Quantum Computing and its Effect on Cryptography

Owen McGlynn
omcglynn@umass.edu
University of Massachusetts Amherst
Amherst, MA, USA

1 INTRODUCTION

To most people, quantum computing sounds like a buzzword; something brought up by your friend as some notion of the future or by that really annoying person you know in an attempt to sound smart. In the past, this line of reasoning was not entirely untrue, as it was a largely unexplored, inaccessible field of theoretical study. However, recent research developments dictate that quantum computing will eventually become a much more important part of our daily lives, for better or worse.

Cryptography is another far off concept for most. Most people are well aware of the importance it holds to our daily lives, but due to a lack of understanding, tendto not be able to fully appreciate why or how. For example, if a company were to say, "Secured with Military-Grade AES-256 Encryption and AI-Proof Cyber Fortifications—Your Data, Impenetrable by Design," most people would think it's some revolutionary new system. In reality, AES is the global standard, and their "AI-Proof Cyber Fortifications" is a way to avoid mentioning Dave from their cybersecurity team, who spends half his time resetting passwords and the other half running automated scans.

The truth is that our current methods of encryption work great against the machines of our time. But quantum computing really is something of the future, and there is no doubt that it will throw everything we currently hold secure into the air for everyone to see. This is something that needs to be seriously thought about before it is developed too far.

In this paper, we will cover a brief overview of quantum computing and cryptography, their intersection, why it is important, and future remediation possibilities to safeguard our world from the threat quantum computing presents to our information security.

2 WHAT REALLY IS QUANTUM COMPUTING?

To be able to understand the massive upset Quantum Computing will cause, one must know a little history in the computing field, and be able to put an actual definition to the term.

In the computers of today, billions of tiny electrical circuits manipulate and process information using "bits", represented logically as 1's and 0's or on and off—often referred to as classical computation. The computers then take various problems and break them down into sets of logical statements that can be processed using binary and what is called bit manipulation. When you get into more complex problems with an increasing number of variables and dependencies, classical computers must handle an exponentially growing number of possible states [10]. Each additional variable or constraint increases the computational overhead, requiring more time and processing power. This is especially problematic for tasks like cryptography, optimization, and large-scale simulations, where classical computers must evaluate numerous potential solutions, either sequentially or in parallel using vast computational resources. As a result, certain problems—such as factoring large numbers, simulating molecular interactions, or optimizing complex systems—become infeasible for even the the pinnacle of modern classical computing.

In quantum computers, rather than using bits, they use a new fundamental building block, the qubit. Qubits fundamentally have the same idea as bits but, while bits must be in either a state of 0 or 1, or on or off, qubits can be in a superposition, means it can represent both 0, 1, and anything in between. While the positions 0 and 1 might be straightforward, qubits work by manipulating the probabilities of each, such that they can be represented as a complex number, based on the probability amplitudes of either state. You can think about this from the context of the *Many Worlds* theory where when you have a simple decision, the universe branches into parallel versions of itself and its superposition is all of those at the same time. Since quantum computers can leverage superposition and quantum interference, they process multiple possible states simultaneously [18]. However, rather than brute forcing every outcome, quantum algorithms manipulate probability amplitudes to amplify correct solutions and suppress incorrect ones. Upon measurement, the qubit will "collapse" into a definite state based on the probability amplitudes it once held. To produce an output, quantum computing also uses a concept called entanglement, a phenomenon where multiple qubits become correlated. When qubits become entangled, it becomes impossible to describe the state of one independentlym and their measurement outcomes are correlated, so when one is measured, the outcome of the other will yield if measured in the same basis is determined [3].

Now that we have gotten through the super dense, barely understandable definition of quantum mechanics, lets take a break with some history. Funnily enough, though many people consider quantum computing as something extremely new and modern, it has actually been around since the 1980's, and the quantum theory its built on, since the 1920's (though we can just ignore that since its not the focus of this paper). Fundamentally, the history of quantum computing is considered to have officially started in 1982. It began when Richard Feynman started his lectures considering potential advantages of computing using quantum models rather than classical ones. In 1985, the first quantum algorithms applied to computing emerged, namely, the Deutsch's Algorithm. This algorithm takes a mystery function which takes in a single bit and outputs a single bit and determines whether the function always outputs the same value (constant) or outputs 0 for one input and 1 for the other (balanced). Classical computation methods would do this by checking both inputs, but, Deutsch's algorithm does it by putting a qubit in superposition, and when the function is called on it, it affects the quantum state such that the final measurement immediately tells whether or not the function is constant or balanced [19]. From a programming perspective, you can think about this as going from an O(n) lookup to a O(1), which can be infinitely more efficient (depending on the size of n).

Skipping ahead to 1994, Peter Shor built on these findings to create Shor's Algorithm, an algorithm more relevant to this paper as it allowed for the finding of prime factors of numbers, effectively breaking the widely used RSA cryptographic method – that is, if a large enough quantum computer were to be built. This introduces the first feasible example of quantum computations threat to our modern day security. Though there still does not exist a quantum computer powerful enough to actually break our modern cryptographic processes, we get closer by the day. This is exemplified by Microsoft's most recent breakthrough; Majorana 1. Majorana 1 is a processor that offers a path to fitting a million qubits on a chip the size of your palm. Though Majorana 1 is still not completed, and it is still ninteen million qubits short of satisfying estimates of Shor's Algorithm's requirements to its fullest extent[19], it is a huge step in the right direction [4].

3 CRYPTOGRAPHY; A SHORT SYNOPSIS

Cryptography truly is one of the most widely used forms of applied mathematics. In fact, every single person that has ever touched the internet has some kind of experience with it, whether they know it or not. It is used everywhere, from the more obvious examples such as password storage and in cryptocurrency, to those that are less so. This includes, encrypting web traffic so an adversary can't just grab plain-text passwords and personal data as it travels through the internet. It truly is a ubiquitous asset in today's digital age.

At its core, cryptography relies on mathematical functions that make encoding and decoding information secure against unauthorized access. These functions often involve hard-to-reverse mathematical problems such as prime factorization, modular arithmetic, and elliptic curves. Encryption follows a structured process: a plaintext message is transformed into ciphertext using an encryption function and a key, and only an authorized recipient with the correct decryption function can revert it. The security of cryptographic systems depends on computational hardness, randomness, and structured key exchange protocols—ensuring that sensitive data remains confidential even in the presence of adversaries.

The observable history of cryptography stretches all the way back to around 1900 B.C., wherein a scribe in Ancient Egypt scribbled some unusual hieroglyphics in the main chamber of the tomb of the nobleman Khnumhotep II. In fact, many ancient societies seemed to use some form of encryption, whether it be a scribe concealing a formula for pottery glaze in Mesopotamia, or in early India, where assignments were said to be given to spies in "secret writing." The most widely known cipher—a secret or disguised way of writing—is probably the Caesar Cipher, used by its namesake, Julius Caesar, to convey secret messages to his generals on the war front.

The Caesar Cipher is quite simple. It is what's called a substitution cipher, in which every letter of the alphabet is replaced with another, one-to-one, and in the case of the Caesar Cipher, this was done using a numeric shift. This can be visualized quite easily by taking two strips of paper containing the alphabet, placing one above the other such that it lines up with itself, and then moving one to the side some number of positions such that each one now lines up with a separate letter [20].

Nowadays, we have developed methods significantly more complex than the ones employed back then. The most widely used and well-known are as follows: TLS/SSL, protocols to encrypt web traffic; RSA, a public key cryptographic system frequently used for digital signatures and secure communication; and AES, as mentioned in the introduction as the global gold standard for data, a symmetric encryption algorithm that protects data during transmission and storage.

For now, let's go into the workings of RSA as we have already introduced a quantum algorithm that could pose a threat to it. RSA is what we call an asymmetric encryption method, meaning, rather than having one key that both encrypts and decrypts messages, every individual using it has two: a public key, which can be freely shared and is used by others to encrypt messages, and a private key, which should remain confidential and can decrypt messages encrypted using the public key. [5]

4 QUANTUM COMPUTING'S THREAT TO ENCRYPTION

Now that some insight has been given into the background and inner workings of the two main topics being discussed, lets dive into why one relates to the other.

Quantum computing enables easier encryption cracking because it dramatically accelerates certain mathematical computations that classical computers struggle with.

In the case of RSA, security relies on the difficulty of factoring large numbers. Given an RSA public key, $N = p \times q$ where p and q are large prime numbers, breaking RSA means finding p and q Classical computers take an impractically long time to do this because the best known algorithms run in sub-exponential time [2],

$$O(e^{(1.9+o(1))(log(N)^{1/3}(log(log(N)))^{2/3}}) \tag{1}$$

However, Shor's algorithm reformulates the problem into a periodicity-finding problem and solves it using the Quantum Fourier Transform (QFT) in polynomial time,

$$O((log(N))^3) \tag{2}$$

This means that a sufficiently powerful quantum computer could factor a 2048-bit RSA key in hours rather than the billions of years it would take classically. More generally, quantum computing threatens encryption because many cryptographic systems rely on the difficulty of problems like factoring and discrete logarithms, both of which can be solved efficiently using quantum algorithms.

Today's quantum computers are nowhere near powerful enough to execute dangerous algorithms like Shor's at a scale that threatens modern cryptography. They typically contain only dozens to a few hundred qubits, and these qubits are "noisy" (prone to errors and decoherence), making sustained, large computations impractical. In fact, breaking a standard RSA-2048 key via Shor's algorithm is estimated to require on the order of millions of stable, error-corrected qubits – one analysis suggests over 4,000 fault-tolerant (logical) qubits (each of which containing 1000+ noisy "physical" qubits) would be needed to crack a 2048-bit RSA key. By contrast, today's largest quantum prototypes have on the order of 10^3–10^4 physical qubits, none of which are fully error-corrected. [9]

The huge gap between current capabilities and the requirements for cryptographic attacks means that, for now, our encryption remains safe from quantum codebreaking. Projections by both researchers and government agencies converge on a similar timeline: a cryptographically relevant quantum computer (one capable of breaking encryption) is expected by the 2030s under optimistic scenarios. In other words, the consensus is that we have perhaps years at best before quantum computers can routinely crack current encryption methods. Some forecasts are even more aggressive – for example, one industry study suggests quantum attacks on certain methods could emerge in the next 5–10 years – while others remain cautious, noting persistent engineering challenges that could delay a breakthrough until the late 2030s. That said, the field of quantum computing is advancing quickly, and experts warn that the situation is ever changing and no estimate is guaranteed to be.

Fortunately, for the time being symmetric-key encryption algorithms remain secure. This is because they do not use deep algebraic problems to hide their security – which is what Shor's algorithm exploits and solves in polynomial time. Instead, their "hardness" is simply that for an n-bit key string, an attacker must locate the one correct key among 2^n possibilities. Thus, the only known quantum attack against it — Grover's algorithm — yields just a quadratic speed-up over classical brute-force search and thus simply increasing key length re-secures the algorithm [7]. Regardless, asymmetric cryptography schemes are still very widely used and will inevitably require securing.

The term "Q-day" has even been coined to denote the day when quantum computers finally overpower our classical methods of cryptography; recent analyses place Q-day sometime around 2025–2040, though the exact timing is, of course, uncertain [8].

5 QUANTUM REMEDIATION

Despite the fact that Post-Quantum-Cryptography (PQC) is an extremely new, and constantly evolving field, the push for quick development and adoption of quantum-secure algorithms is very pressing. Standards and security agencies such as the NIST, ETSI (European Telecommunications Standards Institute), and the Institute for Quantum Computing encourage the adoption of post-quantum encryption algorithms as quickly as possible not only due to the uncertainty of the timeline but also because of the threat of "harvest now, decrypt later" programs which are stashing information encrypted with current day algorithms for use when the post-quantum world arrives[16].

In 2016, NIST initiated the post-quantum cryptography standardization project. This project elicited the opportunity for researchers to submit post-quantum compliant algorithms for the chance to have them evaluated and potentially adopted as the standard going forward. After 8 grueling years consisting of numerous rounds of evaluation, disqualification, and testing, in 2024, 3 of the 68 submissions were finalized and officially released as NIST inaugural post-quantum cryptography standards [16].

Two of the three released standards were derived from submissions from the Cryptographic Suite for Algebraic Lattices, or CRYSTALS for short. Both of these algorithms are based on hard problems over module lattices which the Simons Institute at UC Berkeley defines as lattices "...with an extra algebraic structure that can be used to improve the efficiency of cryptographic constructions"[1]. In the case of these algorithms, those algebraic structures relate to a cryptographic technique called "learning with errors" wherein noise is introduced to the lattices to hide secrets such as private keys[15].

5.1 FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) Standard

FIPS 203 specifies a key encapsulation mechanism (KEM) standard based on the CRYSTALS-Kyber[14]. A KEM is a set of algorithms that allows two parities to establish a shared (symmetric) secret key over a public channel using asymmetric methods. See below graphic for explanation of KEM vs Key Encapsulation (i.e. Diffie-Hellman):

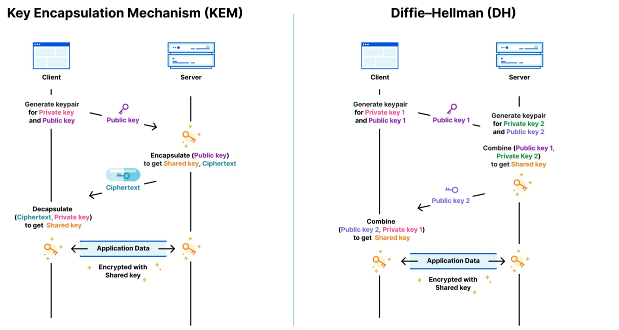
Key Encapsulation Mechanism (KEM)
Client
Server
Generate keypair for Private key and Public key
Public key
Encapsulate (Public key) to get Shared key, Ciphertext
Ciphertext
Decapsulate (Ciphertext, Private key) to get Shared key
Application Data
Encrypted with Shared key
Diffie–Hellman (DH)
Client
Server
Generate keypair for Private key 1 and Public key 1
Public key 1
Generate keypair for Private key 2 and Public key 2
Combine (Public key 1, Private Key 2) to get Shared key
Public key 2
Combine (Public key 2, Private key 1) to get Shared key
Application Data
Encrypted with Shared key

Figure 1: KEM vs. Diffie-Hellman[17]

Key-Encapsulation Mechanisms (KEMs) have been around since the early 2000s, usually hiding a seed in a single group element via modular exponentiation or elliptic-curve multiplication. Kyber's innovation is to replace that one-dimensional algebraic structure with module lattices—high-dimensional grids of polynomial coefficients so that each public key and ciphertext is itself a noisy lattice point. Breaking Kyber means recovering the private key, which requires solving the Module-LWE problem (a variant of the learning with errors problem to relate to module lattices.) which is believed hard even for quantum computers. At the same time, Kyber keeps keys and ciphertexts compact by deriving its public matrix from a small seed and compressing coefficients[12].

5.2 FIPS 204, Module-Lattice-Based Digital Signature Standard

FIPS 204 specifies a post-quantum digital signature standard based on the CRYSTALS-Dilithium algorithm[14]. Dilithium's innovation is to streamline a classical lattice signature technique called "Fiat–Shamir with aborts." In simple terms, the signer uses some random data to form a candidate signature, and if that candidate could leak too much information about the secret key, the algorithm throws it out and tries again (this is the "abort" step). Notably, Dilithium avoids the heavy discrete Gaussian sampling that earlier schemes used and opts for two simpler types of randomness instead [13]:

(1) For each signature, it draws a one-time random value y uniformly from a limited range.
(2) It generates the long-term secret key and error values using centered-binomial sampling, a simple method of adding up random bits to get a bell-shaped distribution without full Gaussian math.
(3) The signer checks that all parts are within allowed limits. If any part are out of bounds, the signer rejects that attempt and retries. This rejection step ensures that the final signature leaks no information about the secret key. It might take a few retries, but it guarantees the security of the output.

Despite these internal retries, Dilithium remains efficient and keeps data sizes low. While lattice-based schemes involve large vectors of polynomial coefficients, Dilithium compresses this data by rounding each coefficient and encoding only its high-order bits, effectively cutting the public-key size roughly in half. Thanks to this compression technique, Dilithium achieves one of the smallest combined public-key and signature sizes among post-quantum signature schemes that avoid Gaussian sampling.

In terms of security, breaking Dilithium would require solving either the Module-LWE problem or the Module-SIS problem (finding a short lattice relation). Both of these mathematical problems are believed infeasible to solve even with the aid of quantum computers—so Dilithium's signatures are considered quantum-resistant[11].

5.3 FIPS 205, Stateless Hash-Based Digital Signature Standard

The FIPS 205 standardization is based off of a submission titled "SPHINCS+", a signature scheme built entirely from hash functions rather than number-theory or lattices.[14] It stitches together two one-time constructs:

(1) WOTS (Winternitz One-Time Signature): using a short chain of hash outputs to sign exactly one message.
(2) FORS (Forest of Random Subsets): groups many tiny WOTS signatures by selecting random subsets of leaves and signing them, cutting down on overall size.

These building blocks sit under a Merkle hypertree, a multi-layer hash tree where each layer's root authenticates the layer below. To verify a signature, you follow a path from the signed leaf up through the hashes to the single public key root.

Security rests solely on hash-function properties (pre-image and collision resistance), which currently resist both classical and quantum attacks.[14]

5.4 Steps Moving Forward

5.4.1 *End-User Integration.*

- **Seamless updates:** Deliver post-quantum cryptography as routine software patches. Client applications automatically load the new libraries, so users won't notice any change in their workflow.
- **Hybrid key exchange:** Default to combined classical + PQC key-exchange schemes. This keeps backward compatibility and ensures handshakes succeed even if one algorithm isn't yet supported.

5.4.2 *Enterprise Migration.*

- **Inventory & risk assessment:** Map out all cryptographic assets, classify systems by sensitivity, and identify legacy components that may require special handling.
- **Pilot & validation:** In staging environments, test the performance (e.g. TLS handshake latency) and compatibility of PQC libraries with existing middleware and hardware.
- **Phased rollout:** Begin with high-value systems (target: complete by 2031), then extend to broader infrastructure in alignment with regular update cycles.
- **Training & documentation:** Update runbooks, DevOps playbooks, and conduct workshops so security teams understand new algorithms and key-management procedures.

5.4.3 *Regulatory Compliance.*

- **NIST/NCSC deadlines:** Align migration plans to meet NIST's recommendation for high-value data by 2031 and full transition by 2035 [21].
- **Certification & audit:** Integrate PQC into existing security frameworks and undergo compliance assessments to validate the migration.
- **Continuous monitoring:** Regularly update cryptographic libraries, track emerging vulnerabilities, and iterate on migration plans as standards evolve.

6 CONCLUSION

6.1 Difficulties

In our research of this topic, there were a couple glaring difficulties. Firstly, the quantum world is something that not even the most tenured researchers fully understand. As aptly summarized by Nobel laureate and one of the founders of quantum physics Richard Feynman in his lecture series titled "The Character of Physical Law", "I think I can safely say that nobody really understands quantum mechanics"[6]. Thus, getting a strong grasp on not only the basic concepts of quantum physics was incredibly difficult, much less understanding the methods used to circumvent it. Secondly, this topic is just so incredibly novel. As mentioned in the report, the NIST only started making standardizations for it in 2016, so not only is information somewhat difficult to find on it, what can be found is more-so oriented towards those who have some kind of background in advanced cryptography techniques, a deep understanding of mathematics, and some semblance of a grasp on quantum physics, none of which any of us have.

6.2 Findings

The coming era of large-scale quantum computing poses a direct threat to public-key schemes, RSA, ECC, and alike, as Shor's algorithm can break 2048-bit keys once devices reach on the order of 10^3–10^4 logical qubits (millions of physical qubits) [9]. While today's prototypes remain far from this scale, optimistic projections place a cryptographically relevant quantum computer within the next 10–20 years [21].

Our analysis underscores three key takeaways:

- **Mature post-quantum standards:** CRYSTALS-Kyber, Dilithium, and SPHINCS+ are NIST-approved, offering robust resistance to quantum attacks.
- **Clear migration roadmap:** Hybrid classical–PQC deployments today, targeted migration of high-value systems by 2031, and full transition by 2035 ensure continuity.
- **Continuous adaptation:** The inherently abstract nature of quantum mechanics makes communication and implementation challenging and evolving PQC recommendations demand ongoing review.

By adopting these proactive measures now, organizations can safeguard their data against the inevitable arrival of powerful quantum adversaries and avoid scrambling under the urgency of "Q-day."

6.3 Final Thoughts

Quantum computing is no longer just a theoretical concept—it is actively developing into a technology with real-world implications, particularly for cryptography. While today's quantum machines remain too primitive to pose an immediate threat, advancements are accelerating, and experts predict that within the next couple of decades, quantum computers capable of breaking widely used encryption standards could emerge. This presents a significant challenge, as much of our digital security infrastructure depends on cryptographic methods that quantum algorithms will render obsolete.

Despite this looming threat, the cryptography community is already working on solutions. Post-quantum cryptographic algorithms, designed to withstand quantum attacks, are under development and standardization and some have already been approved for use. The transition to quantum-safe encryption will be crucial in the coming years to ensure long-term data security. As quantum computing continues to advance, proactive adaptation will determine whether we stay ahead of its challenges or scramble to react once it is too late.

REFERENCES

[1] KU Leuven Alice Pellet-Mary. 2020. An LLL Algorithm for Module Lattices. https://simons.berkeley.edu/talks/lll-algorithm-module-lattices#:~:text=A%20module%20lattice%20is%20a,dimension%20over%20the%20integers%20ZZ.

[2] Anonymous. 2000. Number Field Sieve – from Wolfram MathWorld — mathworld.wolfram.com. https://mathworld.wolfram.com/NumberFieldSieve.html. [Accessed 13-03-2025].

[3] Anonymous. 2001. Quantum Entanglement and Information (Stanford Encyclopedia of Philosophy) — plato.stanford.edu. https://plato.stanford.edu/entries/qt-entangle/. [Accessed 13-03-2025].

[4] Anonymous. 2025. Microsoft's Majorana 1 chip carves new path for quantum computing - Source — news.microsoft.com. https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/. [Accessed 13-03-2025].

[5] Anonymous. 2025. RSA Algorithm in Cryptography - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/rsa-algorithm-cryptography/. [Accessed 13-03-2025].

[6] Sean Carroll. 2024. Even Physicists Don't Understand Quantum Mechanics. https://www.nytimes.com/2019/09/07/opinion/sunday/quantum-physics.html

[7] Sarah D. and Peter C. 2024. On the practical cost of Grover for AES key recovery. https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/on-practical-cost-of-grover.pdf?

[8] Martin Roetteler Damien Fortune and Michael Aiello. 2022. Predicting Q-Day and the impact of breaking RSA2048. https://www.secureworks.com/blog/predicting-q-day-and-impact-of-breaking-rsa2048

[9] Craig Gidney and Martin Ekera. 2021. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. https://quantum-journal.org/papers/q-2021-04-15-433/pdf/

[10] Andrea Laue. 2004. A Companion to Digital Humanities — companions.digitalhumanities.org. https://companions.digitalhumanities.org/DH/?chapter=content/9781405103213_chapter_13.html. [Accessed 13-03-2025].

[11] N/A. 2021. Dilithium. https://pq-crystals.org/dilithium/

[12] N/A. 2021. Kyber - How does it work? https://cryptopedia.dev/posts/kyber/

[13] N/A. 2023. Fiat-Shamir transformation. https://www.zkdocs.com/docs/zkdocs/protocol-primitives/fiat-shamir/

[14] N/A. 2024. Announcing Approval of Three Federal Information Processing Standards (FIPS) for Post-Quantum Cryptography. https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved

[15] N/A. 2024. An LLL Algorithm for Module Lattices. https://pq-crystals.org/index.shtml

[16] N/A. 2025. What Is Post-Quantum Cryptography? https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms

[17] Udara Pathum. 2024. CRYSTALS Kyber : The Key to Post-Quantum Encryption. https://medium.com/identity-beyond-borders/crystals-kyber-the-key-to-post-quantum-encryption-3154b305e7bd

[18] The MIT Press Reader. 2020. The Many-Worlds Theory, Explained — thereader.mitpress.mit.edu. https://thereader.mitpress.mit.edu/the-many-worlds-theory/. [Accessed 13-03-2025].

[19] Resonance. 2020. The History of Quantum Computing You Need to Know [2024] — thequantuminsider.com. https://thequantuminsider.com/2020/05/26/history-of-quantum-computing/. [Accessed 13-03-2025].

[20] Huzaifa Sidhpurwala. 2023. A Brief History of Cryptography — redhat.com. https://www.redhat.com/en/blog/brief-history-cryptography. [Accessed 13-03-2025].

[21] Matt Swayne. 2025. UK Sets Timeline, Road Map for Post-Quantum Cryptography Migration. https://thequantuminsider.com/2025/03/20/uk-sets-timeline-road-map-for-post-quantum-cryptography-migration/#:~:text=Insider%20Brief,remains%20robust%20against%20future%20threats.