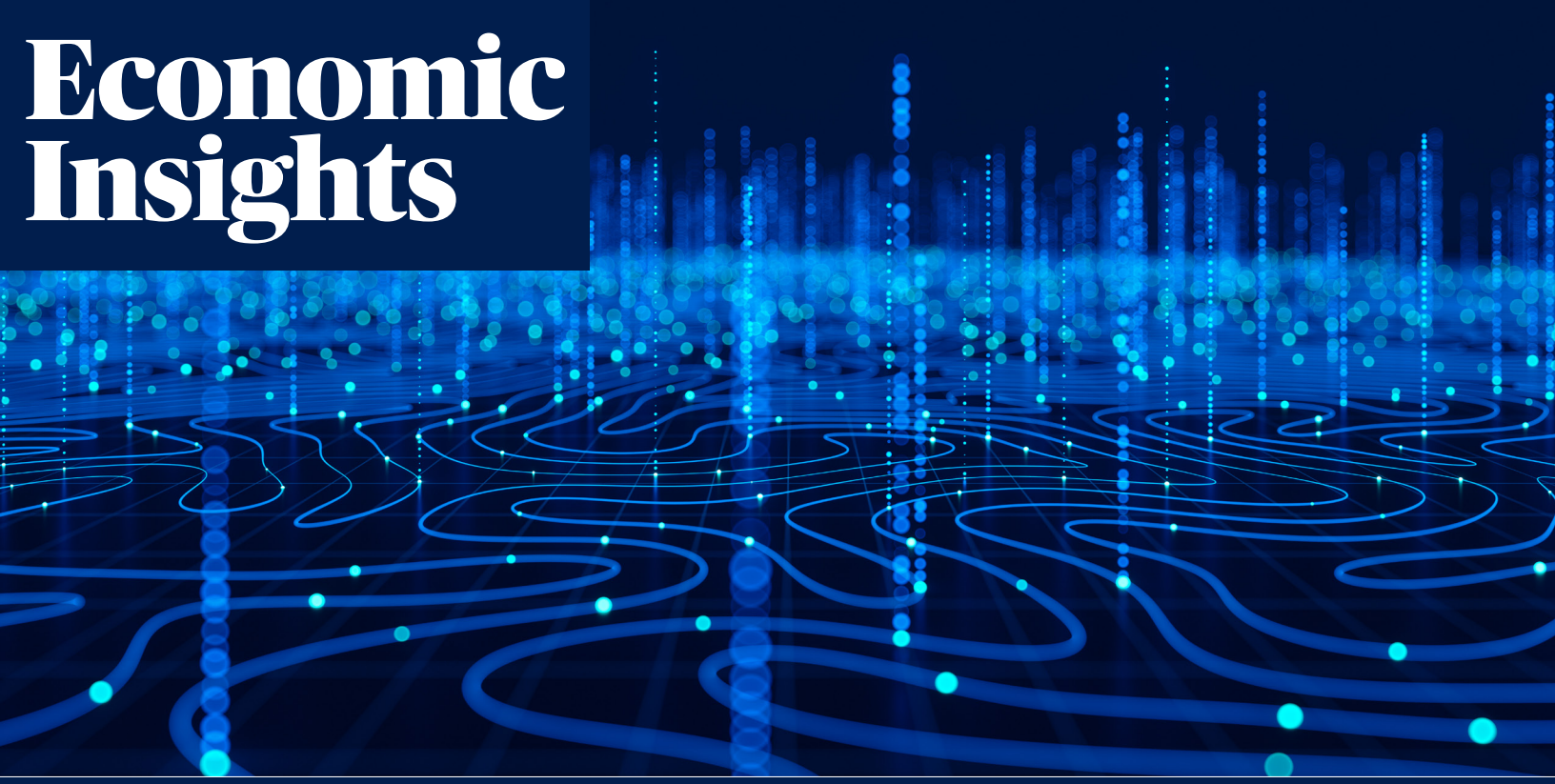
Economic Insights



APR 2026

How Quantum Computing Threatens Cryptography

Quantum computers may one day decrypt our data, but banks can mitigate this risk.

Fabio Sanches
Formerly Director of Quantum Computing
Federal Reserve Bank of Philadelphia

We are fast approaching the day when quantum computers with meaningful capabilities are operational. This has significant implications for cryptography. Quantum computers run algorithms that solve certain problems more efficiently than in classical systems. Notably, one quantum algorithm, Shor's algorithm, provides an exponential advantage in performing prime factorization. Modern public-key cryptography depends on the difficulty of factoring primes. A quantum computer that can easily factor primes can also easily break cryptographic codes.[1]

However, the National Institute of Standards and Technology (NIST) is currently standardizing alternative public-key cryptography systems that are not vulnerable to quantum

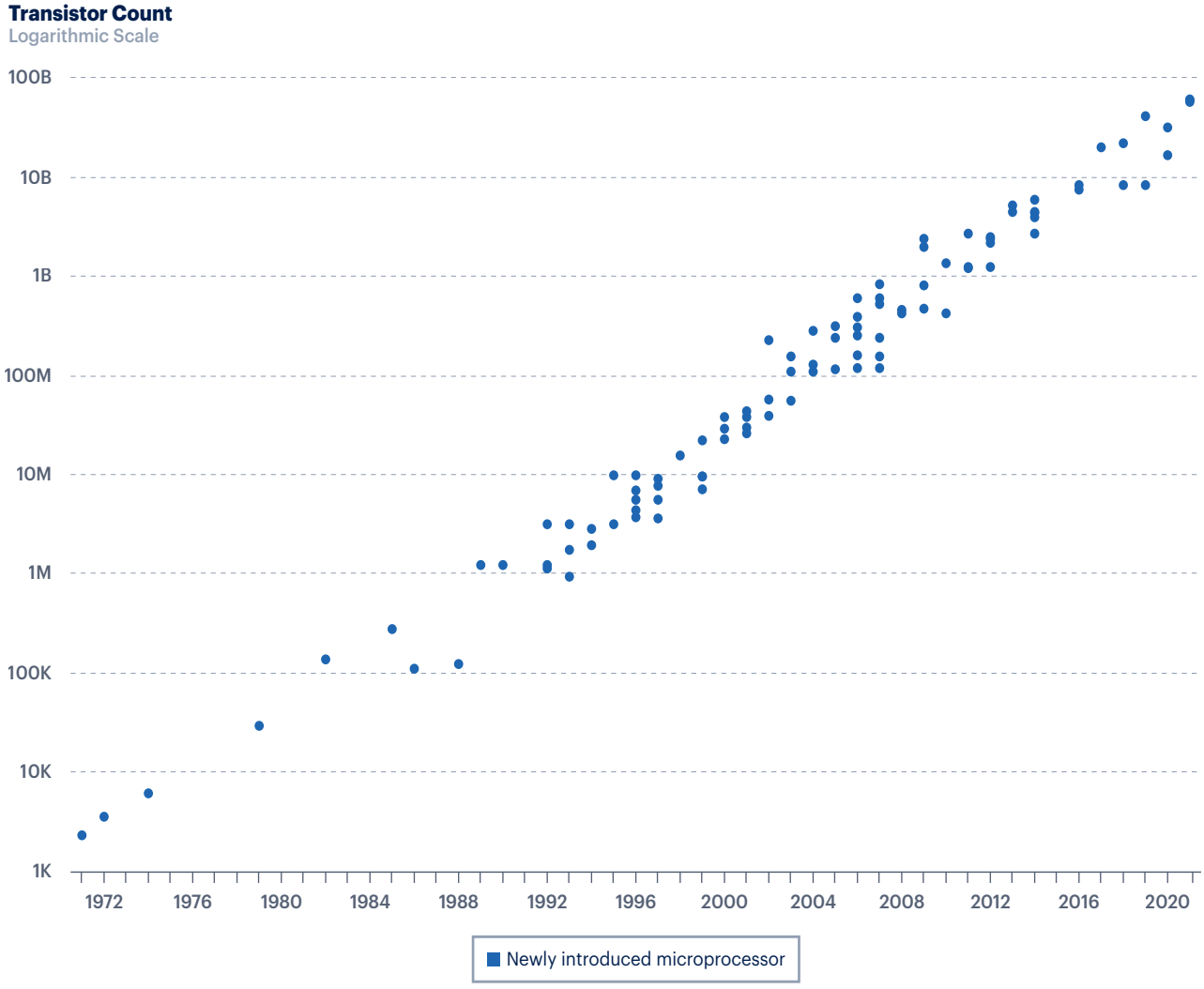
attacks. In this article, I describe this public-key infrastructure (PKI) upgrade and discuss its implications for data security.

Introduction

In 1965, engineer and entrepreneur Gordon Moore predicted that the number of components in integrated circuits would double every year.[2] He claimed that the trend would likely continue until 1975. As we now know, the predicted exponential growth has continued for more than five decades (Figure 1). This physical feat powered the overall increase in computational power, which is why we see a similar exponential trend in floating-point operations per second (FLOPS) for supercomputers (Figure 2).

FIGURE 1

Moore's Law: The Maximum Number of Transistors on a Microchip Doubles Every Two Years

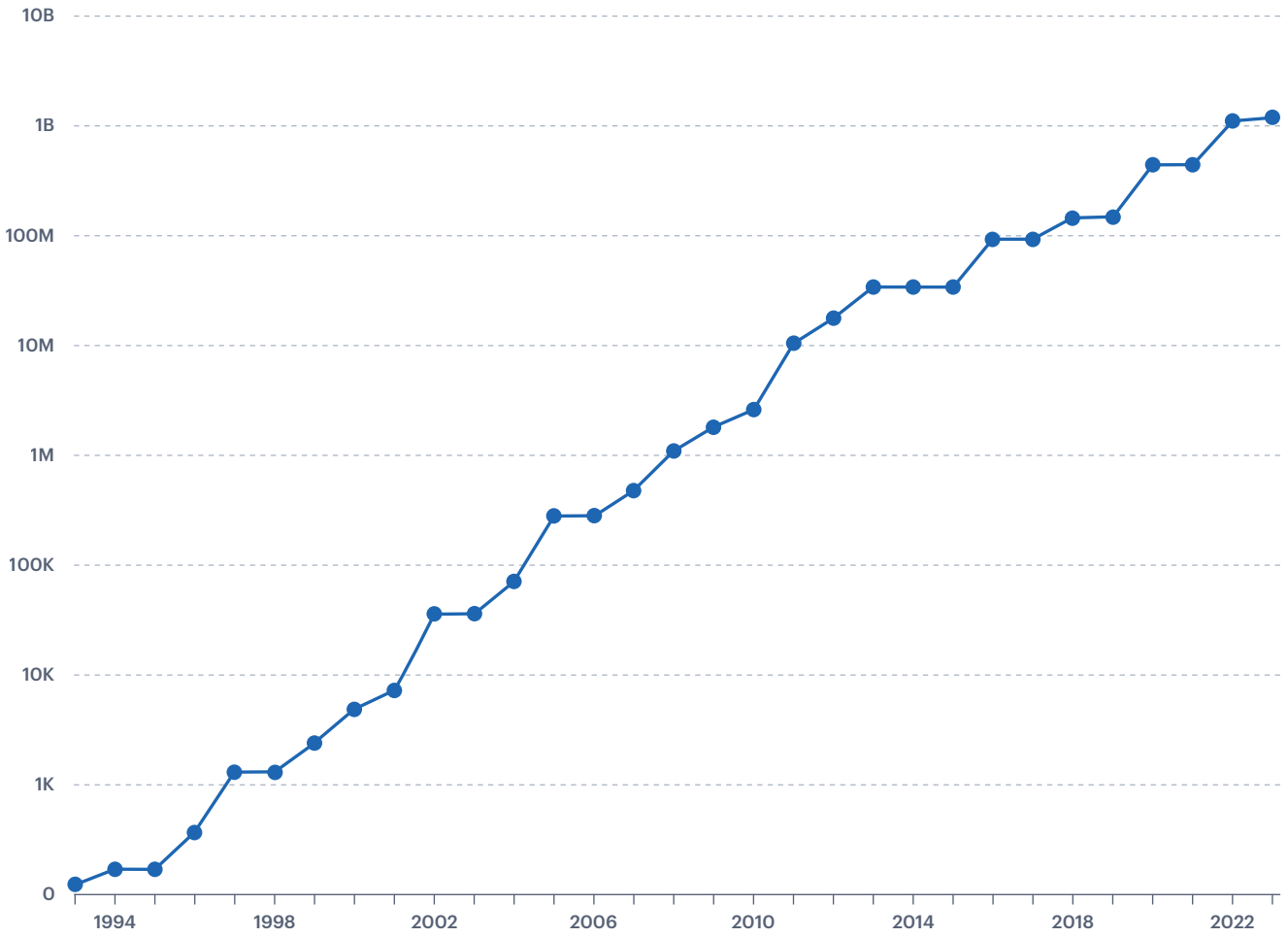


Data Source: Karl Rupp; raw data available under a Creative Commons Attribution 4.0 International Public License and accessible at the Microprocessor Trend Data repository on GitHub, https://github.com/karlrupp/microprocessor-trend-data

FIGURE 2

Computational Capacity of the Fastest Supercomputers

Floating-Point Operations per Second
Logarithmic Scale



Data Source: "Data Page: Computational capacity of the fastest supercomputers," Our World in Data (2025); data adapted from Dongarra et al.; retrieved from https://archive.ourworldindata.org/20250909-093708/grapher/supercomputer-power-flops.html [online resource] (archived on September 9, 2025)

Note: A floating-point operation (FLOP) is a type of computer operation. One FLOP represents a single operation involving floating-point numbers, such as addition, subtraction, multiplication, and division.

Miniaturization has enabled individual chips to process more information faster, increasing the scope of what computers can do. This increased computational capacity has enabled computers to deliver improved performance, leading to an explosion of software and applications across a variety of tasks; the current breadth and uses of software would have been hard to imagine with early computers, given their limited capacity.[3]

But the exponential growth in miniaturization may be coming to an end.[4] If computational capacity is to continue its exponential growth, computer scientists must rely on other tools, including quantum computing.[5] Unlike previous innovations, quantum computing doesn't rely on smaller transistors. Instead, it relies on components in which quantum mechanical effects are controlled and utilized for

computation.[6] These components perform computations in which information is treated quantum mechanically. From this perspective, quantum computing is not the next step in the exponential growth of computational capacity but instead a new computational platform. And we've only just begun this journey: Quantum computing is currently where integrated circuits were at the time Gordon Moore made his famous prediction.

Overview of Quantum Computing

There is more than one way to encode quantum information, but when people talk about quantum computation, they are typically thinking of qubits, a two-state quantum mechanical system that is by far the most common building block used to model a quantum computer. Qubits are analogous to the bits used in a classical computer system. A classical system uses logic gates to manipulate bits (for example, to flip a bit between "0" and "1") in order to process information, whereas a quantum mechanical system manipulates qubits using quantum gates. Quantum gates provide a richer set of operations (meaning that there are more ways to manipulate qubits than to manipulate classical bits), and quantum algorithms leverage this additional complexity to accomplish certain tasks more efficiently—that is, by using fewer operations (gates) on qubits.[7] The more qubits a quantum computer has, the more meaningful algorithms it can execute, so long as the quantum computer is stable enough (that is, so long as it makes a sufficiently small number of errors) to apply the necessary gates to execute the algorithm.

Most approaches to building quantum computers focus on engineering a system (comprising, for example, ions or special components that can be put on silicon chips) that can be used as qubits and then manipulating those qubits effectively. These systems naturally have two states (or are engineered to behave as if they do). In addition to engineering qubits (in a scalable way), a quantum computer must also have good control over these qubits. This entails being able to initialize qubits to a specific state, apply a minimum set of gates to run any algorithm, stabilize qubits to prevent errors during execution (wherein the realized qubit operation was not what was intended), and reliably read qubit states.[8]

Currently, companies building quantum computers aim to both scale up the number of qubits and improve the control they have over those qubits, resulting in a lower error rate when performing gates. Companies have demonstrated systems with hundreds of qubits, some with error rates sufficiently low to perform specific experiments that are difficult to simulate on classical devices. In these experiments, these modest quantum computers have generated outputs for these computational tasks faster than some of the most powerful classical supercomputers.

Although these demonstrations are an important stepping-stone, quantum computers have yet to have real-world business implications because their error rates are still too high. To overcome that hurdle, companies are developing quantum error correction (QEC), which provides a way to detect and correct errors while the quantum computer is executing its program. It accomplishes this by distributing the quantum information across a larger number of physical qubits, thereby creating a single logical qu-

bit that behaves like a single error-resistant unit with a significantly lower error rate. When you combine many logical qubits along with the ability to manipulate them, you create an error-corrected quantum computer. Recently, we've seen important milestones in QEC, including improved performance when increasing the number of physical qubits used in the encoding.[9]

In short, we are entering an era in which classical computers can no longer fully simulate quantum computers, and we have early demonstrations of how to use QEC to lower error rates.[10] The next step is for companies to develop fault-tolerant quantum computers with sufficient qubits to run quantum algorithms at an impactful scale. Simultaneously, progress in quantum algorithms must be made to enable more effective use of this quantum hardware.

Public-Key Cryptography and Shor's Algorithm

Public-key cryptography enables secure digital communication across networks by verifying the authenticity and integrity of messages. It relies on a pair of keys: one public (openly distributed) and one private (kept secret). The security of this asymmetric encryption relies on the computational difficulty (or complexity) of certain mathematical problems. The two most used techniques for producing these two keys are Rivest–Shamir–Adleman (RSA) and elliptic curve cryptography (ECC).

In RSA, key generation begins by selecting two large prime numbers and multiplying them to produce a number called the modulus. This modulus is shared in both the public and private keys. The specific values of the two primes allow the system to calculate a private exponent that is mathematically paired with a chosen public exponent.

When a sender wishes to encrypt a message, they first convert the text into an integer. They then scramble this number using the recipient's public exponent and modulus. The resulting ciphertext can be transmitted safely over public channels because reversing this process without the private key would require identifying the original two prime numbers from the modulus alone. This is a mathematical feat no known (classical) algorithm can perform efficiently. The recipient, however, holds the private key, which allows them to easily decrypt and read the message.

The security of this encryption process relies on the computational difficulty of reversing the exponentiation performed in the encryption process. (The challenge is factoring the product of two large primes and computing discrete logarithms.)

But as Peter Shor showed in 1994, a quantum computer with sufficient capacity can reverse the process and obtain the private key from the modulus and the public key—and thus decrypt the message. He did so by demonstrating that integer factorization can be reduced to finding the period of a specific function (that is, the distance of a repeating pattern).

This related problem is intractable for classical computers, but Shor's algorithm solves it efficiently by using the quantum Fourier transform (QFT). After encoding the problem into a quantum state, the QFT

enhances the amplitude of the period of the specific function it's applied to. This amplitude is directly related to the probability of reading out the associated outcome. Therefore, the algorithm outputs the information needed to perform integer factorization.

Beyond decrypting messages, this capability undermines authentication more broadly. Digital signatures rely on similar concepts, meaning that Shor's algorithm can also be used to forge signatures without private keys and thus impersonate their owners.

Threat Timeline, Postquantum Cryptography, and Cryptographic Agility

How long will it take until we see quantum computing hardware able to run Shor's algorithm at a scale that threatens widely used keys? According to EvolutionQ's 2024 survey, technical experts believe that such a computer may be operational in five to 10 years and will likely be operational in 15 years (Figure 3).[11] An operational quantum computer that can break cryptographic codes presents a serious risk for the financial system. Even if we're unlikely to see such a computer before 2030, the risk is too high to ignore, especially given how severe the consequences are.

The time sensitivity of the data should also be considered when planning for this risk. Michele Mosca, a professor at the University of Waterloo and CEO of EvolutionQ, often discusses this risk in terms of a simple inequality when thinking about risk mitigation in this area: Let x be the number of years some data need to be safe, y the number of years it will take to upgrade the cryptographic infrastructure, and z your estimated timeline for the successful development of a cryptographic quantum computer, perhaps based on your risk threshold: If $x + y > z$, you're already beyond your risk tolerance threshold.

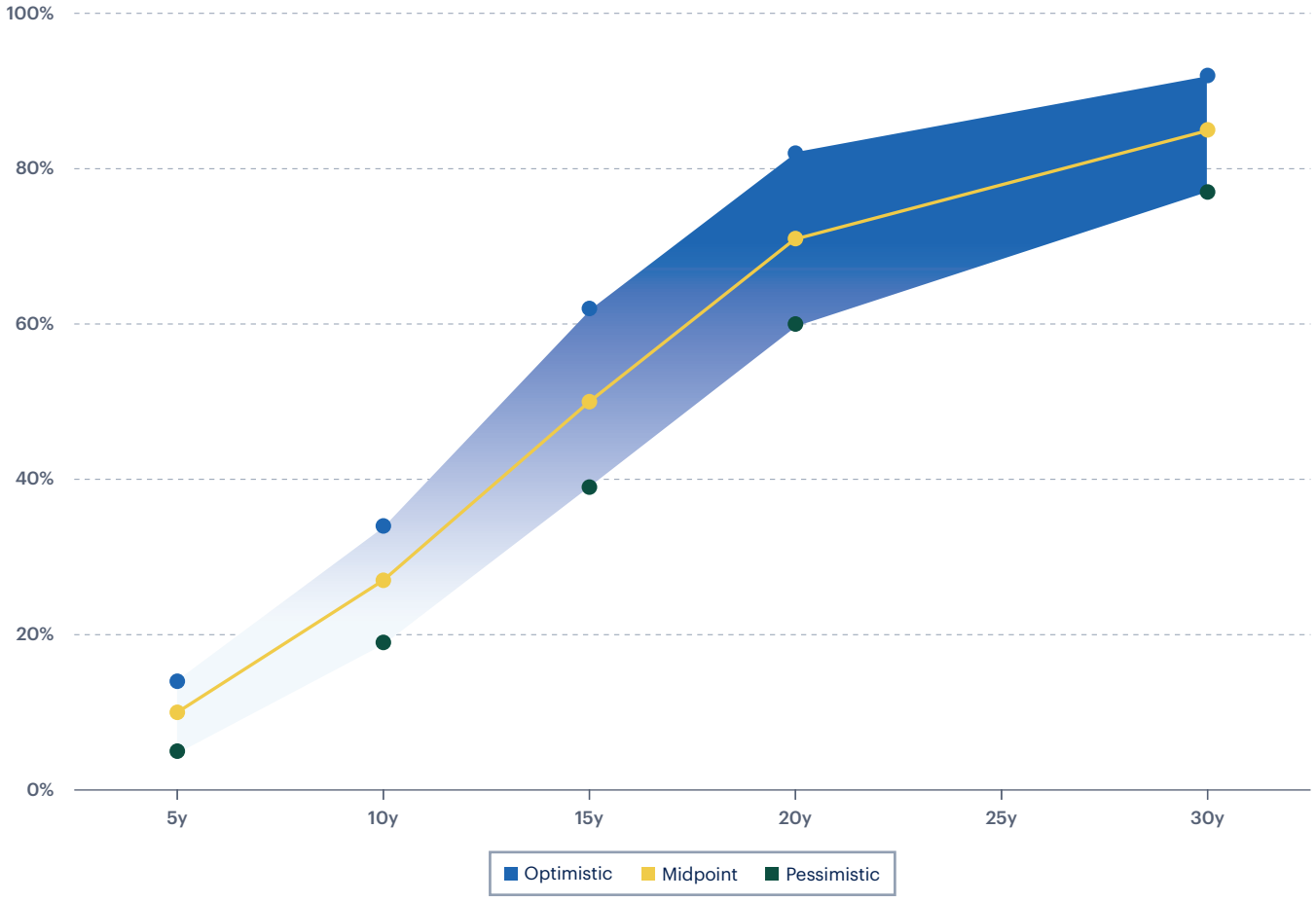
However, quantum computers are not exponentially faster than classical computers at every task. Some problems are still difficult for quantum computers to solve, which opens the door for postquantum cryptography (PQC), or encryption algorithms that quantum computers cannot efficiently break.[12]

NIST recently initiated a process to standardize PQC algorithms. In August 2024, NIST finalized its selection and released three PQC standards.[13] Although these algorithms have been studied by cryptography experts and appear dependable, they have not been deployed as broadly as RSA and ECC. As a stopgap measure, traditional cryptography is being deployed alongside PQC, creating hybrid encryption schemes. This is why the first step in an organization's effort to upgrade its encryption system is to take an inventory of its existing cryptographic assets. Because upgrades to the core algorithmic infrastructure are rare, most organizations will need to identify which applications are vulnerable to the threat and prioritize their upgrade based on data security risks and application criticality.

Beyond deploying both current and quantum-resilient cryptography, cryptographic agility aims to design security systems that can easily leverage and update cryptographic algorithms. The result is a system that is more responsive to emerging threats and enables organizations to efficiently change infra-

FIGURE 3

Survey Results: When Will a Quantum Computer Be Capable of Breaking RSA 2048 in Less than 24 Hours?



Data Source: Mosca and Piani (2024)

Note: Estimates of the cumulative probability of a cryptographically relevant quantum computer in a time range between the average of an optimistic (top value) or pessimistic (bottom value) interpretation of the estimates indicated by the respondents, and midpoint. The security of RSA 2048 is based on the difficulty of factoring a 2,048-bit number.

structure to use newer and more secure algorithms when necessary. To achieve this, a system must be able to support and maintain compatibility across multiple algorithms, seamlessly transition to newer algorithms designed to fix vulnerabilities as requirements change, and upgrade cryptographic algorithms without requiring a redesign of the underlying system.

Quantum Key Distribution

Unlike postquantum cryptography, in which classical computation is still used for public-key cryptography but is secure from quantum attacks, quantum key distribution (QKD) directly leverages quantum mechanics to establish a shared key between two parties. This key can then be used in symmetric encryption schemes, in which the same key is used for the encryption and decryption of data.

Typically, QKD involves preparing qubits that are sent through a quantum channel and then measured/observed in certain bases.[14] (Bases are sets of reference states used to interpret a qubit's value, such as horizontal/vertical polarization in the case of photons.) This method enables secure communications because it is impossible to create identical copies of unknown quantum states, and because any third party that tries to observe a qubit in the quantum channel would inevitably disturb its state and therefore not be able to extract reliable information from it.

In a simplified example of the most cited protocol, BB84, Alice sends Bob a private key. Alice prepares qubits in one of four states (two for each chosen basis, which creates a coordinate system that can be used to describe the outcome). If you don't know the chosen basis, you cannot distinguish between the states with certainty. Alice then sends these qubits to Bob through a quantum channel. Bob measures these qubits in randomly chosen bases. At this point, Bob does not know the intended message because he does not have access to the chosen bases for each qubit Alice sent. This is also true for any third party trying to obtain the message. In addition, any measurement performed by a third party would disturb the chosen bases for that qubit. But through a public classical channel, Alice and Bob can compare the bases they have prepared and measure the qubits. When their measurements do not agree, they discard the qubit. Finally, they publicly announce a random portion of the bits resulting from the measurements to check for agreement. They use the remaining qubits to establish a secret connection.

QKD is *not* a replacement for PQC. It is typically envisioned in combination with other classical cryptography schemes as an additional security layer for certain applications. QKD systems are also still limited in the transmission distances they can be used over and typically require special hardware to establish a quantum channel. The specific implementation may also be susceptible to attacks due to hardware flaws. Nonetheless, some firms are currently deploying QKD in limited settings.[15]

Outlook

Companies that produce quantum computing hardware continue to work toward fault-tolerant and scalable quantum computers. This technology is not currently mature enough to affect applications that are important to organizations or break public-key cryptography, but the timeline for achieving systems capable of such tasks appears to be roughly on track with company roadmaps. This means that there are real cybersecurity risks to prepare for, and it is not too early for organizations to prioritize this effort.

To prepare for their PQC transition, organizations can take the following steps.[16] First, assign a team to scope the migration effort and create a roadmap. Second, inventory cryptographic assets by identifying vulnerabilities across the organization and considering data security requirements to help with prioritization. Third, ask vendors about their roadmaps and migration efforts. Fourth, develop a migration strategy with priorities informed by the previous steps. And fifth, educate and train employees.

Notes

1 Shor's algorithm converts the prime factorization problem into a discrete logarithm problem, threatening any encryption algorithm that can be converted into a discrete log problem.

2 See Moore (1998).

3 However, the miniaturization in components has implications for productivity and, thus, productivity growth. See Azar (2022).

4 See Woods (n.d.).

5 Computer scientists are also testing new architectural designs, parallelization, and other algorithmic and engineering innovations. See Leiserson et al. (2020).

6 Quantum mechanics is a fundamental framework describing how nature behaves. The theory emerged in the early 20th century. Unlike classical physics, it successfully explains phenomena at the atomic and subatomic scales. Central to quantum mechanics is the concept of the wavefunction—a mathematical object that encodes the probabilities of possible outcomes when measuring a system.

7 Most of the time, we care about the scaling of the number of operations required to solve a problem as the problem size increases. We say that a quantum algorithm is more efficient when it provides better scaling in solving a given problem.

8 These conditions are often referred to as DiVincenzo's criteria, named for physicist David DiVincenzo.

9 See, for example, Svore (2024) and Google Quantum AI et al. (2025).

10 We can fully simulate the operation of a sufficiently small quantum computer on a classical computer by modeling its quantum mechanical behavior, but doing so often costs significant computational resources and is not feasible above a certain scale.

11 Mosca and Piani (2024).

12 Two notable approaches are lattice-based cryptography, which uses high-dimensional lattices and can be used for encryption and digital signatures, and hash-based cryptography, which takes an input and returns a fixed-size string.

13 These three PQC standards are the module-lattice-based key-encapsulation mechanism (ML-KEM), the module-lattice-based digital signature algorithm (ML-DSA), and the stateless hash-based digital signature algorithm (SLH-DSA).

14 A quantum channel is a communication link that allows for the transmission of quantum information between parties—for example, the sending of qubits from one location to another.

15 See JPMorgan Chase (2024).

16 These steps courtesy of the National Institute of Standards and Technology (2023).

References

Azar, Pablo D. "Moore's Law and Economic Growth," Federal Reserve Bank of New York Staff Reports 970 (2022), https://www.newyorkfed.org/research/staff_reports/sr970.

Google Quantum AI and Collaborators. "Quantum Error Correction Below the Surface Code Threshold," *Nature*, 638 (2025), pp. 920–926, https://doi.org/10.1038/s41586-024-08449-y.

JPMorgan Chase. "JPMorgan Chase Establishes Quantum-Secured Crypto-Agile Network" (2024), https://www.jpmorgan.com/technology/news/firm-establishes-quantum-secured-crypto-agile-network.

Leiserson, Charles E., Neil C. Thompson, Joel S. Emer, et al. "There's Plenty of Room at the Top: What Will Drive Computer Performance After Moore's Law?" *Science*, 368:6495 (2020), https://doi.org/10.1126/science.aam9744.

Moore, Gordon E. "Cramming More Components onto Integrated Circuits," *Proceedings of the IEEE*, 86:1 (1998), pp. 82–85, https://doi.org/10.1109/JPROC.1998.658762.

Mosca, Michele, and Marco Piani. *Quantum Threat Timeline Report 2024*. Global Risk Institute (2024), https://globalriskinstitute.org/publication/2024-quantum-threat-timeline-report/.

National Institute of Standards and Technology, U.S. Department of Commerce. "Quantum-Readiness: Migration to Post-Quantum Cryptography" (2023), https://www.cisa.gov/sites/default/files/2023-08/Quantum%20Readiness_Final_CLEAR_508c%20%283%29.pdf.

Svore, Krysta. "Microsoft and Quantinuum Create 12 Logical Qubits and Demonstrate a Hybrid, End-to-End Chemistry Simulation," Microsoft (2024), https://azure.microsoft.com/en-us/blog/quantum/2024/09/10/microsoft-and-quantinuum-create-12-logical-qubits-and-demonstrate-a-hybrid-end-to-end-chemistry-simulation/.

Woods, Audrey. "The Death of Moore's Law: What It Means and What Might Fill the Gap Going Forward," MIT CSAIL Alliances (n.d.), https://cap.csail.mit.edu/death-moores-law-what-it-means-and-what-might-fill-gap-going-forward.