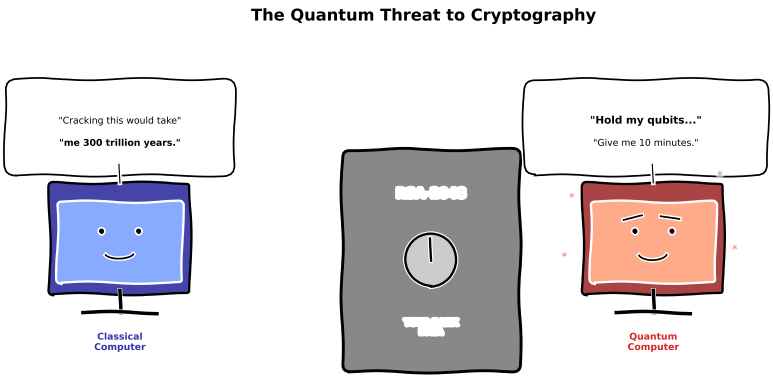
Lesson 8.2: Quantum Computing and Post-Quantum Cryptography

Module 8: The Future Problem

Prof. Dr. Joerg Osterrieder



The safe hasn't changed. The tools have.

A light moment before we dive in – humor helps learning stick.

Learning Objectives

After this lesson you will be able to:

1. **Explain** at a conceptual level what qubits, superposition, and entanglement mean — without mathematics [Understand]
2. **State** which cryptographic systems are broken by quantum computers and which are not [Remember]
3. **Describe** the "harvest now, decrypt later" threat and why migration is urgent today [Understand]
4. **Name** the three families of post-quantum cryptography selected by NIST and their tradeoffs [Remember]
5. **Outline** the four-phase migration path from classical to post-quantum cryptography [Understand]
6. **Identify** near-term quantum computing applications in finance [Understand]

Bloom's levels: Remember (2,4), Understand (1,3,5,6). Intentionally conceptual for non-quantum-physicists.

Data and Identity Require Cryptography — But That Cryptography Faces an Existential Threat

What we have built (Modules 1–7):

- Digital payments secured by RSA and elliptic curve cryptography
- Blockchain transactions signed with ECDSA
- TLS/HTTPS protecting every online banking session
- Digital identities verified by public key infrastructure
- All of modern digital finance rests on the assumption that these algorithms are unbreakable

What this lesson addresses:

- Quantum computers can break RSA and ECDSA
- This is not science fiction — it is an engineering timeline
- Adversaries are already collecting encrypted data to decrypt later
- New "post-quantum" algorithms exist and are standardized
- The migration has begun; financial institutions must act now

The cryptography protecting your bank account was designed in a world without quantum computers. That world is ending.

This lesson is conceptual. We explain what quantum computing does to cryptography, not how quantum physics works. No mathematics required.

Key Definition: The Qubit (NEW)

Qubit (Quantum Bit)

A **qubit** is the fundamental unit of quantum information. Unlike a classical bit (which is definitively 0 or 1), a qubit exists in a *superposition* — a simultaneous combination of both 0 and 1 states — until measured. Measurement forces the qubit to "collapse" to a single definite value.

The coin analogy:

- A **classical bit** is like a coin lying on a table: it is either heads (0) or tails (1). Always one or the other.
- A **qubit** is like a coin spinning in the air: while spinning, it is *both* heads and tails at the same time. Only when you catch it (measure it) does it become one definite value.
- This "both at once" property is called **superposition**.

Why this matters:

- A classical computer with n bits can represent exactly *one* number at a time
- A quantum computer with n qubits can, in a sense, explore *many* possible answers simultaneously
- For specific types of problems, this gives quantum computers an enormous speed advantage

Superposition does not mean a qubit "is" 0 and 1 simultaneously in a classical sense. It means the qubit exists in a state that has the potential to be measured as either 0 or 1, with certain probabilities.

Entanglement: Qubits That Are Connected

The paired-coin analogy:

- Imagine two coins that are magically linked: whenever you flip one and it lands heads, the other *always* lands heads too — no matter how far apart they are
- This is **entanglement**: measuring one qubit instantly determines the state of its entangled partner
- Entanglement does not transmit information faster than light, but it allows quantum computers to correlate computations across many qubits simultaneously

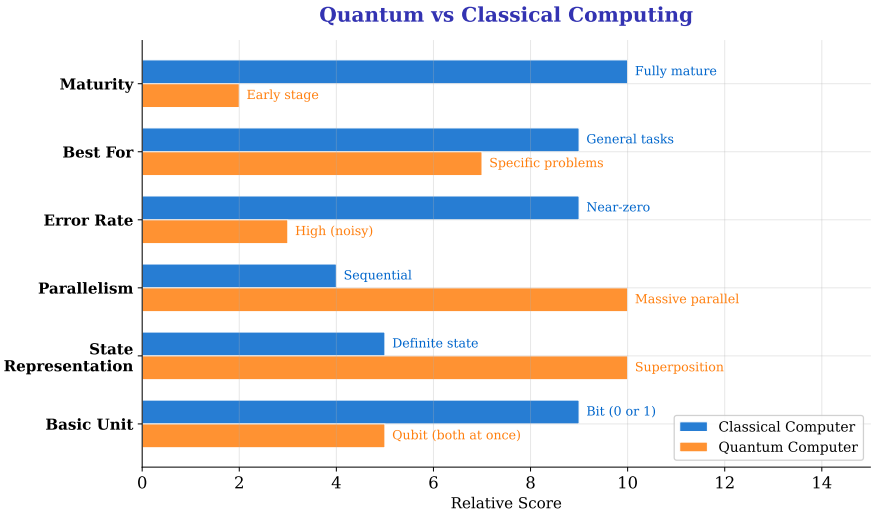
Why it matters for computing:

- Entanglement lets a quantum computer coordinate operations across all its qubits in ways a classical computer cannot replicate
- Combined with superposition, this enables certain algorithms to solve problems exponentially faster than any classical approach
- The key word is *certain*: quantum computers are not faster at everything — only at specific problem types

---

Quantum advantage is not universal. A quantum computer will not make your email faster. But for problems involving large-number factoring or optimization, the speedup can be astronomical.

Quantum vs Classical Computing



- **What you see:** Side-by-side comparison showing quantum excels at parallelism and state representation but suffers from high error rates and low maturity

Where Is Quantum Computing Today? (2026)

Current state of quantum hardware:

Metric	Current Status (2026)	Required to Break RSA-2048
Qubits	IBM Heron 156 qubits; IBM Condor 1,121 qubits; Google Willow 105 qubits with error correction (*IBM Quantum roadmap; Google Quantum AI*, 2024–2025) — physical count alone is no longer the primary metric	$\sim$4,000–20,000 *logical* qubits (millions of physical qubits)
Error rate	$\sim$0.1–1% per physical operation; Willow demonstrated below-threshold error correction	$<$0.01% (error-corrected logical)
Coherence	Microseconds to milliseconds	Hours of stable computation
Milestone	Google Willow (2024): first below-threshold error-corrected logical qubit	Orders of magnitude beyond current capability

Key takeaway: No quantum computer today can break any cryptographic system in use. But progress is accelerating, and estimates for a "cryptographically relevant" quantum computer range from 2030 to 2045.

The uncertainty is the problem: We do not know *exactly* when — but we know it is coming. And migrating cryptographic infrastructure takes years.

Estimates vary widely. Some researchers are more optimistic (2030), others more conservative (2045+). The responsible approach is to prepare now regardless of the exact timeline.

Why Cryptography Matters for Finance

Every digital financial transaction depends on cryptographic algorithms:

Financial System	Cryptography Used
Online banking (TLS/HTTPS)	RSA or ECDSA for key exchange; AES for encryption
SWIFT interbank messaging	RSA signatures for message authentication
Credit/debit card payments (EMV)	RSA or ECDSA for card authentication
Bitcoin & Ethereum	ECDSA for transaction signatures
Digital identity (eIDAS, KYC)	RSA or ECDSA certificates
Secure email (S/MIME)	RSA encryption and signatures
CBDC prototypes	Various public-key schemes

Common thread: Nearly all of these systems rely on **RSA** or **ECDSA** — both of which are broken by a sufficiently powerful quantum computer.

AES (symmetric encryption) and SHA-256 (hashing) are weakened by quantum computers but not broken. The critical threat is to public-key cryptography: RSA, ECDSA, and Diffie-Hellman.

Shor's Algorithm — What It Breaks

In 1994, mathematician Peter Shor proved that a quantum computer can:

1. **Factor large numbers** exponentially faster than any known classical algorithm
2. **Solve the discrete logarithm problem** — the mathematical foundation of elliptic curve cryptography

What this breaks:

- **Rivest-Shamir-Adleman (RSA):** Security relies on the difficulty of factoring the product of two large primes. Shor's algorithm makes this easy.
- **Elliptic Curve Digital Signature Algorithm (ECDSA):** Security relies on the discrete logarithm problem on elliptic curves. Shor's algorithm solves this.
- **Diffie-Hellman:** Key exchange relies on the same discrete logarithm assumption. Broken.

You do not need to understand how Shor's algorithm works. The important fact is: it exists, it is mathematically proven, and it will break RSA/ECDSA once a sufficiently large quantum computer is built.

Shor's Algorithm — What Survives

What Shor's algorithm does NOT break:

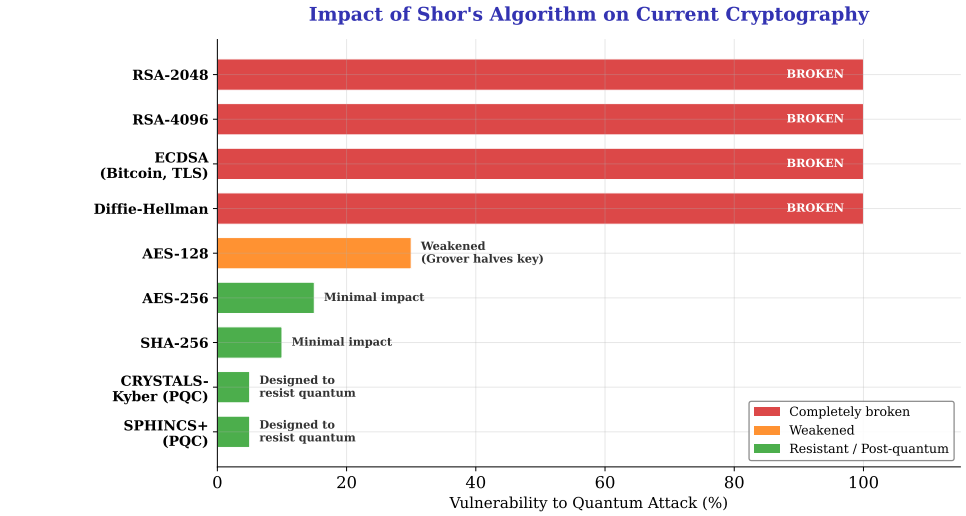
- **AES-256:** Grover's algorithm halves the effective key length (256 $\rightarrow$ 128-bit security), but 128-bit is still sufficient
- **SHA-256:** Grover provides a square-root speedup for finding collisions, but SHA-256 remains computationally hard

Summary:

Algorithm	Type	Quantum Impact
RSA	Public-key	Completely broken
ECDSA	Public-key	Completely broken
Diffie-Hellman	Key exchange	Completely broken
AES-256	Symmetric	Weakened (still usable)
SHA-256	Hash	Weakened (still usable)

Public-key cryptography (RSA, ECDSA, Diffie-Hellman) is completely broken. Symmetric cryptography (AES) and hash functions (SHA-256) are weakened but remain usable with larger key sizes.

Which Cryptographic Systems Are at Risk?



- **What you see:** RSA and ECDSA show 100% vulnerability (red); AES shows 15-30% impact (weakened but not broken); post-quantum algorithms (CRYSTALS-Kyber, SPHINCS+) show minimal vulnerability

"Harvest Now, Decrypt Later" — The Urgent Threat

Definition: Harvest now, decrypt later (HNDL) is an attack strategy where adversaries collect encrypted data today with the intention of decrypting it once quantum computers become available.

Why this matters right now:

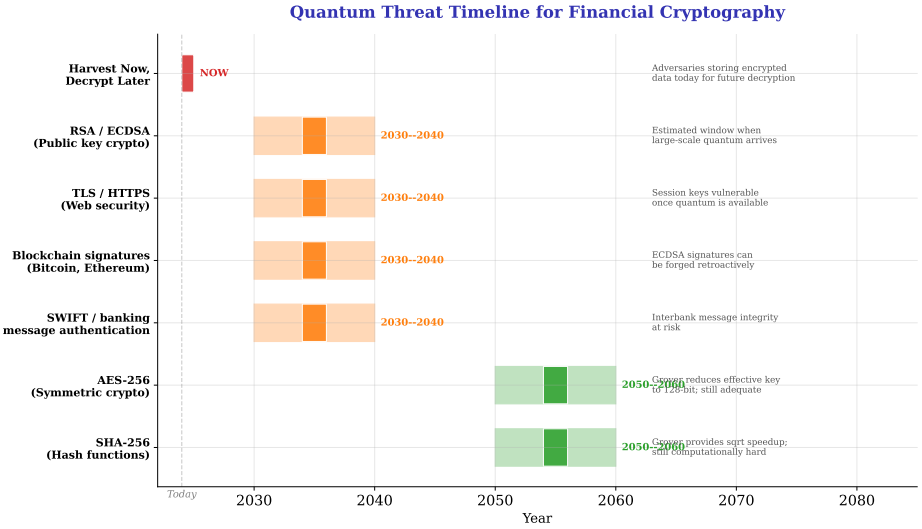
- Encrypted data transmitted today over TLS can be intercepted and stored
- Financial data, medical records, government secrets, and corporate communications are being collected by state-level actors
- If quantum computers arrive in 2035, any data with a confidentiality requirement beyond 10 years is at risk *today*
- Mortgage data, pension records, health insurance claims, and trade secrets all have multi-decade shelf lives

The timeline equation:

$$\textit{If (years until quantum)} < \textit{(years data must stay secret)} + \textit{(years to migrate)}$$
$$\Rightarrow \textit{you are already too late.}$$

HNDL is not a theoretical risk. Intelligence agencies and other state actors are widely believed to be stockpiling encrypted traffic today. The NSA has publicly urged migration to post-quantum cryptography.

Quantum Threat Timeline



- **What you see:** Timeline showing harvest-now-decrypt-later is active today (red); RSA/ECDSA face 2030-2040 window (orange); AES-256 and SHA-256 remain safe until 2050+ (green).

What Is Post-Quantum Cryptography (PQC)?

Definition: Post-quantum cryptography (PQC) refers to cryptographic algorithms that are designed to be secure against attacks by both classical *and* quantum computers.

Key clarifications:

- PQC algorithms run on **classical computers** — they do not require quantum hardware
- They are based on mathematical problems that Shor's algorithm cannot solve
- They are "drop-in replacements" for RSA and ECDSA in most protocols (TLS, S/MIME, code signing)
- The term "post-quantum" means "secure in a world where quantum computers exist" — not "requires quantum computers"

Three main families:

1. **Lattice-based:** Based on the difficulty of finding short vectors in high-dimensional lattices
2. **Hash-based:** Based on the security of cryptographic hash functions (e.g., SHA-256)
3. **Code-based:** Based on the difficulty of decoding random linear error-correcting codes

---

PQC is not exotic. It is conventional cryptography based on different hard mathematical problems. The key innovation is choosing problems that quantum computers cannot solve efficiently.

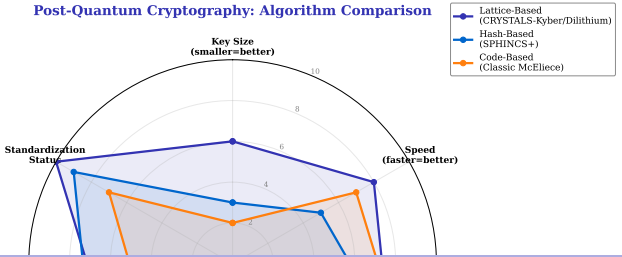
Classical vs. Post-Quantum Cryptography: Key Differences

Classical Public-Key Crypto:

- **RSA:** Based on integer factorization
- **ECDSA:** Based on elliptic curve discrete logarithm
- **Key sizes:** 2048–4096 bits (RSA), 256–384 bits (ECDSA)
- **Performance:** Fast key generation, moderate signature size
- **Quantum vulnerability:** Completely broken by Shor's algorithm

Post-Quantum Crypto:

- **Lattice (Kyber, Dilithium):** Based on lattice hardness
- **Hash (SPHINCS+):** Based on hash function security
- **Key sizes:** 800–1600 bytes (lattice), larger for others
- **Performance:** Comparable to RSA; larger signatures
- **Quantum security:** No known quantum attack



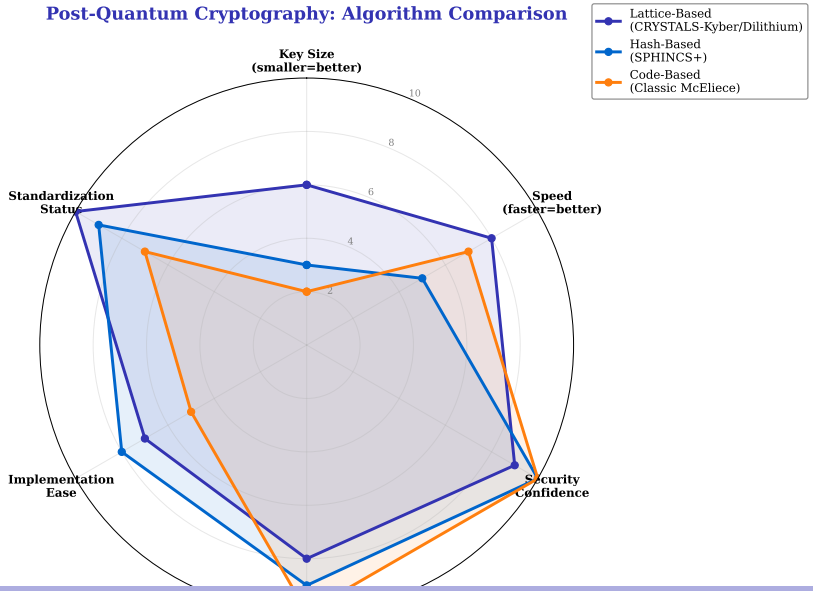
The Three PQC Families (Named, Not Derived)

Family	Hard Problem	NIST Standard	Tradeoff
Lattice-based	Finding short vectors in high-dimensional mathematical grids	CRYSTALS-Kyber (encryption), CRYSTALS-Dilithium (signatures)	Small keys, fast; relatively new mathematical foundation
Hash-based	Reversing cryptographic hash functions	SPHINCS+ (signatures)	Very high confidence; large signatures, slower
Code-based	Decoding random error-correcting codes	Classic McEliece (encryption, round 4)	Extremely well-studied (since 1978); very large public keys

Why multiple families? If one mathematical problem turns out to be easier to solve than expected, the other families provide a fallback. Diversification in cryptography is a form of risk management.

You do not need to understand lattices, hash trees, or error-correcting codes. The key fact is: NIST has evaluated these for 8 years and selected them as the replacements for RSA and ECDSA.

PQC Algorithm Comparison



NIST Post-Quantum Cryptography Standardization

The process: In 2016, the U.S. National Institute of Standards and Technology (NIST) launched a public competition to select quantum-resistant algorithms. This mirrors the process that produced AES in 2001.

Selected algorithms (finalized 2024):

1. **FIPS 203 — ML-KEM (CRYSTALS-Kyber):** Key encapsulation mechanism for secure key exchange. Replaces RSA/ECDH in TLS, VPNs, and messaging.
2. **FIPS 204 — ML-DSA (CRYSTALS-Dilithium):** Digital signature algorithm. Replaces RSA/ECDSA for document signing, code signing, certificates.
3. **FIPS 205 — SLH-DSA (SPHINCS+):** Hash-based digital signature. Backup signature scheme with conservative security assumptions.

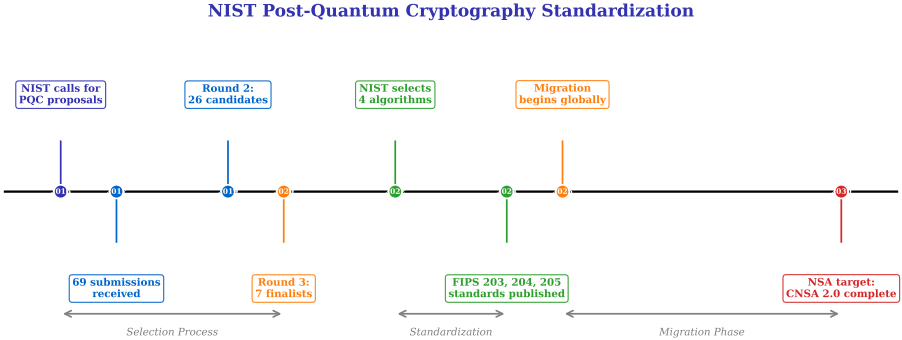
Still under evaluation (Round 4):

- Classic McEliece, BIKE, HQC — additional encryption schemes for algorithm diversity

---

FIPS = Federal Information Processing Standard. These are mandatory for U.S. government systems and widely adopted as global benchmarks. European agencies (BSI, ANSSI) are also endorsing these algorithms.

NIST PQC Standardization Timeline



- **What you see:** 8-year process from 2016 call for proposals (69 submissions) to 2024 FIPS standards (4 algorithms selected), followed by 2025-2030 migration phase
- **Key pattern:** Selection took 6 years; migration is expected to take another 5-10 years — total transition spans 15+ years from start to finish
- **Takeaway:** NSA mandates federal agencies complete migration by 2030-2035; financial institutions should begin hybrid deployment now

Eight years from call for proposals to published standards. The migration window is now: organizations should begin deploying these algorithms in

What Is Crypto Agility?

Definition: Crypto agility is the ability of a system to switch between cryptographic algorithms without redesigning the entire system.

Why it matters:

- Most financial systems today have cryptographic algorithms **hard-coded** into their infrastructure
- Changing from RSA to a PQC algorithm requires updating: TLS libraries, certificate authorities, HSMs (hardware security modules), key management systems, and compliance documentation
- A crypto-agile system abstracts the algorithm choice behind an interface, so swapping algorithms is a configuration change rather than a rewrite

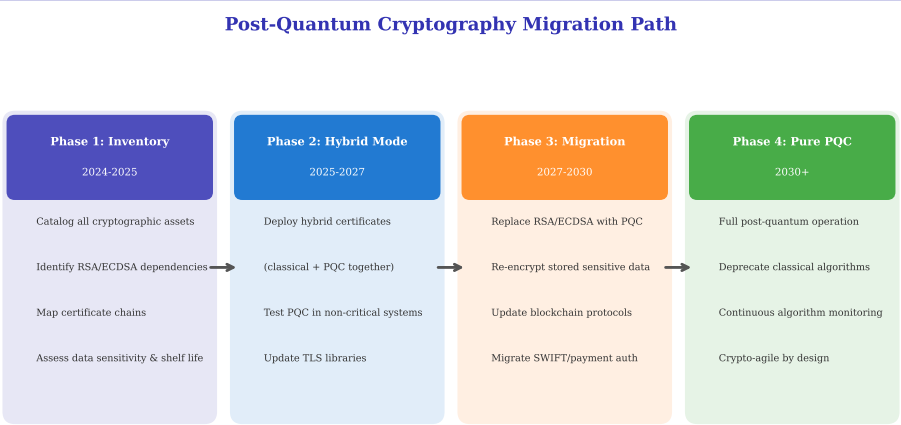
Crypto agility in practice:

1. **Protocol negotiation:** TLS already supports algorithm negotiation; PQC cipher suites can be added
2. **Hybrid certificates:** A single certificate contains both a classical (RSA) and a PQC (Kyber) key pair — systems that support PQC use it; legacy systems fall back to RSA
3. **Algorithm registries:** Central configuration of which algorithms are approved, with the ability to deprecate algorithms rapidly

---

Crypto agility is the single most important architectural decision for long-term cryptographic security. Systems designed today without crypto agility will face costly rewrites within a decade.

The Four-Phase Migration Path



Organizations that start Phase 1 today protect data that must remain confidential for 10+ years.

- **What you see:** Four sequential phases spanning 2024-2030+, from inventory (catalog cryptographic assets) through hybrid deployment to full PQC migration

Quantum Computing Applications in Finance

Quantum computers are not only a threat — they also offer potential benefits for finance:

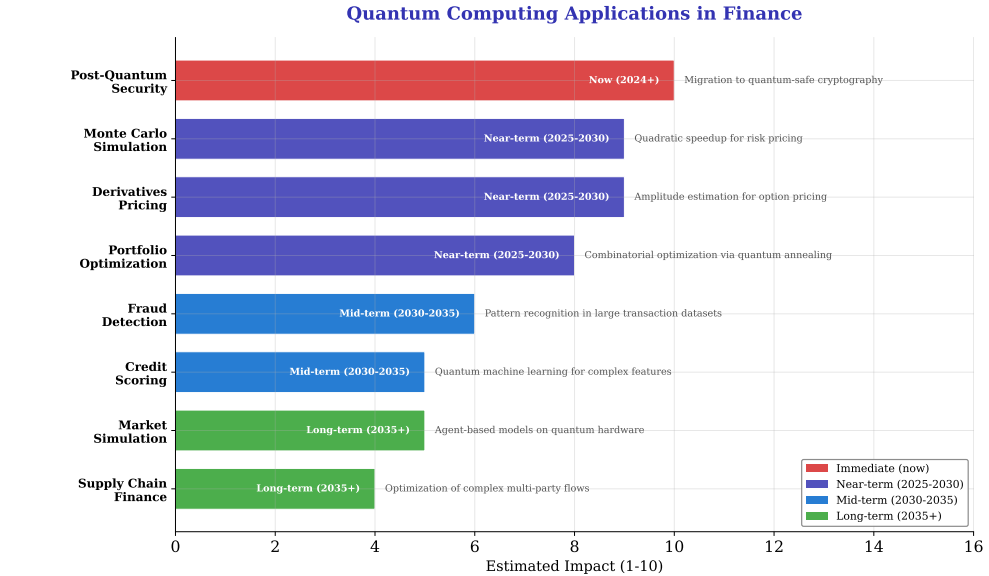
- **Portfolio optimization:** Quantum annealing can explore vast combinations of asset allocations to find optimal portfolios faster than classical optimizers (near-term, hybrid quantum-classical)
- **Monte Carlo simulation:** Quantum amplitude estimation offers a quadratic speedup for pricing derivatives and calculating risk measures like VaR (near-term research)
- **Fraud detection and pattern recognition:** Quantum machine learning may identify complex fraud patterns in large transaction datasets (longer-term, unproven at scale)

Industry activity:

- JPMorgan, Goldman Sachs, HSBC, and Barclays have quantum computing research teams
- IBM and Google offer cloud-based quantum computing access (IBM Quantum, Google Quantum AI)
- The European Investment Bank issued the first quantum-secured bond communication in 2023

Quantum finance applications are early-stage. No quantum computer today outperforms classical systems for any real financial problem. But R&D investment is substantial and accelerating.

Quantum Finance: Applications by Timeline and Impact



Impact on Blockchain and Cryptocurrencies

Blockchain is especially vulnerable because transactions are public and permanent:

Specific risks:

- **ECDSA signatures:** Bitcoin and Ethereum use ECDSA. A quantum computer could forge signatures and spend anyone's coins.
- **Exposed public keys:** Bitcoin addresses that have been used (public key visible on-chain) are immediately vulnerable. Unused addresses (only hash visible) have a longer runway.
- **Immutability problem:** Blockchain's greatest strength — immutability — becomes a weakness. Historical transactions cannot be retroactively re-signed with PQC algorithms.

Migration challenges:

- Bitcoin has no built-in upgrade mechanism for signature algorithms; requires a hard fork
- Ethereum's account model makes migration somewhat easier (contract upgrades possible)
- All users must migrate their funds to PQC-secured addresses before quantum arrival
- Coins in "lost wallets" (estimated 3–4 million BTC) become vulnerable and potentially stealable

Current response: Ethereum researchers are exploring account abstraction with PQC signatures. Bitcoin's path is less clear due to its conservative upgrade philosophy.

The blockchain quantum threat is unique: you cannot retroactively protect assets on an immutable ledger. Migration requires users to actively move funds to new address types.

Regulatory and Industry Response

Governments and regulators are taking the quantum threat seriously:

Actor	Action
NSA (USA)	CNSA 2.0 (2022): mandates PQC migration for national security systems by 2030–2035
NIST (USA)	Published FIPS 203/204/205 PQC standards in 2024
White House	Executive order (2022) requiring federal agencies to inventory cryptographic systems
BSI (Germany)	Recommends hybrid (classical + PQC) certificates for government systems
ANSSI (France)	Published PQC migration guidance; recommends hybrid approach
ECB	Assessing PQC implications for the digital euro and TARGET2
BIS	Published report on quantum threats to financial infrastructure (2024)
SWIFT	Testing PQC integration for interbank messaging

Key trend: Regulators are moving from "awareness" to "mandates." Financial institutions that have not begun a cryptographic inventory will face compliance pressure within 2–3 years.

The regulatory signal is clear: post-quantum migration is not optional. It is becoming a compliance requirement for critical financial infrastructure.

Migration Priorities — Inventory and Hybrid Deployment

What should a bank or financial institution do today?

1. **Cryptographic inventory (immediate):**
   - Catalog every system, protocol, and certificate using RSA, ECDSA, or Diffie-Hellman
   - Identify data with long confidentiality requirements (10+ years)
   - Map dependencies: which vendor systems, HSMs, and APIs depend on specific algorithms?
2. **Hybrid deployment (2025–2027):**
   - Deploy hybrid TLS certificates (classical + PQC) for external-facing systems
   - Test PQC algorithms in non-critical environments
   - Engage vendors on their PQC roadmaps (HSM vendors, core banking providers, cloud providers)

Step 1 is free and can start today. Most organizations discover they have far more cryptographic dependencies than expected — starting the inventory is the hardest part.

Migration Priorities — Full Migration (2027–2033)

3. **Full migration (2027–2033):**
   - Replace classical algorithms with PQC across all systems
   - Re-encrypt stored sensitive data with PQC-protected keys
   - Update all digital certificates and key management infrastructure

Key planning considerations:

- Migration is a multi-year program, not a single project
- Vendor readiness varies: cloud providers lead, core banking lags
- Hybrid mode (classical + PQC) provides backward compatibility during transition
- Budget early: cryptographic migration touches every system in the organization

---

Full migration will take 5–8 years for large institutions. Starting with the inventory and hybrid phase now buys time without disrupting operations.

The Cost of Inaction

Why "wait and see" is the wrong strategy:

If you start migration now:

- Gradual, planned transition over 5–8 years
- Hybrid mode provides backward compatibility
- Time to test, validate, and train staff
- Regulatory compliance built incrementally
- Data collected today remains protected

If you wait until quantum arrives:

- Emergency migration under time pressure
- All systems must change simultaneously
- Data collected in the interim is exposed (HNDL)
- Vendor capacity constrained (everyone migrates at once)
- Regulatory penalties for non-compliance

Historical parallel: The Y2K migration (1995–1999) cost an estimated $300–600 billion globally. PQC migration is a comparable infrastructure-level transition, but with a less certain deadline — which makes procrastination more dangerous, not less.

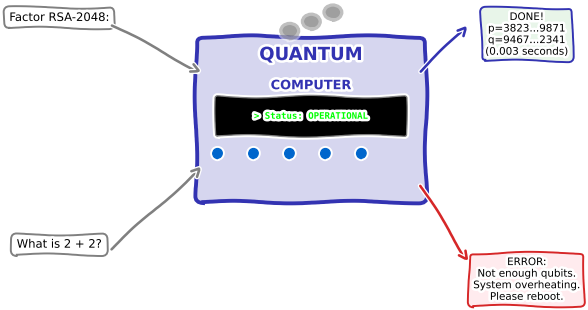
The asymmetry is clear: early migration is cheap and orderly; late migration is expensive and chaotic. The "harvest now, decrypt later" threat means data is already at risk.

Common Misconceptions About Quantum Computing

Misconception	**Reality**
"Quantum computers are just faster classical computers"	Quantum computers are fundamentally different. They are faster only for specific problem types (factoring, optimization, simulation).
"Quantum computers will break all encryption"	Only public-key cryptography (RSA, ECDSA) is broken. Symmetric encryption (AES-256) and hash functions (SHA-256) remain secure with larger parameters.
"The threat is decades away — no need to act"	The "harvest now, decrypt later" attack is happening today. Data with long shelf life is already at risk.
"My organization is too small to be targeted"	State-level actors perform mass data collection. All encrypted internet traffic is a target.
"Bitcoin is safe because it uses SHA-256 for mining"	Bitcoin mining uses SHA-256 (safe), but *transactions* use ECDSA (broken by quantum). The signature scheme, not the mining algorithm, is the vulnerability.

Misconceptions delay action. The most dangerous misconception is that quantum computing is a distant future problem — the migration window is closing.

One Last Thought...



Quantum computers: Phenomenally powerful at factoring large primes, baffled by basic arithmetic.

Sometimes the best way to remember a concept is to laugh about it.

Key Takeaways

1. **Quantum computers** use qubits (superposition, entanglement) to solve specific problems exponentially faster than classical computers — but they are not universally faster
2. **Shor's algorithm** will break RSA, ECDSA, and Diffie-Hellman — the cryptographic foundations of all digital finance. AES-256 and SHA-256 survive with larger parameters
3. **"Harvest now, decrypt later"** means the threat is not future — adversaries are collecting encrypted financial data today for quantum decryption later
4. **Post-quantum cryptography (PQC)** — lattice-based, hash-based, and code-based algorithms — has been standardized by NIST (FIPS 203/204/205) and runs on classical hardware
5. **Crypto agility** is essential: systems must be designed to swap algorithms without full redesign. Hybrid certificates (classical + PQC) enable a gradual transition
6. **Blockchain faces unique risks** because immutable ledgers cannot retroactively upgrade signature schemes. Users must proactively migrate funds to PQC-secured addresses
7. **Financial institutions must act now:** cryptographic inventory (free), hybrid deployment (2025–2027), full migration (2027–2033). Waiting is the most expensive option

---

The quantum threat is real, the solutions exist, and the migration has begun. The question is not whether to migrate — it is whether to do it now (orderly) or later (chaotic).

Summary / Next Lesson Preview

This lesson:

- Explained qubits, superposition, and entanglement using conceptual analogies
- Stated which cryptographic systems quantum computers will break (RSA, ECDSA) and which survive (AES-256, SHA-256)
- Analyzed the "harvest now, decrypt later" threat and its urgency for financial data
- Named the three PQC families and the NIST standardization outcome
- Outlined the four-phase migration path and the concept of crypto agility
- Identified near-term quantum computing applications in finance (portfolio optimization, Monte Carlo, derivatives pricing)

What comes next in Module 8 – The Future Problem:

- How climate risk and ESG considerations are reshaping financial technology
- Sustainable finance infrastructure: carbon markets, green bonds, ESG data
- The intersection of financial regulation and environmental policy

---

Quantum computing represents a cryptographic paradigm shift. The next lesson examines another paradigm shift: the integration of climate and environmental risk into the financial system.

Self-Check: Quantum Computing

Attempt these before turning the page.

1. [Remember] Which of these are broken by Shor's algorithm: RSA-2048, SHA-256, AES-256, ECDSA-P256? Which by Grover's?
2. [Understand] Explain "harvest now, decrypt later" in one sentence. Why does it make 2025 post-quantum migration urgent even without a quantum computer yet existing?
3. [Analyze] NIST's Dilithium (signatures) adds 10x signature size vs ECDSA. For a blockchain that stores signatures on-chain, estimate the storage cost impact over 1 year at 300M transactions/day.

Solutions hidden unless `\solutionstrue` is set before compiling.