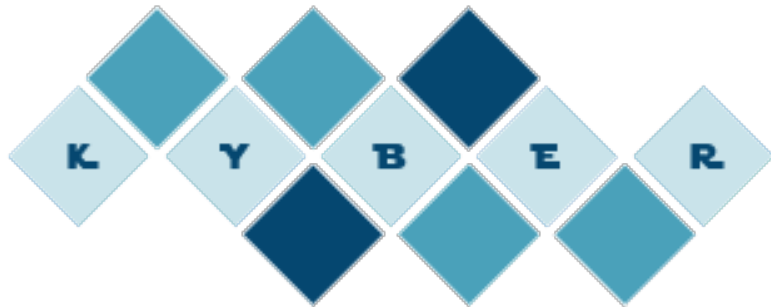


[NIST retires SHA-1 224 bit](#)

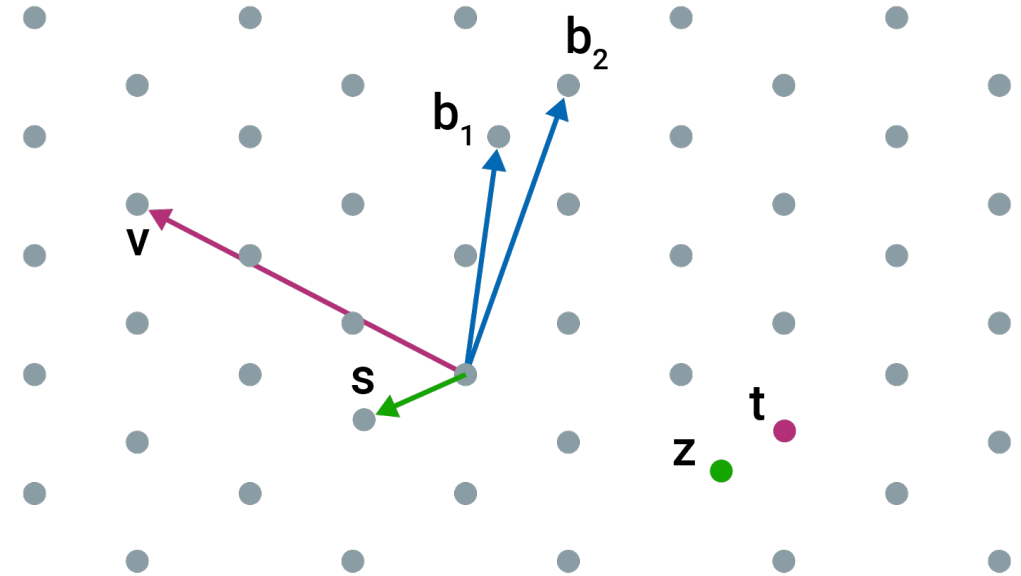
# THE SOLUTION

N.I.S.T PQC 3<sup>rd</sup> winner (KEMs)

Lattice-based



[Crystals-kyber](#)



(SK, PK, CT)

Post-Quantum Key Agreement x +

← → ↻ 🔒 pq.cloudflareresearch.com

## Cloudflare Research: Post-Quantum Key Agreement

On essentially all domains served through **Cloudflare**, including this one, we have enabled hybrid post-quantum key agreement. Read [our blog](#) for the details.

You are using `X25519Kyber768Draft00` which is **post-quantum secure**.

### Deployed key agreements

Available with TLSv1.3 including HTTP/3 (QUIC)

Key agreement	TLS identifier
X25519Kyber512Draft00	0xfe30
<b>X25519Kyber768Draft00</b>	0x6399 (recommended) and 0xfe31 (obsolete)
X25519Kyber[x]Draft00 is a <b>hybrid</b> of <b>X25519</b> and <b>Kyber[x]Draft00</b> (in that order).	

### Client support

- **Chrome Canary** (restricted to ≤HTTP/2) if you turn on *TLS 1.3 hybridized Kyber support* (`enable-tls13-kyber`) in `chrome://flags`. **[new!]**
- Our **fork of Go**.
- **BoringSSL [new!]**. Upstream only supports 0x6399; for the others use our old **fork**.
- Our **fork of QUIC-go**.
- *Goutam Tamvada's fork of Firefox*.
- **Open Quantum Safe** with the **right setting**. **[new!]**
- **Zig nightly**. **[new!]**

### Contact

You can reach us directly at [ask-research@cloudflare.com](mailto:ask-research@cloudflare.com) with questions and feedback.

**Chrome Beta (≥115.0.5748.0)**  
**supports hybrid scheme**  
**post-quantum key exchange**  
**behind a feature flag.**

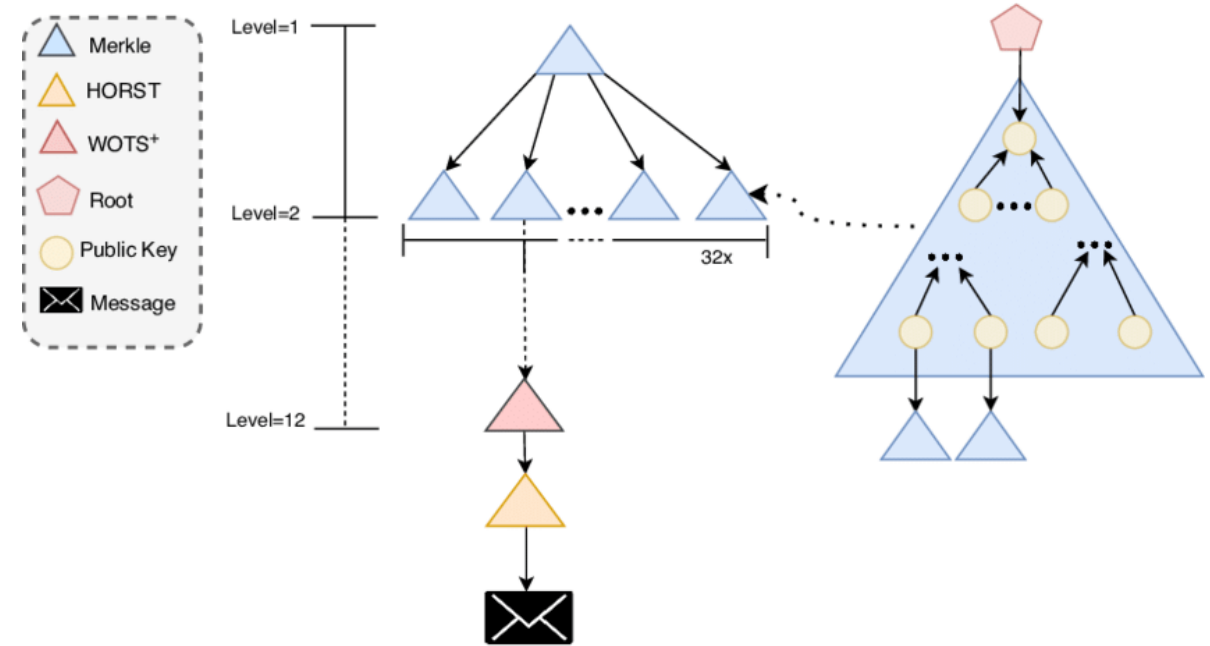
N.I.S.T PQC 4<sup>rd</sup> winner

PQC (DSA)

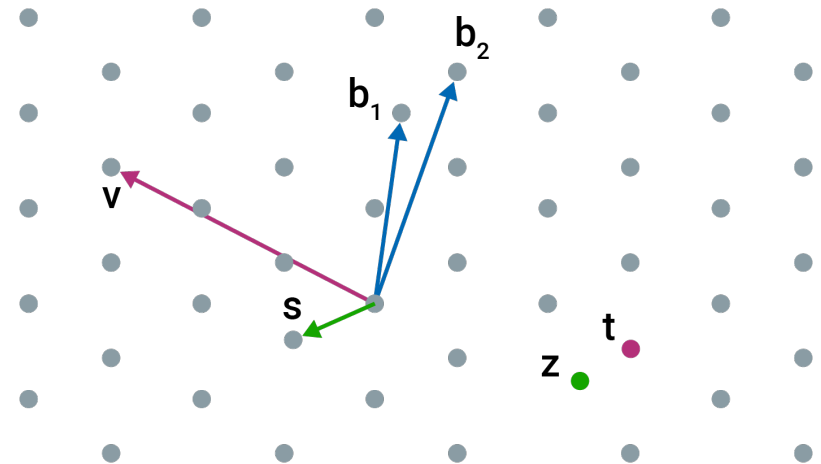
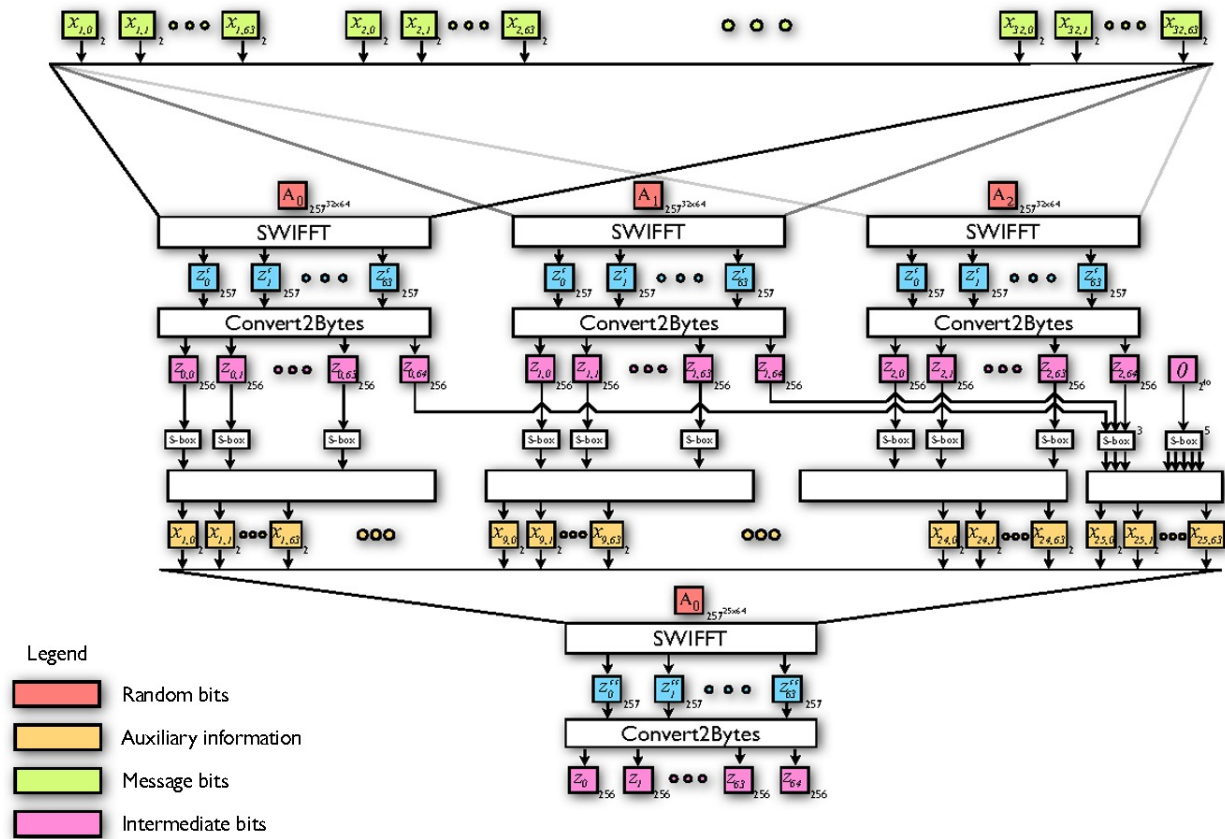
“Stateless” hash-based

# SPHINCS+

[Sphincs+](#)



Option 1  
(verify, sign, hash)



# SWIFFTX

## Lattice-based Hash function

# Quantum Cryptography aka QKD

NIST

## Using quantum technology to build cryptosystems

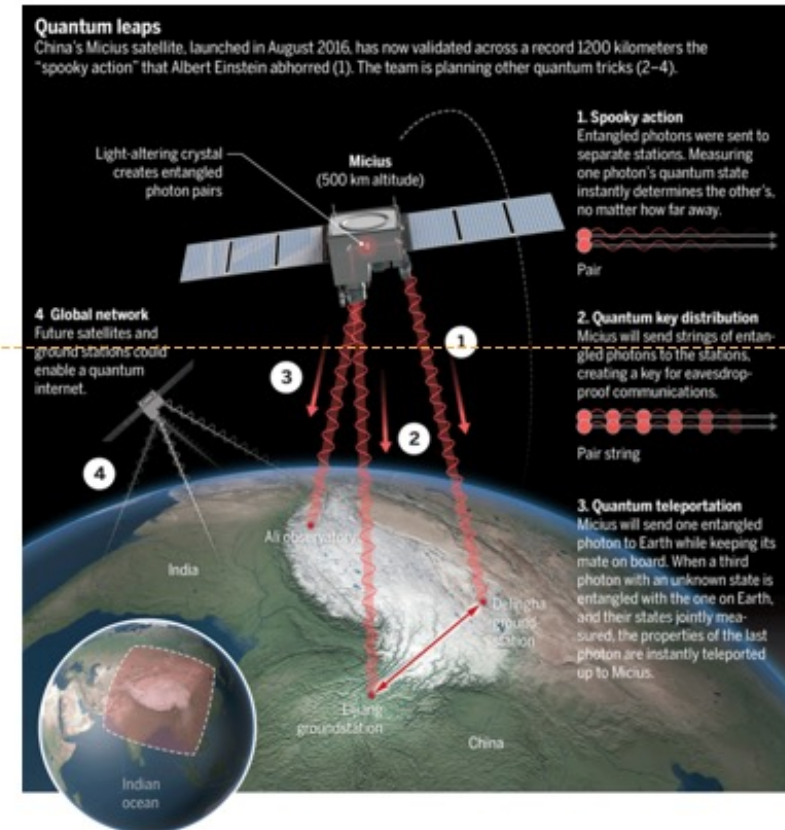
- Theoretically unconditional security guaranteed by the laws of physics

## Limitations

- Can do encryption, but not authentication
- Quantum networks not very scalable
- Expensive and needs special hardware

Lots of money being spent on “quantum”

This is NOT our focus





# SPHINX

<https://github.com/SPHINX-HUB-ORG>

