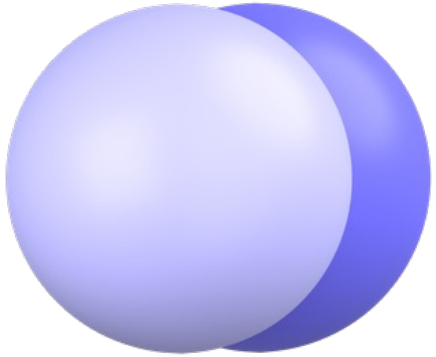
SPHINX



An open-source Post-Quantum blockchain layer 1

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

https://linktr.ee/sphinx.org

Vision

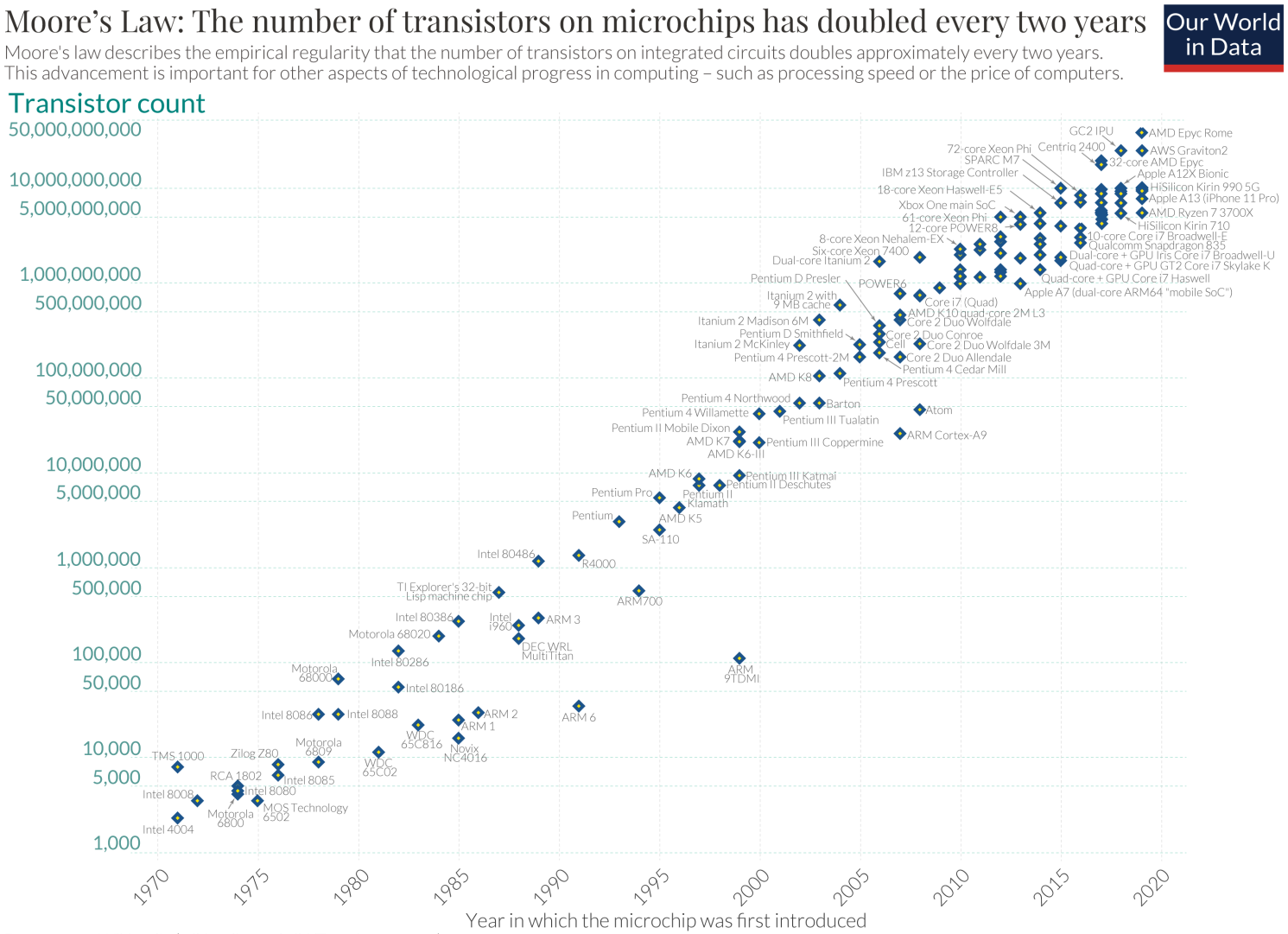
Unlocking the Power of Post-Quantum Technology;

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



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

[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.

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

Peter W. Shor†

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

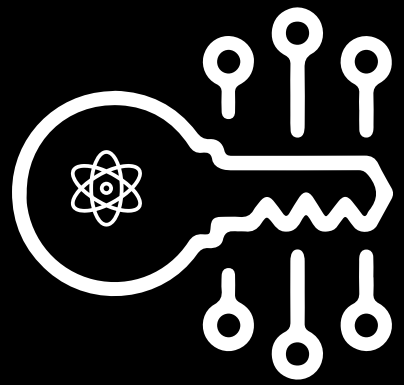
Figure 1
Quantum computing's potential for significant speedup over classical computers, according to IBM internal analysis

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

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.

UNIVERSITY OF SUSSEX



Encryption & KEM's

How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates

Daniel Litinski @ PsiQuantum, Palo Alto

We use Shor's algorithm for the computation of elliptic curve private keys as a case study for resource estimates in the silicon-photonics-inspired active-volume architecture. Here, a fault-tolerant surface-code quantum computer consists of modules with a logarithmic number of non-local inter-module connections, modifying the algorithmic cost function compared to 2D-local architectures. We find that the non-local connections reduce the cost per key by a factor of 300-700 depending on the operating regime. At 10% threshold, assuming a 10-μs code cycle and non-local connections, one key can be generated every 10 minutes using 6000 modules with 1152 physical qubits each. By contrast, a device with strict 2D-local connectivity requires more qubits and produces one key every 38 hours. We also find simple architecture-independent algorithmic modifications that reduce the Toffoli count per key by up to a factor of 5. These modifications involve reusing the stored state for multiple keys and spreading the cost of the modular division operation over mul-

dreds or thousands of physical qubits. There exist two types of general-purpose architectures for surface codes: baseline architectures with nearest-neighbor logical two-qubit operations on a 2D grid [14–18], and the recently introduced active-volume architecture [19] utilizing a logarithmic number of non-local connections between patches. The latter leverages non-local connections to parallelize the execution of logical operations, resulting in a significant speedup compared to a fault-tolerant quantum computer with the same footprint, but strict 2D-local connectivity.

Different architectures. The existing literature on FTQC resource estimates primarily focuses on baseline architectures, relying on determining logical qubit counts (n_Q) and Toffoli gate counts (n_{Tof}) due to the relevance of the cost function $n_Q \cdot n_{\mathrm{Tof}}$. Resource estimates for active-volume architectures are different, albeit not more complicated. Instead of counting qubits and gates, different fundamental subroutines are counted based on their specific costs, known as the active volume. This paper aims to guide the reader through a simplified active-volume resource estimation

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.

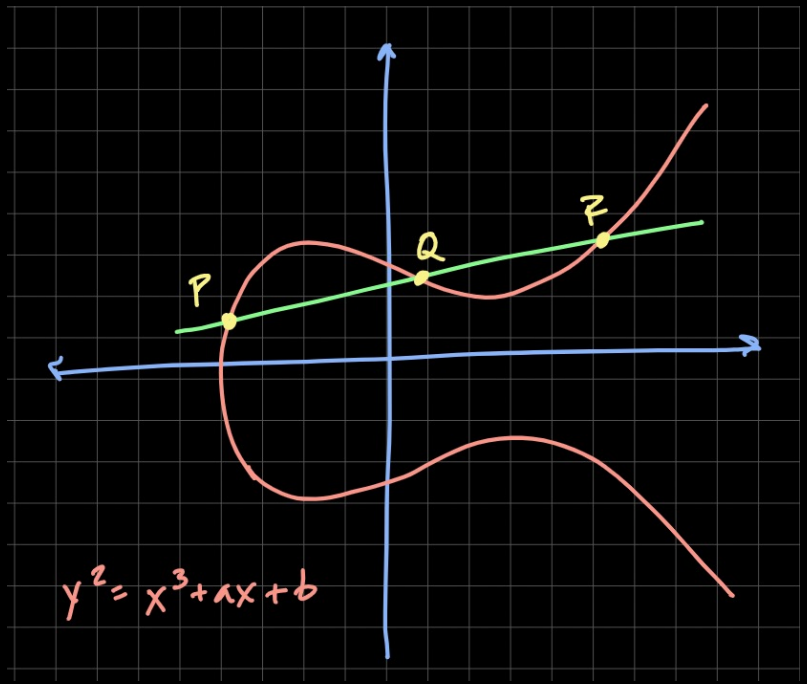
And, tying this back to ECDSA, once we can compute the discrete log of the public key, that's the private key, and that allows the attacker to sign any message he wants.

[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 known x, y and n (which we do)

source

Curve 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 224 bit

Quantum-computing landscape:

IBM Roadmap.

IBM quantum centric 100,000 qubits.

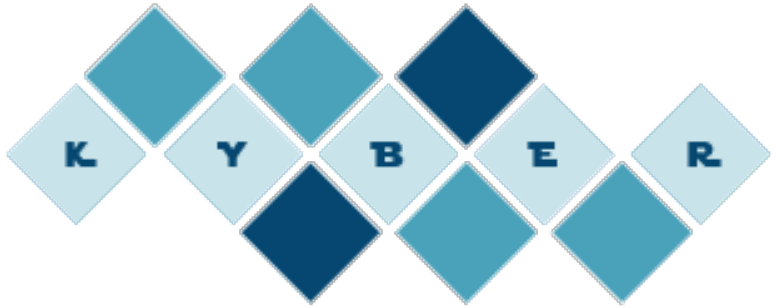
(EuroHPC JU) quantum-computers.

CISA & NSA was warn for global community the deadline to we are migrate into post-quantum era until 2035.

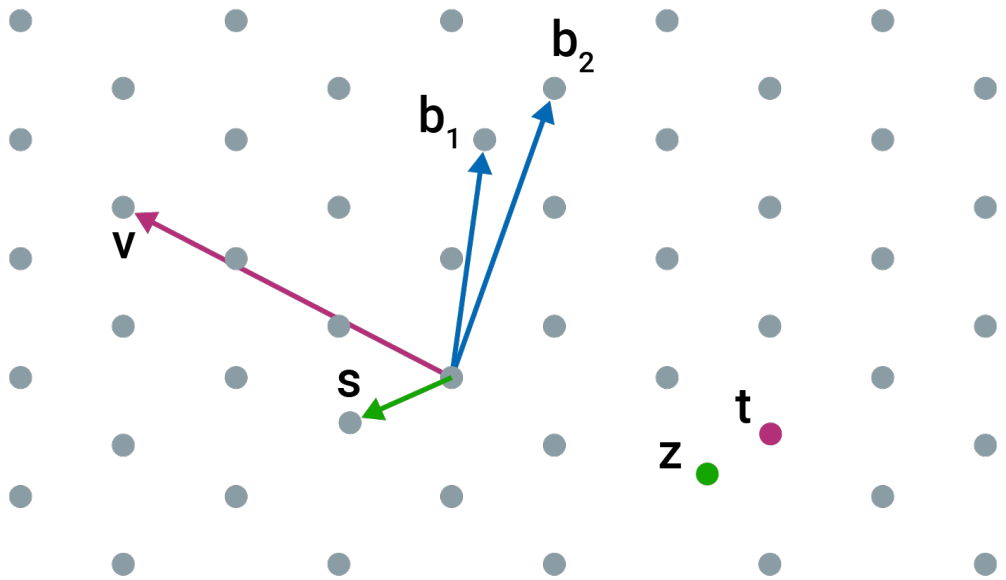
THE SOLUTION

N.I.S.T PQC 3rd winner (KEMs)

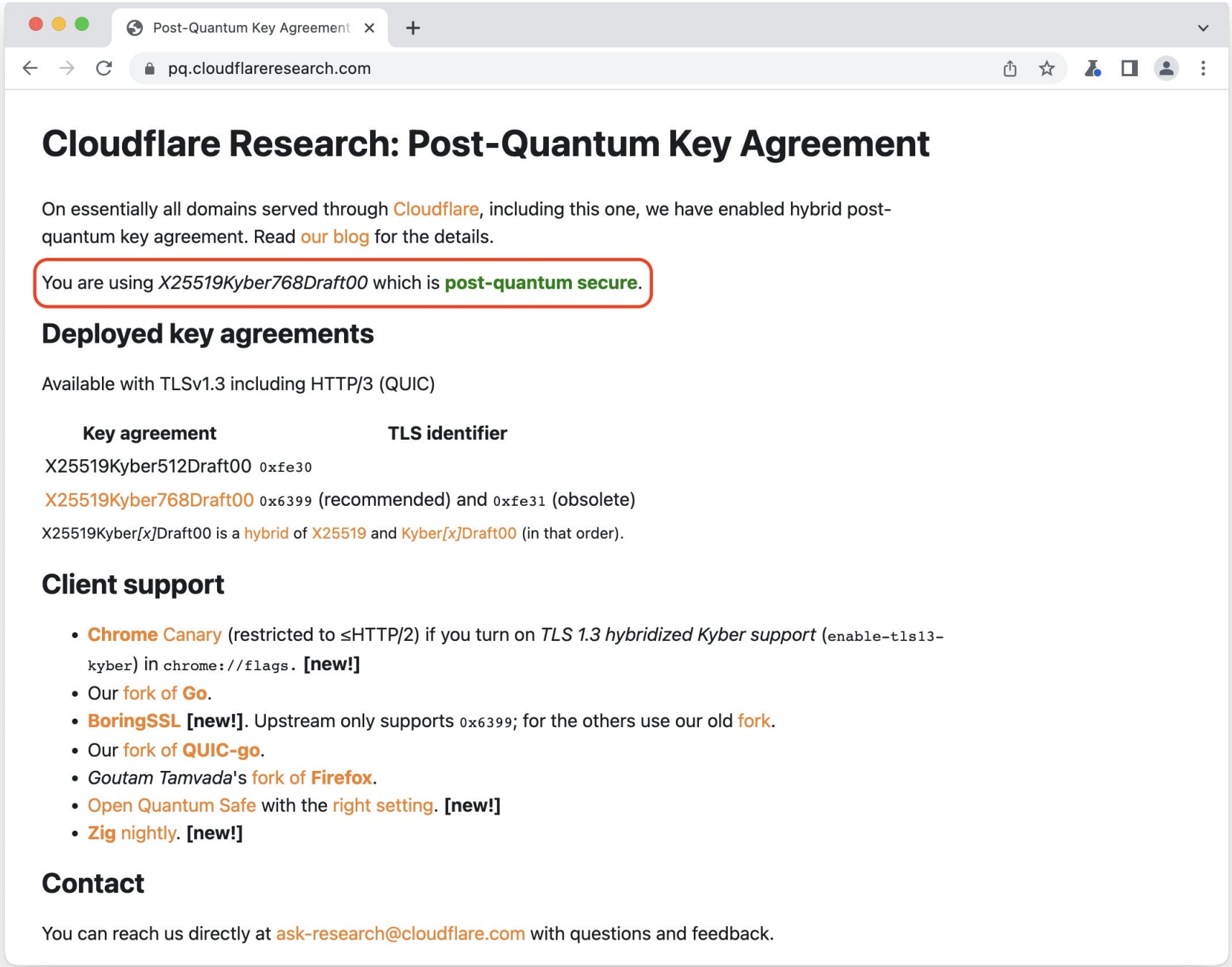
Lattice-based



Crystals-kyber



(SK, PK, CT)


Post-Quantum Key Agreement
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
X25519Kyber768Draft00 0x6399 (recommended) and 0xfe31 (obsolete)
X25519Kyber[x]Draft00 is a hybrid of X25519 and Kyber[x]Draft00 (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 with questions and feedback.

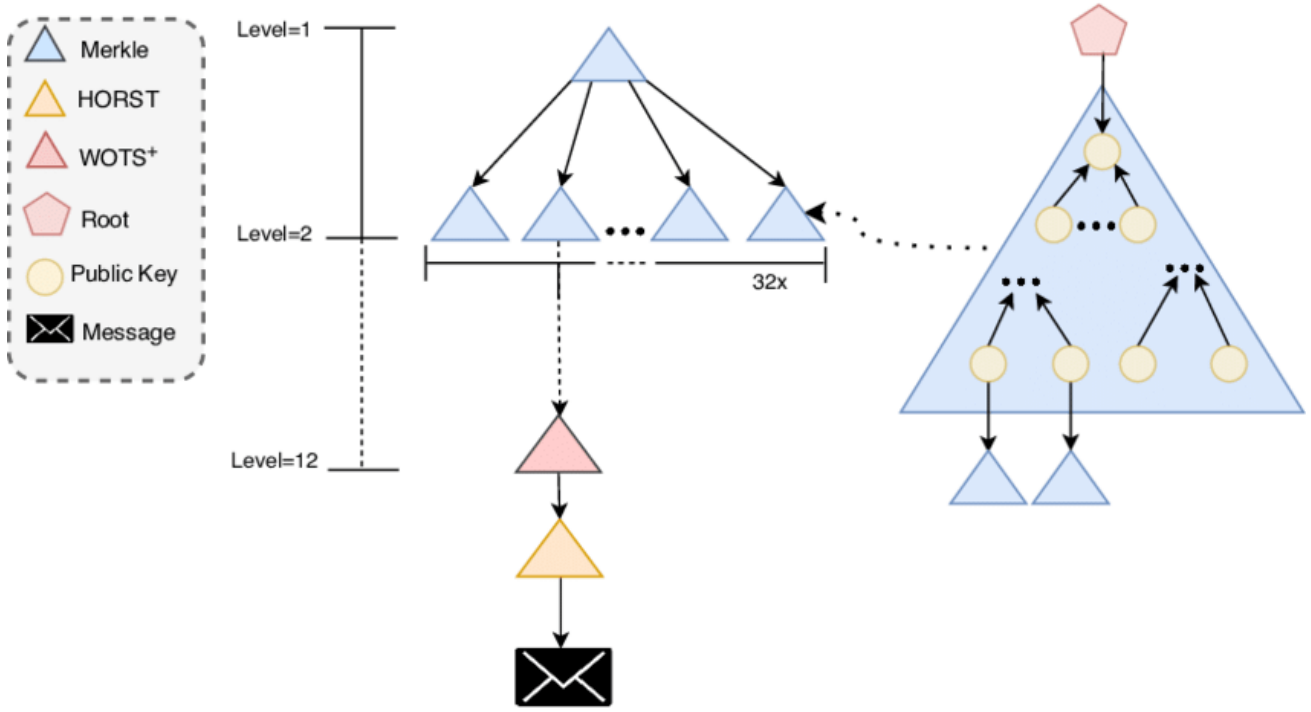
N.I.S.T PQC 4rd winner

PQC (DSA)

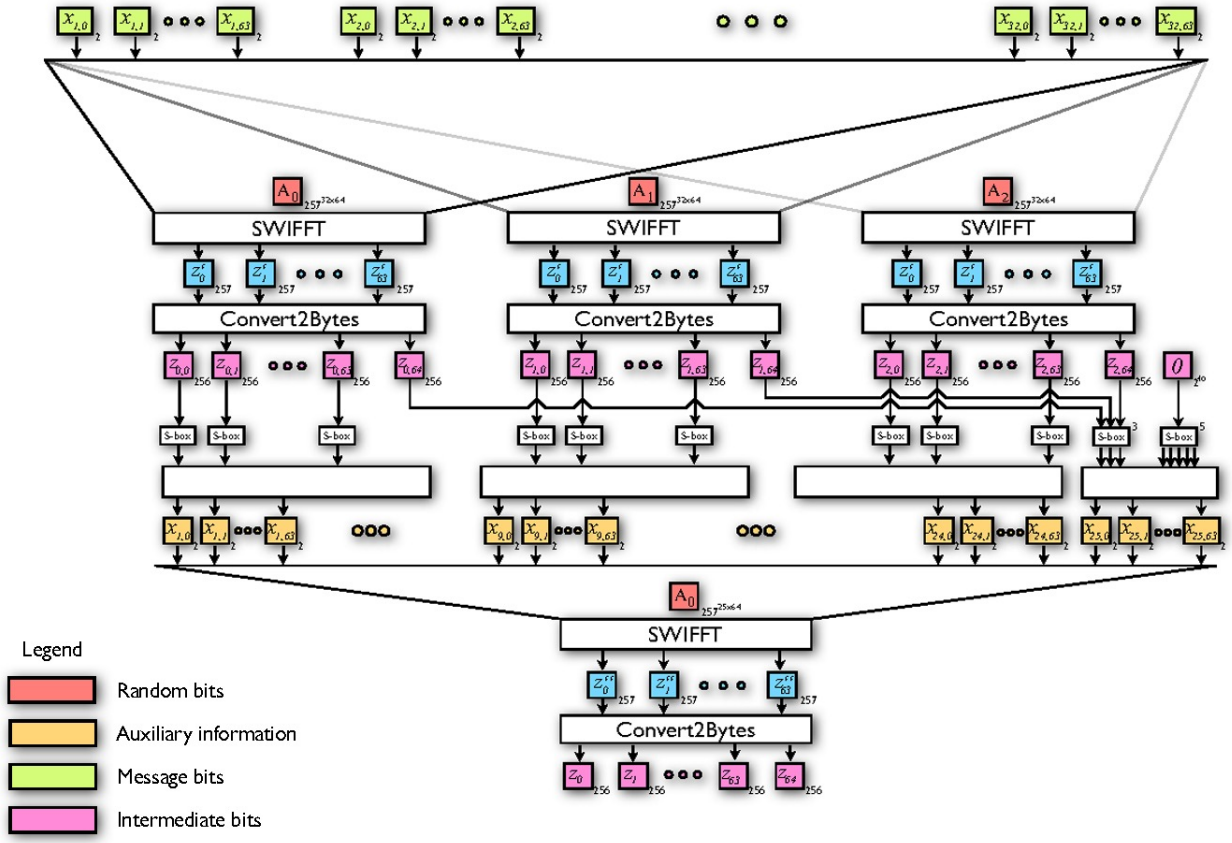
“Stateless” hash-based

SPHINCS+

Sphincs+



(verify, sign, hash)



SWIFFTX
Lattice-based
Hash function

Quantum Cryptography aka QKD

NIST

Using quantum technology to build cryptosystems

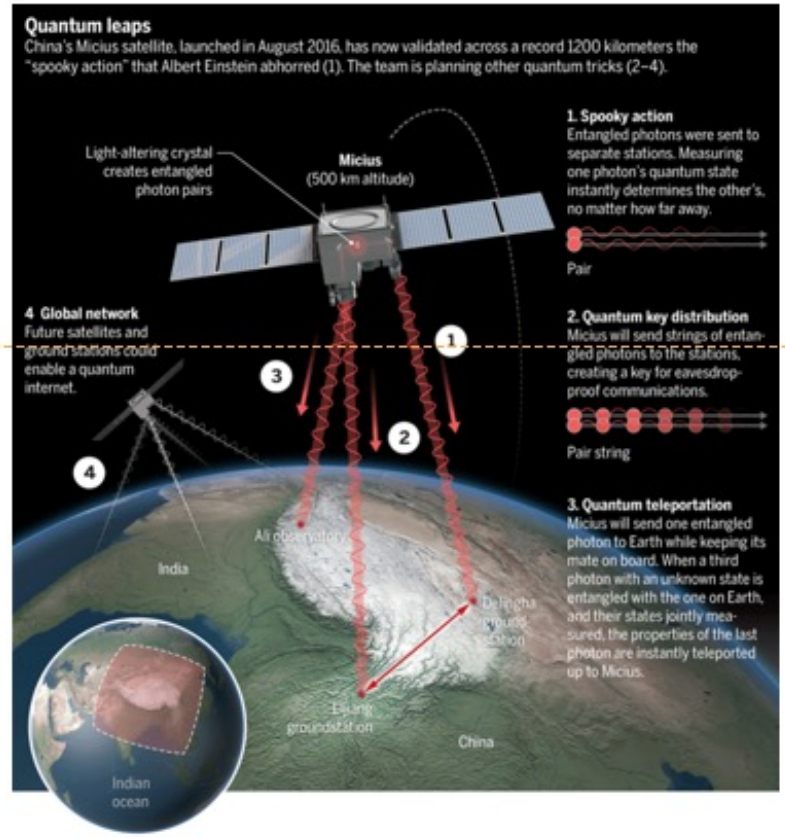
- Theoretically unconditional security guaranteed by the laws of physics

Limitations

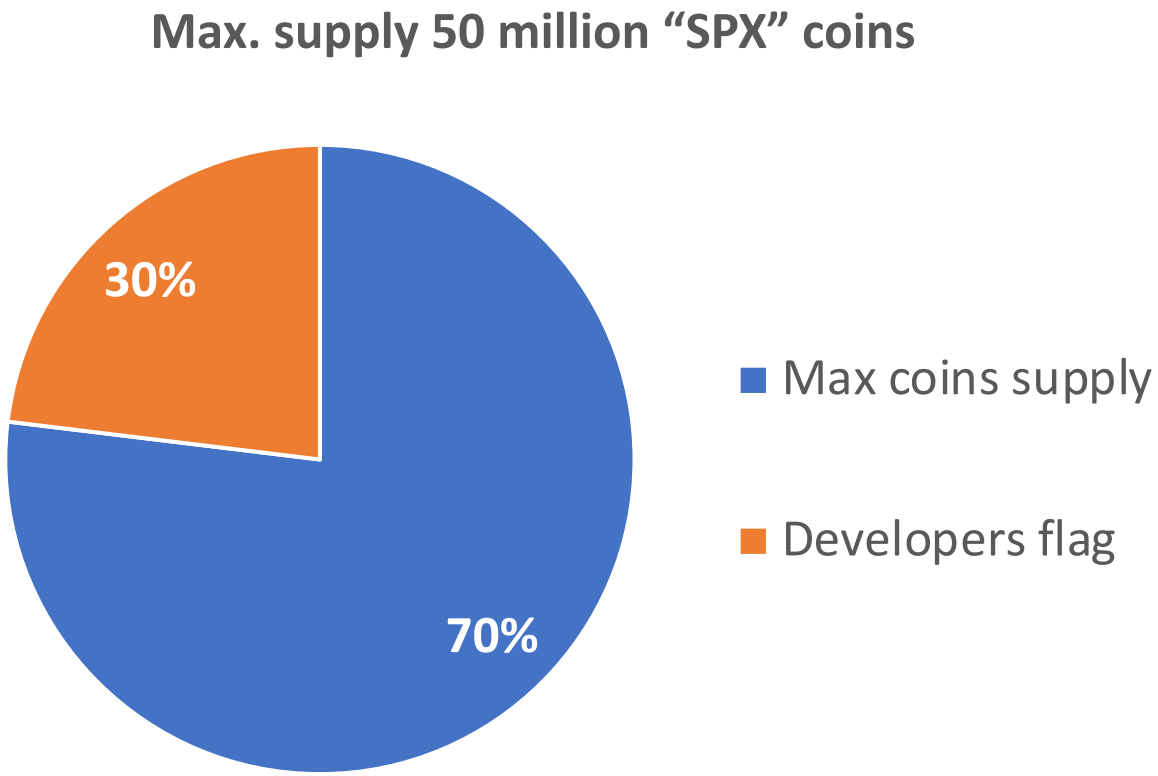
- Can do encryption, but not authentication
- Quantum networks not very scalable
- Expensive and needs special hardware

Lots of money being spent on “quantum”

This is NOT our focus



TOKENOMIC


Max. supply 50 million “SPX” coins
30%
70%
Max coins supply
Developers flag

SPHINX

For more detil just visited our workspace;

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