SAGE Science Publishing

Data Science Approaches to Quantum Vulnerability Assessment and Post-Quantum Cryptography Schemes

Nur Alam Farhad Shakil[1], Imtiage Ahmed[2] and Ripan Mia[1]

[1]Information Technology, Washington University of Science and Technology, Alexandria, Virginia, USA
[2]Department of Computer Science, Fitchburg State University, USA

Submitted: 2023-10-12
Accepted: 2024-01-20
Published: 2024-02-05

Abstract

Quantum computing poses a threat to classical cryptographic systems, and proactive measures need to be taken to analyze vulnerabilities and design secure cryptographic protocols. This study provides an end-to-end review of data science methods being utilized for the *quantum-vulnerability-assessment* coupled with a survey of *post-quantum cryptography* (PQC) protocols. We begin by characterizing the quantum computing advances that imperil widely used public-key algorithms, highlighting the urgency of determining which digital assets are at risk. We then clarify how data-driven approaches—specifically, statistical analysis, machine learning, and large-scale cryptographic data audits—can identify probable weaknesses and guide risk remediation. These techniques enable the comprehension of the extent of quantum threats through cryptographic usage pattern analysis, and enable assessing algorithmic strength against quantum attack models, and prioritizing systems for near-term cryptographic upgrade. The highlight of the paper is a survey of the most well-known PQC algorithms that are secure against quantum attacks like lattice-based, code-based, multivariate, and hash-based constructions. For every class, we explain the tough issues underlying and the current state of security and performance. We discuss the real roadblocks to PQC deployment, ranging from implementation to performance concerns, and talk about how data analysis is can help inform efforts along the way. The study concludes by noting that protecting digital infrastructure from impending quantum capability requires the coming together of stringent data-driven research and cryptographic innovation. Integrating perspectives from cryptography and data science, this study explores how combining these disciplines can help enhance security in the emerging post-quantum world, while recognizing the associated challenges and complexities.

Keywords: Cryptographic transition, Data-driven analysis, Post-quantum cryptography, Quantum computing threats, Quantum-vulnerability-assessment, Risk mitigation, Secure algorithms

Introduction

Quantum computing is a computation model that exploits the rules of quantum mechanics to process information in fundamentally different ways than traditional computing (Tan *et al.* 2022). Quantum computing, in its basic sense, uses quantum bits, or *qubits*, which exist in superpositions of states so that many values can be encoded simultaneously. This is as opposed to classical bits that are binary, either 0 or 1.

A qubit is defined in terms of a linear combination of its basis states 0 and 1, expressed as

$$\psi = \alpha 0 + \beta 1,$$

where α and β are complex probability amplitudes which satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. This representation allows a qubit to be in coherent superposition of both states simultaneously, a property which is central to quantum computing. The state of a qubit can be described on the *Bloch sphere*, a unit sphere where any point on the surface represents a potential pure state of the qubit. The poles are associated with the conventional states 0 and 1, while equatorial points are represented by equal superpositions with relative phases.

Entanglement is yet another non-classical quantum mechanical feature used in quantum computing. If two qubits are entangled, the state of one influences the state of the other instantly regardless of how far apart they are. The non-classical correlation is defined by the Bell states, say,

$$\Phi^+ = \frac{1}{\sqrt{2}}(00 + 11),$$

which are maximally entangled two-qubit states. Entanglement makes quantum computers able to perform smart things that classical computers cannot because it makes them able to represent and manipulate exponentially large state spaces. (Raavi *et al.* 2021)

Quantum gates effect the implementations of quantum operations, thus denoting transformations that are unitary in nature

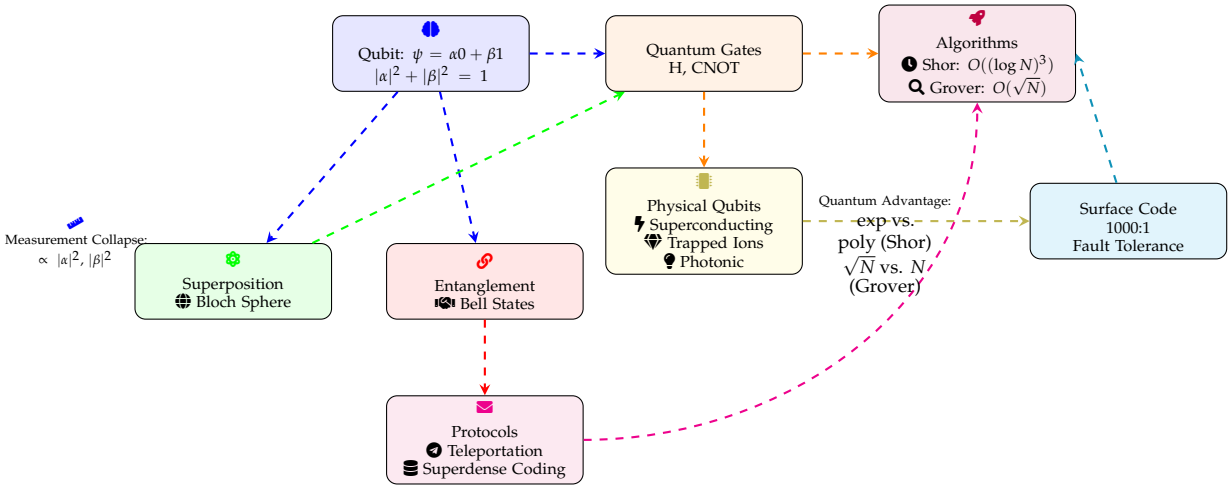

Figure 1 Architecture of quantum computing systems, depicting fundamental principles (qubit states, superposition, entanglement), frameworks (state representation, unitary transformations), physical implementations (qubit technologies), computational primitives (quantum gates), algorithmic applications (factorization, search), and supporting protocols (state transfer, communication). Arrows indicate dependencies and operational relationships between components.

and act on qubits. These gates manipulate the probability amplitudes of qubit states, and so form the building blocks of quantum circuits. Areas in which single-qubit gates are common include the Pauli-X, -Y, and -Z gates, as well as Hadamard gates that superpose states. Multi-qubit gates, like Controlled Naught (CNOT) gates, are used to entangle qubits. Depending on how these gates are put together in circuits, quantum algorithms can be designed to compute.

With the phenomena of superposition and entanglement, quantum algorithms gain superiority over all classical algorithms. *Shor's algorithm* outperforms classical methodologies and factors large integers vastly faster than known classical algorithms, leaving some implications for cryptographic schemes. Grover's algorithm provides a quadratic speedup concerning unstructured search problems; accordingly, the number of required searches is reduced from $O(N)$ to $O(\sqrt{N})$. This and other algorithms envision quantum computing as an instrument for solving certain problems more efficiently than classical computing systems.

The physical realization of a qubit can be accomplished by making use of several technologies, such as superconducting circuits, trapped ions, and photonic systems (Das and Ghosh 2023). Each provides specific benefits and tradeoffs with respect to coherence times, gate fidelities, and scalability. For example, superconducting qubits employ Josephson junctions to create anharmonic energy levels defining the qubit states. Trapped ion qubits exploit the internal electronic states of ions–confined in electromagnetic traps and which get supplemented by laser pulses–for encoding. Photonic qubits store information within the polarization or path of photons, which can hence operate at high speeds and room temperature.

Manipulation and measurement of the qubits require extremely precise control to maintain the coherence and minimize the errors. *Quantum error correction* codes prove to be very useful for detecting and correcting errors without directly measuring quantum information, thus preserving the still-inexistence states of the fragile quantum. In codingindirect logicals qubit into the entangled states of multiple physical qubits, the error localization can be done for decoherence and other noise sources. The fault-tolerant quantum computation will need a huge overhead in the number of physical qubits for effectively protecting logical qubits (Mone 2020).

Quantum computing enables new modes of information processing that include *q*uantum teleportation and *s*uperdense coding. Quantum teleportation allows transfer of a qubit state from one point to another point without transmitting the qubit itself using entanglement-in-coherent communication. Superdense coding allows two classical bits to be sent through just one qubit by exploiting entangled states held by the sender and receiver in bounds. These two protocols reveal great potentials of quantum information processing with zero classical analogs.

Quantum gates form the bases of quantum circuits, wherein qubits are transformed under unitary manipulations. These have a clear difference from classical logic gates that perform deterministic operations. On the other hand, quantum gates act on probability amplitudes, thus realize reversible and probabilistic computations. Commonly used gates include Pauli-X (analog to a classical NOT gate), Hadamard (enters into superposition), and CNOT (single two-qubit entangling gate). Thus quantum algorithms, for example, Shor's algorithm for integer factorization or Grover's algorithm for unstructured search, gain speedups on the scales of exponential or quadratic when compared to the classical counterpart using those gates, while Grover's algorithm accelerates database searches, reducing the time complexity from $O(N)$ to $O(\sqrt{N})$.

Cryptography is a fundamental building block of information security today, facilitating confidentiality, integrity, and authenticity of information for nearly all digital communication (Beri 2021). However, quantum computing presents an unparalleled threat to traditional cryptosystems in the form of quantum algorithmic attacks. Specifically, Shor's algorithm can be utilized by the quantum adversaries to efficiently solve problem instances (of integer factorization and discrete logarithms) thought to be infeasible to classical computers that are used in several widespread public-key cryptosystems. This impending "quantum threat" has prompted pressing questions that a large number of encryption and digital signature algorithms now in use—especially RSA, elliptic curve cryptography (ECC), and

Diffie–Hellman key exchange—would be vulnerable when sufficiently large quantum computers become accessible. While quantum computers of that scale to break these systems do not yet exist, the intense pace of quantum technological advancement and the unpredictable timescale for breakthroughs require proactive risk planning and action.

Two general lines of research have emerged in this situation: (1) assessing the vulnerabilities of current cryptographic infrastructures to quantum-enabled attackers, and (2) developing and implementing post-quantum cryptography (PQC) protocols that are secure against a quantum-enabled adversary. The first direction is to examine which systems and assets are most at risk and how soon those risks might be realized—a difficult issue that can be assisted by data science techniques to examine cryptographic use at scale and simulate or model quantum attack situations. The second direction is to seek cryptographic innovation, developing new algorithms from hard math problems that are believed to be resistant to known quantum algorithms. It is a question of balancing these two efforts: not only creating and standardizing quantum-resistant algorithms, but also understanding where and how to deploy them by identifying current points of weakness (Khodaiemehr *et al.* 2023a). This paradigm shift calls for an interdisciplinary solution—marrying cryptographic research with data-driven analysis and prudent risk management—to oversee the transition.

Data science is an answer to the initial problem of quantum vulnerability assessment. Contemporary businesses and infrastructures use cryptography in numerous applications (web traffic, financial transactions, software updates, IoT device authentication, etc.), generating vast volumes of cryptographic keys, protocols, and usage patterns. Data-driven approaches—statistical analysis, machine learning, and large-scale scanning—can go a long way toward charting this cryptographic landscape and identifying vulnerabilities.

For example, it can be possible to examine large collections of public keys or certificates and determine insufficient key sizes or deprecated algorithms in use that would be weaknesses in a post-quantum world. Machine learning routines can assist cryptanalysts in detecting non-random patterns or side-channel leakages that may not be detected via classical analysis. Data science basically gives tools to ingest sophisticated data regarding cryptographic implementations and prioritize remediation based on empirical risk factors (7 2023). In the meantime, the development of post-quantum cryptography has been accelerated to attain long-term solutions. Post-quantum cryptographic systems are new (or revived) cryptosystems that are designed in such a manner that even an attacker equipped with a quantum computer would find them computationally infeasible to break.

In the last decade, several families of PQC algorithms (e.g., lattice-based, code-based, multivariate polynomial, hash-based, etc.) have been suggested by researchers, and an intense public evaluation process—a good example of which is the National Institute of Standards and Technology (NIST) PQC Standardization Project—has been underway to test and standardize the most promising ones. They are intended to substitute or augment insecure public-key primitives in applications and protocols worldwide. This paper is a formal exploration of both of the foregoing dimensions: the use of data science in measuring and reducing quantum-related cryptographic vulnerabilities, and the technical status of leading post-quantum cryptography schemes. The remainder of the paper is organized as follows: we initially outline the nature of quantum threats to existing cryptographic algorithms and the extent of vulnerabilities in existing systems. Next, we proceed with data science-enabled approaches to exploring cryptographic vulnerabilities, including the extent to which big data and machine learning techniques are helping cryptanalysis and risk prediction in the age of quantum computing. This is followed by a comprehensive overview of post-quantum cryptographic schemes, including the major categories of quantum-resistant schemes and their foundations of security (Liu *et al.* 2022). We compare their relative strengths and weaknesses from the technical standpoint. Lastly, we discuss pragmatic aspects of transitioning to these new primitives and provide some final thoughts on the intersection of cryptography and data science in protecting the post-quantum world.

Quantum Computing and Cryptographic Vulnerabilities

A quantum computer is based on far different principles from a normal computer, employing quantum-mechanical phenomena such as superposition and entanglement to calculate certain things with awesome efficiency. For cryptography, the capability means seriously very powerful new algorithms to shatter the assumptions of security applied in most present-day cryptosystems. The most important one is Shor's algorithm, which performs factoring large numbers and discrete logarithms in polynomial time on a quantum computer. Shor's 1994 breakthrough theorem that any advanced enough quantum computer would be able to break the RSA encryption system (whose security relies on the problem of factoring) and elliptic curve cryptography (based on the discrete logarithm problem over elliptic curves). In practice, this would leave commonly used key exchange and digital signature schemes like RSA-based key transport, the Diffie–Hellman family of key exchange (finite-field DH and elliptic curve DH), and RSA/ECDSA digital signatures at risk if the attacker had a large-scale quantum computer. An RSA or binary software authenticated with ECDSA message is susceptible to being spoofed or decrypted by running Shor's algorithm on the associated public key data (Lyssenko and Komolafe 2023). The implications of such a capability cannot be overstated: it would undermine the integrity and privacy guarantees that electronic commerce, communication, and records-of-things rely on today.

A second important quantum algorithm, invented by Grover in 1996, affects symmetric cryptography and hashing. Grover's algorithm provides a quadratic speed-up for unstructured search problems, which in cryptographic terms can halve the complexity of brute-force search of keys or hash preimage search. Unlike Shor's devastating impact on public-key technology, Grover's algorithm does not entirely undermine symmetric ciphers and hash functions but effectively halves the security level of their exponent. A 128-bit symmetric key cipher (like AES-128), which would take 2^{128} operations to brute-force, might be searched with approximately 2^{64} operations on a quantum computer employing Grover's algorithm. In order to have the same security margins in a post-quantum setting, symmetric algorithms would need to double their key sizes (for example, from AES-128 to AES-256) and hash functions would need larger output sizes (for example, 256-bit output instead of a 128-bit output). Fortunately, the sort of installations can be done with current technology and a variety of symmetric algorithms (SHA-256, SHA-3, AES-256, etc.) are already in general use or can be made to function retrofitted without needing fundamental recasting. Symmetric message digests and encryptions are thus **quantum-resistant** with primarily cosmetic enhancements.

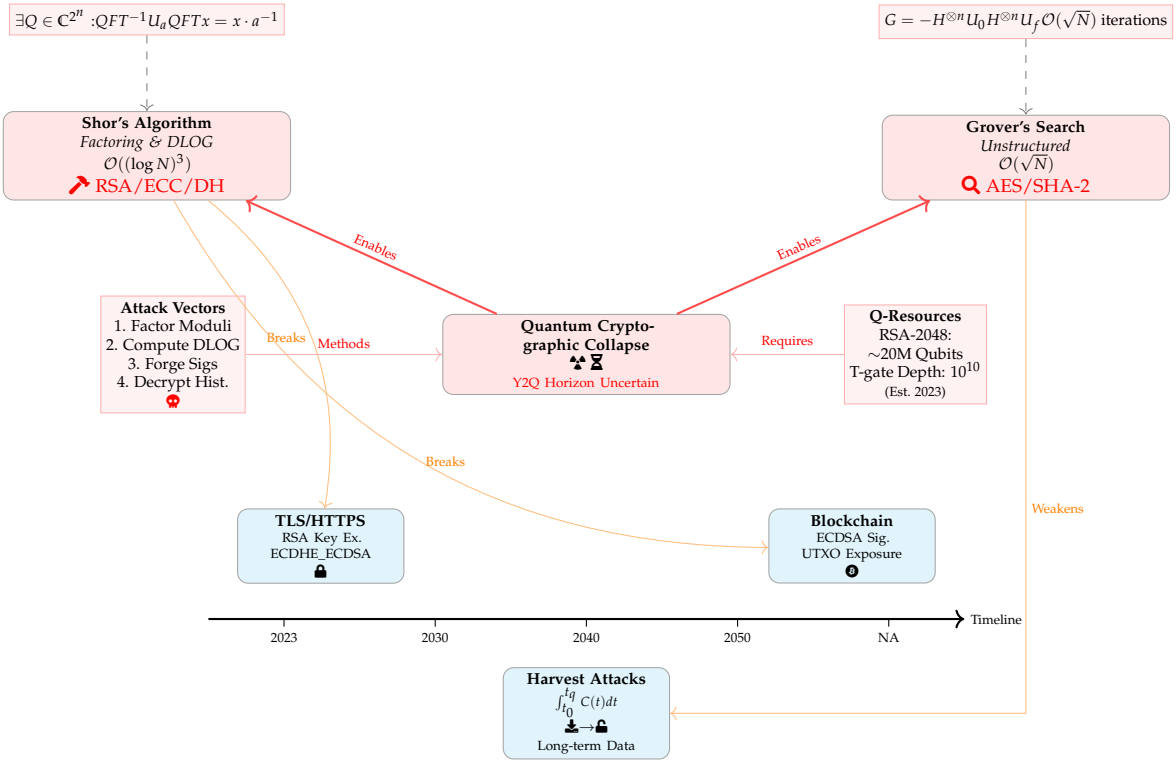

Figure 2 Shows quantum vulnerability depicting core cryptographic risks, vulnerable implementations, and relationships. It contains foundations, attack vectors, resource requirements, and temporal dimensions of quantum threats. Arrows indicate exploit pathways and theoretical dependencies.

There isn't an easy solution, however, for salvaging classic public-key schemes like RSA or ECC – new algorithms must be invented (Das *et al.* 2023). (Luckily, such symmetric primitive measures are already well known and, in some cases, already in place – i.e., a large majority of security protocols have moved to 256-bit symmetric keys and 384-bit or larger hashes to further future-proof them.)

The scope of vulnerability induced by quantum is broad. Public-key cryptography forms the foundation of numerous security protocols: TLS and HTTPS use RSA or ECC to exchange keys and certificate signatures, secure email (PGP/S/MIME) often relies on RSA keys, digital code signing of software updates is done with RSA/ECDSA in most implementations, and cryptocurrencies as well as blockchain systems utilize ECC-based digital signatures to verify transactions. All these systems would be at risk if their public-key algorithms, which form the basis of them, can be broken by quantum computation. For instance, an attacker with a quantum computer could potentially decrypt intercepted TLS traffic secured by RSA key exchange, impersonate a server by forging a certificate signature, or steal cryptocurrency by computing private keys from a leaked blockchain public key. It is worth noting that protocols employing ephemeral Diffie–Hellman exchanges to derive session keys (providing forward secrecy) limit the amount of quantum compromise—no single static key can decrypt enormous amounts of traffic—although they do not eliminate the underlying vulnerability, as a quantum attacker can still break each session's ephemeral key with sufficient computing power. These illustrate why a future quantum attacker represents a systemic cybersecurity threat: it isn't a single specific protocol that is broken, but more or less any security mechanism depending on the supposed hardness of discrete logs or factoring.

While, as far as the author knows at the time of writing, no quantum computers exist with the numbers of qubits and low enough error rates required to execute Shor's algorithm on RSA-2048 (a standard key size), quantum computing is moving ahead full-steam (Banafa 2023). Working prototype quantum processors have achieved order hundreds of physical qubits, and while noisy and error-prone, effort towards quantum error correction and scaling is in progress. Experts vary as to when a "cryptographically significant" quantum computer – one capable of breaching real-world cryptographic keys – will be built. Some put it a decade or two in the future, but more conservative estimates suggest that it may take longer. Whatever the exact timeline, there is overall agreement that the threat is severe enough to necessitate preparedness now. One of those reasons for this feeling of urgency is the notion of "harvest-now, decrypt-later" attacks: attackers might intercept and store encrypted data now, with the expectation of decrypting it later when resources on quantum computers will be made available. Sensitive information that must remain confidential for decades (such as government data, long-term intellectual property, or private data with long-term privacy requirements) can be compromised unless current-day encryption is updated in a timely manner. Similarly, infrastructure that cannot be easily updated – such as embedded systems and critical hardware that rely on hard-coded cryptographic keys – could be hacked retroactively once quantum decryption becomes feasible.

In view of such thoughts, quantum computing weaknesses are not a remote theoretical danger but an immediate current threat (Aoun and Tarifi 2004). It has been observed by the cryptography community that essentially all of the present public-

key protocols possess an exponential quantity of remaining lifetime, dictated not by classical computational progress (in terms of Moore's Law) but by quantum breakthrough. Thus, there is the urgent need to assess what systems are most at risk, project how long it would take those vulnerabilities to be exploitable, and implement countermeasures. This quantum vulnerability assessment informs wise decision-making about where to put security upgrade investment. It requires accurate knowledge of the properties of quantum algorithms (knowing what breaks) and the current state of cryptographic deployment (knowing what's exposed). The next few sections address how data-oriented techniques can be used to aid in this analysis, and then what alternative cryptographic schemes are currently being developed to secure the post-quantum world.

Data Science Techniques for Cryptographic Vulnerability Analysis

With more intricate and larger cryptographic systems, data science has proven very useful in exposing vulnerabilities and improving cryptanalysis. In standard cryptanalysis, the strength of a cipher is usually determined through examination of extensive amounts of information – for instance, testing large numbers of plaintext–ciphertext pairs for statistical bias that can betray information on the secret key (Pandya 2015). This data-driven activity by its nature has been enhanced in recent decades by sophisticated computing techniques, e.g., automated data analysis, machine learning, leading to nothing short of "data-driven cryptanalysis." Data science and cryptography are distinct disciplines but converge in significant areas when the security is decided:

Cryptographic data can be used to train algorithms in machine learning that can identify patterns in cryptographic data that are difficult to recognize through human analysis. For instance, neural networks have been employed as distinguisher machines in differential cryptanalysis, which is a technique applied to cryptanalyze symmetric ciphers. Through training a neural model on numerous plaintexts and corresponding ciphertexts from a cipher (or its truncated-round variant), researchers have demonstrated that the model can potentially learn to distinguish very faint non-random patterns indicating the cipher's structure or even learn partial information regarding the key. Such an attack, termed as neural cryptanalysis, has proved capable of augmenting attacks against certain lightweight or truncated-round ciphers. Although it hasn't broken hard-evidence standard algorithms like full AES yet, it's doing well as a cipher evaluation tool – essentially leveraging data-driven pattern recognition to enhance human-crafted cryptanalytic techniques. Machine learning is also used in analyzing outputs of cryptographic algorithms under novel circumstances, perhaps those of post-quantum environments, in an effort to spot anomalies or weaknesses that wouldn't be revealed by theoretical analysis alone. In fact, scientists now begin to use such tactics on post-quantum algorithms themselves – e.g., training models to search for even minimal biases or correlations in the output of lattice-based cryptography or other schemes that would indicate potential weaknesses (Bhushan 2022). No practical attack yet arose from such activity, but this extra level of testing does serve to make more certain that new cryptosystems have no hidden exploitable patterns.

Cryptographic vulnerabilities typically appear not in the ideal algorithm but in its implementation. Side-channel attacks exploit quantifiable side-channel information revealed by computation (e.g., time, energy consumption, or electromagnetic radiation) to derive secret keys. Here, data science intervenes to examine and learn from such side-channel signals. In a modern side-channel attack, the attacker might collect thousands or millions of power traces from a device that is performing cryptographic operations. With the help of signal processing and machine learning classifiers on the large database of data, the attacker is able to successfully "learn" the correlation between small variations in the trace and the secret key bits, generally succeeding where simpler analytical methods fail. New developments in deep learning have significantly improved side-channel attacks against symmetric cryptography (such as AES), making it possible to steal keys using fewer traces and less domain expertise in feature engineering than classic methods. The same data-driven techniques are being investigated in the context of post-quantum cryptographic implementations; e.g., investigating the behavior of power consumption pattern of a lattice-based decryption algorithm to ascertain whether machine learning can identify the impact of secret key on it (Berezin 2007). On the defensive side, implementers use similar statistical techniques to determine if an implementation is leaking information, essentially examining the implementation as a data set to test for correlations between secrets and observable values.

Algorithm 1 Neural Distinguisher Training

Require: Dataset $\mathcal{D} = \{(P_i, C_i, y_i)\}$, model f_θ, learning rate η, epochs E
Ensure: Trained parameters θ^*
Initialize θ randomly **for** $e \leftarrow 1$ *to* E **do**
    Shuffle $\mathcal{D}$ **for** *each minibatch* $B \subset \mathcal{D}$ **do**
        Compute predictions $\hat{y} \leftarrow f_\theta(P)$ for $(P, y) \in B$
Compute loss $L \leftarrow \text{CrossEntropy}(\hat{y}, y)$
$\theta \leftarrow \theta - \eta \nabla_\theta L$
(Sun and Huang 2022)
return θ

At a level higher than the algorithmic, data science techniques have been used to analyze real-world cryptographic artifacts at scale, revealing systemic vulnerabilities. A fine example is public key and digital certificate analysis garnered from the Internet. Millions of RSA public keys (for instance, from TLS certificates, PGP keys, or IoT device firmware) have been accumulated by researchers and then computed batch GCD on all pairs of moduli – something only made feasible through effective processing of data and algorithmic optimizations. In doing so, they discovered that most RSA keys (generated on poorly configured machines or software) share common prime factors between them. Any two keys with a shared prime factor can be easily compromised by computing their greatest common divisor (GCD), effectively factoring both moduli. This large-data experiment, a data-driven vulnerability discovery, uncovered real instances of compromised keys being utilized that were impossible to detect from the analysis of any single key in isolation. It highlights how large-data methods can uncover vulnerabilities in this instance (garbage random number generation that generates non-distinct primes) that undermine cryptographic security (Bandi *et al.* 2023). Similarly, research into cryptographic protocols "in the wild" – such as scanning hundreds of thousands of TLS servers for ciphersuites and key exchange algorithms they support – gives a glimpse into how many systems remain

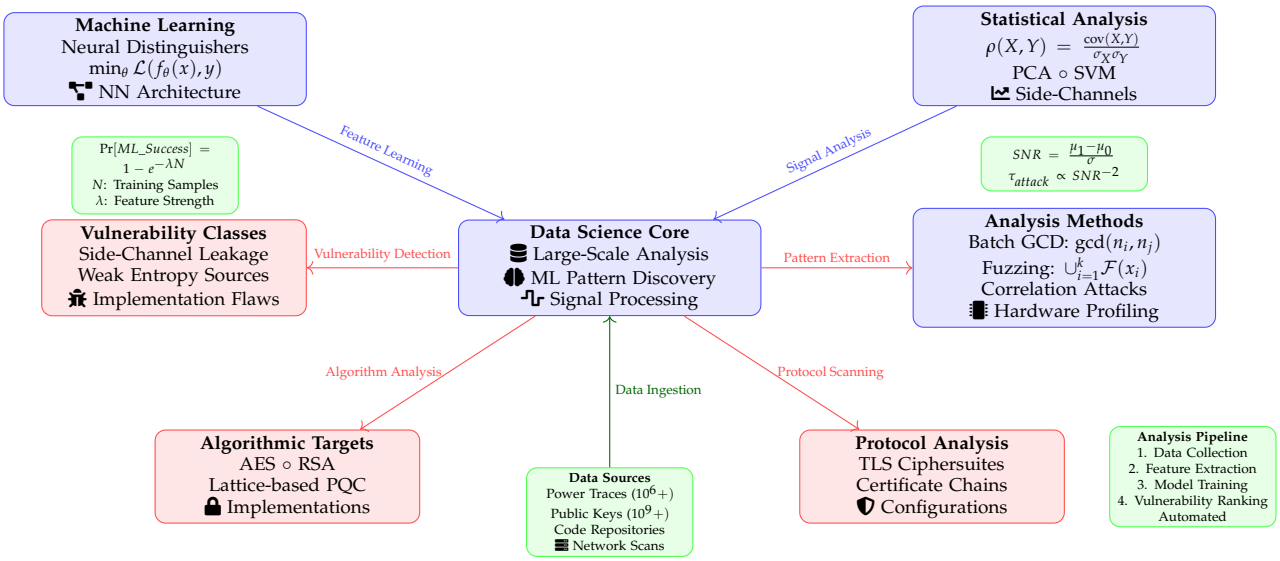

Figure 3 Data science techniques in cryptographic vulnerability analysis, showing core methods, data sources, and vulnerability targets. Arrows indicate data flows and analytical relationships between machine learning approaches, statistical methods, and cryptographic system components.

founded on algorithms that would be vulnerable to quantum attacks. Empirical surveys like these guide risk estimates by quantifying the exposure of cryptography.

Data science also intersects with software security analysis tools that search for crypto vulnerabilities. Techniques like fuzz testing (producing massive inputs to test a system autonomously) can be steered by machine learning to attempt edge cases that have the potential to lead to logical flaws in crypto implementations. While fuzzing isn't really cryptanalysis, it has revealed issues like faulty error handling or leakage in libraries that an attacker could potentially use. Pattern matching algorithms help to scan source code or binaries for known insecure cryptography patterns (e.g., identifying use of old algorithms or failure to use randomness) in large codebases. In preparing for the quantum era, organizations are employing automated scanners to identify where weak algorithms (RSA, ECC, etc.) are being used within their software and hardware stacks. It is essentially a technology stack data mining exercise that provides data that security teams can utilize to prioritize remediation (Raina and Kaushal 2019).

In all of them, the common denominator is the application of massive amounts of data and computationally aided pattern discovery in order to identify security weaknesses. Cryptanalysis progressed from being a craft discipline to one that remains accessible to computation assistance; academics today apply AI and big data not to replace theoretical analysis but to supplement it. Data science techniques can try out cryptographic security in a variety of ways pure mathematics can't – by brute-force find-out, by learning from experience, or by gathering insight from large samples. Particularly with quantum-era uncertainties to face, this empiric perspective is worth it. We may not know yet exactly how to quantifiably analyze the security of a given post-quantum candidate or implementation nuance, but broad testing and data-based understanding can provide world assurance or point out where such assurance fails.

Algorithm 2 Correlation Power Analysis (CPA)

Require: Power traces $\{T_i\}_{i=1}^{N}$, plaintexts $\{P_i\}$, leakage model $\ell(\cdot)$
Ensure: Key byte guess k^* **for** $k \leftarrow 0$ *to* 255 **do**
1: $v_j \leftarrow \ell(h_k(P_j))$ for $j = 1, \ldots, N$ (Schindler 2022)
2: $\text{corr}[k] \leftarrow \left|\text{PearsonCorr}(v, \{T_j\})\right|$
3:
4: $k^* \leftarrow \arg\max_k \text{corr}[k]$
5: **return** k^* =0

It has to be said that data science can serve as much to point out weaknesses as to further enable confidence in absence of weakness on detailed examination. For example, if a new cryptographic algorithm is subjected to intense machine learning attacks with no exploitable patterns being discovered, such an outcome gives one confidence (but not evidence) in the algorithm's strength. In essence, the infusion of data science into vulnerability cryptanalysis has emerged as a central piece of the modern security research toolkit. It helps in the identification of cryptographic weaknesses—either in algorithms, implementations, or usage—before they can be exploited by attackers, thereby informing the development of more secure systems and the wise transition to quantum-resistant cryptography.

Algorithm 3 Batch GCD Key Recovery

Require: RSA moduli $\{n_i\}_{i=1}^{m}$
Ensure: List of compromised indices and factors
$N \leftarrow \prod_{i=1}^{m} n_i$ **for** $i \leftarrow 1$ *to* m **do**
  $g_i \leftarrow \gcd(n_i,\ N/n_i)$ (Liu and Ren 2023) **if** $1 < g_i < n_i$ **then**
    Record (i, g_i)
return all recorded (i, g_i) =0

Algorithm 4 ML-Guided Fuzz Testing

Require: Target B, initial corpus $C \leftarrow C_0$, ML model M, budget
Ensure: Set of crashes R
$R \leftarrow \varnothing$ **while** *budget not exhausted* **do**
  Compute features $\phi(x)$ for each $x \in C$ (Ravi *et al.* 2022)
  Score $s(x) \leftarrow M(\phi(x))$
  Select top-k inputs $S \subset C$ by $s(x)$ **for** *each* $x \in S$ **do**
    Generate mutations $\mu(x)$
    Run B on $\mu(x)$; if crash, add to R **if** *new coverage* **then**
      $C \leftarrow C \cup \{\mu(x)\}$
(Miller 2012a)
return R

Algorithm 5 TLS Cipher-Suite Scanner

Require: Endpoints $\{e_i\}_{i=1}^{n}$, ciphersuite catalog $\mathcal{C}$
Ensure: Mapping $e_i \mapsto S_i$ **for** *each endpoint* e **do**
  $S_e \leftarrow \varnothing$ **for** *each* $c \in \mathcal{C}$ **do**
    TLS handshake to e with only c succeeds
  $S_e \leftarrow S_e \cup \{c\}$
return $\{e \mapsto S_e\}$ =0

Assessing and Prioritizing Quantum Security Risks

Confronted with the reality that most public-key cryptography currently employed will eventually be vulnerable to quantum attackers, organizations and security professionals must undertake quantum vulnerability assessments to determine where their greatest weaknesses lie. It is a naturally multidisciplinary task, encompassing technical cryptographic knowledge mixed with data inventory, risk management, and forward thinking (Brotherton and Gupta 2023). The objective is to identify what assets are at risk, measure the impact and urgency of the threat, and prioritize mitigation to enable a smooth transition to post-quantum security before any catastrophic breach.

The first step in quantum risk assessment is to have insight into all cryptographic keys and algorithms being used in a system or organization. That would mean enumerating cryptographic assets such as the algorithms of network protocol (TLS, VPN, secure shell, and so on), the encryption algorithms and key lengths of stored data, the cryptography types of used digital signatures used in code-signing and for documents, even the cryptographic primitives of third-party software or hardware devices. In large settings, such a catalog would be ginormous in scope, from public web servers to internal databases and edge appliances deployed out in the field. Techniques such as code scanning and network scanning (as previously outlined) are very useful in this discovery phase of work by deconstructing configuration files, binaries, and network handshake data to determine instances of algorithms such as RSA, ECC, or other recognized quantum-vulnerable schemes. The deliverable of this task is a step-by-step walk-through of where and how vulnerable cryptography can be found, including contextual information (e.g., key sizes, usage rates, whether data encrypted with those keys are sensitive in the long term). Above all, there is already in place government policy and industry regulation that mandates conducting such inventories, a demonstration of a converging consensus around the identification of quantum-vulnerable systems as an intermediate step towards defense.

Algorithm 6 Cryptographic Asset Inventory

Require: Set of targets $\mathcal{T}$ (servers, codebases, devices), scan tools S
Ensure: Asset catalog $\mathcal{A}$
$\mathcal{A} \leftarrow \varnothing$ **for** *each target* $t \in \mathcal{T}$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \text{discoverAlgorithms}(t, S)$
$\mathcal{A} \leftarrow \mathcal{A} \cup \text{extractKeys}(t, S)$
$\mathcal{A} \leftarrow \mathcal{A} \cup \text{mapProtocols}(t, S)$
(Gupta *et al.* 2021)
return $\mathcal{A}$ =0

Algorithm 7 Quantum Risk Characterization

Require: Asset catalog $\mathcal{A}$, factor weights $w_{\text{sens}}, w_{\text{role}}, w_{\text{vis}}, w_{\text{key}}$
Ensure: Priority list $\mathcal{P}$ **for** *each asset* $a \in \mathcal{A}$ **do**
  $\text{Impact} \leftarrow w_{\text{sens}} \cdot \text{sensitivity}(a) \ + \ w_{\text{role}} \cdot \text{role}(a)$
$\text{Likelihood} \leftarrow \frac{1}{\text{updatability}(a)}$
$\text{Mitigation} \leftarrow \text{keyStrength}(a)$
$\text{Score}[a] \leftarrow \frac{\text{Impact} \times \text{Likelihood}}{\text{Mitigation}}$ ▷ Higher means more urgent
$X \leftarrow \text{migrationTime}(a)$
$Y \leftarrow \text{dataLifetime}(a)$ **if** $X + Y > Q$ **then**
  mark a as HIGHRISK
$\mathcal{P} \leftarrow$ sort descending by $\text{Score}(\mathcal{A})$
return $\mathcal{P}$ =0

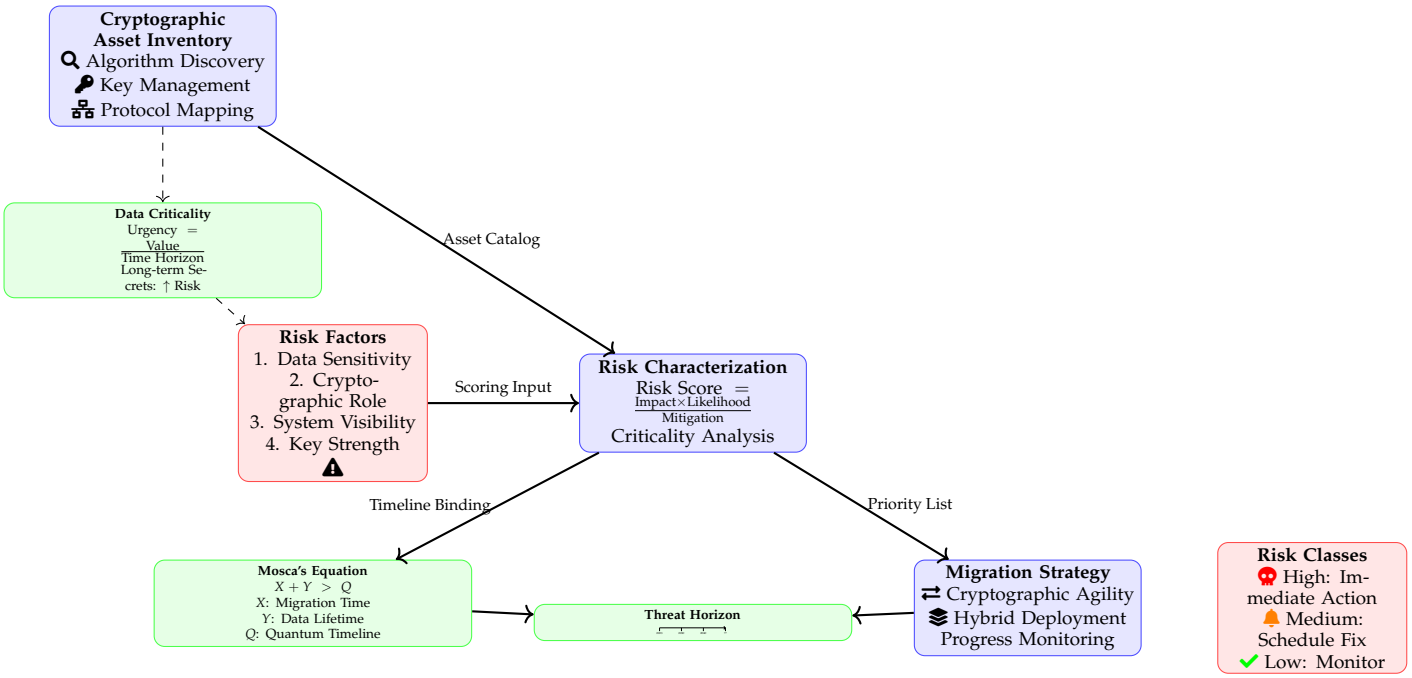

Figure 4 Quantum security risk assessment framework showing inventory processes, risk factors, timeline considerations, and mitigation planning. Arrows indicate workflow dependencies between asset discovery, risk scoring, timeline projection, and migration strategy components.

Algorithm 8 Migration Strategy Planning

Require: Priority list $\mathcal{P}$, post-quantum suite catalog $\mathcal{Q}$
Ensure: Migration plan M
$M \leftarrow [\,]$ **for** *each asset* a *in top tiers of* $\mathcal{P}$ **do**
  $p \leftarrow \text{selectPQAlgorithm}(a, \mathcal{Q})$
  $m_a.\text{mode} \leftarrow \{\text{HYBRID}, \text{AGILE}\}$
  $m_a.\text{algorithm} \leftarrow p$
  $m_a.\text{deadline} \leftarrow \text{computeDeadline}(a)$
  append m_a to M
return M =0

Risk characterization: Not every vulnerable application to quantum is as urgent to repair. Once the vulnerable algorithms have been identified in the inventory, each instance needs to be assessed for urgency and impact of quantum threat. A good model for such assessment considers factors such as: Sensitivity and timescale of data: If the cryptography is protecting data of very high sensitivity that needs to remain a secret for many years (e.g., health data, state secrets, or intellectual property of long-term importance), then even the probability of quantum decryption in ten years contributes to the risk – this is an argument for "harvest-now, decrypt-later" concern. Conversely, encrypting low-value or transient data (e.g., session data that will be stale in a day or a few minutes) is perhaps less important. Breaking digital signature and authentication weaknesses can lead to immediate catastrophic consequences – e.g., being able to produce a fake digital signature on a patch or certificate can allow active malicious attacks when a quantum computer is within an attacker's reach. Or it may be at best possible to find historic data if used in an evaporative environment. In general, identity threats (certificates, signatures) and integrity threats (authentication keys) are more worrying as they facilitate direct active attacks like impersonation or system compromise (Wang and Lai 2023; Balakumar 2022). Visible systems (widely distributed software, public servers) provide an increased attack surface if insecure. Similarly, difficult-to-repair or upgrade systems (such as embedded systems in critical infrastructure, IoT devices with relatively rarely updated firmware, or hardware crypto modules that are costly to replace) are at higher risk since they can remain vulnerable for longer. If they are weak-crypto-based, then it would be reasonable for an organization to consider them high-risk assets since they represent "obdurate" sources of vulnerability that will be attacked long after the remainder of the world has passed them by. Technical factors such as the key sizes employed. Specifically, RSA 1024-bit keys are already classically breakable in some cases and inconsequentially tiny for any plausible quantum attacker, whereas RSA 4096-bit keys, although still ultimately vulnerable to Shor's algorithm, might be beyond the capabilities of a somewhat bigger quantum computer. They are all ultimately vulnerable to quantum attack, but lower-level keys might be broken quicker and provide some sort of measure in which to replace them.

Upon evaluating these factors, a level of risk or priority may be ranked for each discovered vulnerability. The majority of firms utilize a scoring process or tiered classification (i.e., high, medium, low risk) to prioritize quantum vulnerabilities (Gupta 2017). For instance, a detailed evaluation might conclude that "code-signing certificates for automatic software updates" is high-priority risk (high impact, need long-term integrity, device updates are likely to be infrequent), whereas "VPN encryption for routine internal communications" might be medium (sensitive, but possibly mitigated by forward secrecy or ease to update), and "temporary session keys for public web traffic" might be low (short-lived data, forward-secret protocols like TLS 1.3 already do damage limiting). Such classification enables

resources and efforts to be initially directed towards the most mission-critical domains.

In practice, organizations can incorporate these considerations into existing risk management procedures. Through translating technical requirements into business impact terms, certain frameworks help decision-makers prioritize quantum mitigation alongside other enterprise risks. They stress that addressing the quantum threat is not merely a cryptographic issue but also one of governance and strategy.

One of the challenging aspects of quantum risk management is timeline uncertainty in advances in quantum computing. Since nobody can reliably forecast when quantum computers will be capable of breaking specific cryptosystems, organizations must plan in uncertainty (Sharma and Ramachandran 2021). A realistic plan is to make conservative estimates – assume worst-case but plausible timing for when there will be a cryptographically important quantum computer (perhaps in the 2030s) – and then transition all valuable assets long before then. There is a rule-of-thumb widely cited, and sometimes referred to as Mosca's rule or Mosca's equation, stating that if X is the number of years that would be expected before their systems turn over to all-new cryptography and Y is the number of years one needs to secure data, one should mark any cryptographic scheme for which X + Y reaches into the predicted time of arrival of quantum breaking capability as unsafe. That is, if it takes 5 years to upgrade them all, and details about those systems need to be kept hidden for 10 years, then one can confidently assume that no quantum computer that can execute the relevant attack will probably be found in the next 15 years. Governments and organizations have begun to set target dates on which they intend to complete transitions on such a basis. Most national security agencies are now demanding new systems to be "quantum-resistant" by default and asking for crypto vulnerability inventories as reporting, an institutional acknowledgment of the clock ticking on classical cryptography.

After risks are identified and ranked, the next is to design a particular migration plan to post-quantum alternatives. This involves selecting appropriate post-quantum algorithms (which we discuss in the following sections) to replace or supplement the compromised cryptography. Cryptographic agility – the ability of a system to swap out cryptographic algorithms comparatively easily – is one of the key concepts in this planning (Miller 2012b). Cryptographically agile systems possess modular, flexible cryptographic implementations (for instance, by using standard libraries and protocols that can support a variety of algorithm options). Cryptographic agility provides the assurance that new algorithms may be added to protocols (or run in parallel with current algorithms) once standards for PQC are established, without needing to redesign everything. In practice, migration might take place in phases: initially deploying hybrids of classical and post-quantum algorithms with an additional layer of security, and eventually phasing out the classical algorithms. As an example, an secure communications protocol might initiate two parallel key exchanges – an ECDH and a post-quantum key encapsulation – in such a way that the shared secret will be safe so long as one of the schemes is not compromised. This guards against the threat of the quantum attack and also against some as-yet undiscovered weaknesses of the new PQC scheme.

For data-driven monitoring when planning mitigations, monitoring of progress can be helpful. An organization might have a dashboard of all identified vulnerable components and monitor, over time, how many have been replaced with quantum-safe ones (Badhwar 2021). Such a dashboard would be provided by continuous scanning and testing, essentially using data analytics to the task of cryptographic transition. Via regularly scanning systems for legacy algorithm usage, one is able to ensure that risk reduction efforts are having an effect and that no unseen legacy usage is lurking (a common problem when dealing with sprawling IT environments).

Prioritizing and assessing quantum security risks is a question of knowing what is at risk, how it is at risk, and when it needs to be defended, and acting on that knowledge. The use of data science on the process – from discovery of vulnerable instances on through modeling timelines and tracking remediation – gives a quantitative basis to what could otherwise be an overwhelming task. It allows security planners to replace guesswork with informed analysis and to demonstrate progress in measurable terms. Equipped with a snapshot of their risk profile, organizations can now tackle the solution side of the equation: selecting and deploying post-quantum cryptography to remove these vulnerabilities in advance of the quantum threat's arrival.

Post-Quantum Cryptography Schemes

Designing cryptographic algorithms that can withstand quantum attacks has been a major focus of research in the past decade. Unlike quantum cryptography (such as quantum key distribution, which relies on the laws of physics and requires specialized hardware), post-quantum cryptography (PQC) refers to conventional (classical) cryptographic algorithms that run on today's digital computers but are built on mathematical problems believed to be resistant to quantum algorithms (Reisner and Donkor 2000). These schemes aim to replace vulnerable public-key algorithms like RSA and ECC in applications ranging from internet communication to digital signatures and beyond. A variety of mathematical approaches have been explored, and several categories of post-quantum algorithms have emerged as front-runners. In 2016, the U.S. National Institute of Standards and Technology (NIST) initiated an open competition-like process to evaluate and standardize PQC algorithms; similar efforts have taken place in academic and industry circles worldwide. As of the mid-2020s, this process has identified a handful of leading candidates that are likely to form the core of our cryptographic infrastructure in the quantum era. Each of these candidates falls into one of a few broad families of hard mathematical problems. Below, we provide an overview of the main categories of PQC schemes, along with examples and key characteristics of each. One advantage of the post-quantum approach is that it leverages existing communication infrastructure and can be implemented via software updates, in contrast to quantum cryptographic methods that require physical quantum channels (Khan *et al.* 2023).

Lattice-Based Cryptography

Lattice-based cryptography is likely the most popular and researched post-quantum algorithm family. For a lattice here, one can think of a regular grid of points in a multi-dimensional space (e.g., all integer linear combinations of some fixed set of basis vectors). Security of lattice-based schemes relies on, for instance, the Shortest Vector Problem (SVP) or the Learning With Errors (LWE) problem. SVP asks: with a basis of a high-dimensional lattice, find the shortest non-zero vector in the lattice – a problem which appears to be very hard when the lattice dimension is large (even for quantum computers). LWE, though, is the task

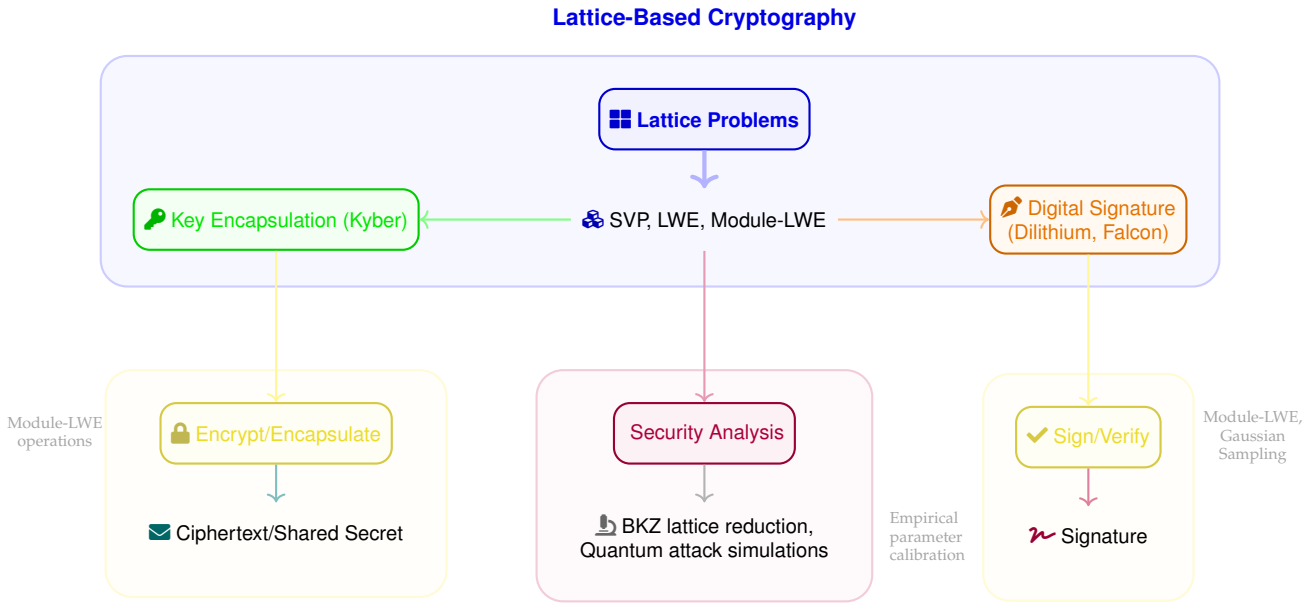

Figure 5 Overview of lattice-based cryptography illustrating the key encapsulation and signature schemes derived from fundamental lattice problems (SVP, LWE). It also shows security analysis procedures using lattice reduction techniques like BKZ and quantum attack simulations.

of solving a perturbed or "noisy" system of linear equations; it is also extremely well connected theoretically to a wide variety of lattice problems (worst-case reductions for hardness show that it is as hard to break LWE as to solve some instances of SVP in the worst case). Intuitively, these challenges remain resistant because adding noise or finding short vectors in high dimensions spoils approaches that otherwise can solve structured linear problems.

Cryptographically, lattices have been a rich source of constructions (Khodaiemehr *et al.* 2023b). Among the very first lattice-based cryptosystems is NTRU (circa late 1990s), a polynomial lattice-based scheme yet to be broken by any known classical or quantum attacks since its inception some decades ago. Lattice-based schemes have become increasingly dominant of the outcome of the NIST PQC project more recently. As an example, CRYSTALS-Kyber is a lattice-based key encapsulation mechanism (essentially an encryption scheme for establishment of shared secrets) selected for standardization. Kyber's design is based on a type of LWE known as Module-LWE, which is secure and fast by nesting the underlying lattices such that they facilitate rapid arithmetic (the same design in elliptic curves that made ECC cost-effective). For signatures, CRYSTALS-Dilithium (yet another lattice-based scheme, founded on the difficulty of the Module-LWE and Module-SIS problems) has been selected as a lead post-quantum signature scheme. Another lattice-based signature, Falcon (using the NTRU lattice problem and the hash-and-sign paradigm with Gaussian sampling), is equally secure with even shorter signature lengths, but with even more complex implementation requirements. As an example, Falcon uses floating-point arithmetic in its signature generation (in order to generate from a Gaussian distribution), which is picky to implement in a constant-time, secure manner on any given platform. This is a signal for the trade-offs that designers balance between performance and conciseness against output size in lattice signature design (Ali *et al.* 2023). Such lattice-based schemes share the characteristic of having somewhat larger public keys and signatures than RSA/ECC ones but staying in manageable sizes (e.g., a Dilithium public key is tens of kilobytes and a signature is around 2–3 kilobytes, which is perfectly alright for most applications). Also, their computational expense is high – encryption, decryption, signing, and verification can be done in milliseconds or less on general-purpose processors, typically faster than traditional RSA (albeit not in size).

There are numerous reasons why lattice-based cryptography is appealing. First, lattice problems have resisted extremely powerful cryptanalytic attacks; even in the aftermath of quantum algorithm research, no quantum algorithm for solving general lattice problems (like SVP or LWE) has been found, except for some quantum versions of lattice reduction that don't seem to threaten well-constructed cryptographic lattices. Second, lattice schemes are worst-case to average-case reductions, providing a strong theoretical foundation: it would be impossible to compromise the cryptosystem and the underlying lattice problem in the worst case. Third, lattices are versatile – they support not only simple signatures and encryption, but also stronger primitives like fully homomorphic encryption and identity-based encryption (but uses beyond the scope of this paper). Besides, the actual security of lattice schemes is evaluated through aggressive use of the strongest known attacks. When running lattice reduction algorithms on smaller instances and extrapolating to larger dimensions, designers determine parameters that offer an adequate safety margin (e.g., aiming for attacks to require on the order of 2^n steps, for large n (Muthumanickam *et al.* 2023)). This empirical calibration complements theoretical proofs, giving confidence that recommended parameters for schemes like Kyber or Dilithium are beyond the reach of both classical and anticipated quantum computers. The major trade-offs of lattice-based cryptography are the increased size of keys and ciphertexts/signatures and the need to treat error terms and probability (e.g., the need to sample from distributions) with care in implementations, which can be the source of side-channel leakage if not treated carefully. However, lattice-based schemes

are now being regarded as front-runners to replace classical public-key cryptography due to their strong security and favorable performance.

Data-driven parameter adjustment for lattice schemes begins with structured benchmarking. Developers generate test data by experimenting with encryption, decryption, signing, and verification time across a grid of lattice dimension (e.g., 512, 768, 1024 bits) and noise-distribution parameter values. Ordinary least squares or ridge regression model how each parameter influences runtime and memory use so that predictions can be made about untested configurations. Cross-validation ensures these models generalize, so groups can choose, for example, a 768-dimension sample of LWE that meets a 1 ms decryption requirement with an estimated 128-bit classical security level (Saki *et al.* 2021).

From the security side, quantum-attack cost estimation is derived from lattice-reduction algorithm simulations (e.g., BKZ) on tiny lattices. When running tens or hundreds of reduction experiments and recording block size, pruning method, and sieve operations required, researchers calibrate generalized additive models that extrapolate to full-scale lattices. Statistical approach enhances conservative security margins—rather than worst-case guesses, designers decide on empirical curves mapping BKZ block size to quantum-cost estimates.

Code-Based Cryptography

Code-based cryptography is one of the earliest techniques for building quantum-resistant algorithms, having been proposed in 1978 by Robert McEliece. It utilizes coding theory problems – specifically, the difficulty of decoding a general linear error-correcting code. In the McEliece cryptosystem, the public key is in fact a generator matrix of a linear error-correcting code (originally a Goppa code) shuffled up by permutations, and the private key is the trapdoor structure which allows efficient decoding (having knowledge of the specific code structure). Encrypting the message is equivalent to adding a random error to an example of this public code codeword; decrypting without the trapdoor is equivalent to decoding a large error from otherwise random linear code, which is conjectured to be infeasible for long code parameters. For over forty years, the McEliece scheme has withstood cryptanalysis, even quantum cryptanalysis; no algorithm significantly faster than brute-force decoding is known for random linear codes of sufficient length (Sreelatha *et al.* 2023).

The main drawback of the classical McEliece scheme is its public key size. A typical McEliece public key (with a security level comparable to RSA-2048) would likely be several hundred kilobytes or even a megabyte in size – orders of magnitude larger than RSA or lattice-based keys. This makes it less convenient for some applications, especially where transferring or storing the same keys is restricted (e.g., in embedded devices or protocols that tend to transfer keys frequently). However, where extensive key distribution is allowed, code-based encryption remains a nice option due to its history of security. The resulting ciphertexts of McEliece encryption are relatively small (of the order of a few hundred bytes) and encryption/decryption times are fast, due to the simple linear algebra and error-correction computations involved.

As an alternative to the key size issue, researchers have explored modifications of code-based schemes that utilize structured codes in an attempt to reduce public key length. An alternative is to use quasi-cyclic codes or other algebraic code structures that allow the public key to be more efficiently stored. Some of the PQC competition proposals (e.g., BIKE or HQC proposals) took this path, offering acceptable key sizes at the cost of sacrificing some known security margins (they rely on specific structure assumptions for the codes, which are well-studied but less time-tested than McEliece's binary Goppa codes) (Easttom 2020). While a number of these coded-structure proposals had promise, they sometimes became a victim of security backsliding (with cryptanalysts uncovering weaknesses or structure-specific decoding attacks). To date, the initial McEliece based on big binary Goppa codes remains a gold standard for code-based security – it was selected as an "alternate" by NIST, reserved for standardization in case lattice-based alternatives have problems or for particular use-cases where its massive keys are acceptable.

Code-based cryptography is used primarily for encryption/KEM (key encapsulation mechanisms); it has not worked as well in the digital signature arena. It has proved difficult to create an actual real-world signature scheme based on coding theory since most attempts resulted in unbearably huge signatures or were susceptible to attacks. Code-based algorithms are thus primarily discussed when it comes to public-key encryption replacement and key transport (where they can be a viable quantum-safe choice if one can tolerate working with key size).

For McEliece and variants, data science workflows typically start with synthetically created attack-simulation datasets. They set varying code length (e.g., 2^{23} to 2^{2} bits), dimension, and error weight, and perform information-set decoding algorithms to obtain success probabilities and average iteration counts (Naik *et al.* 2023). A logistic-regression model derived from this data estimates decoding success for random parameter sets, guiding selection of error weights to maintain success probability below, for instance, 2^{-128}.

Performance enhancement depends on profiling traces from real implementations (C or hardware). A multivariate analysis of variance (ANOVA), by function-level timing across different input patterns traced, will identify which of the code paths—matrix multiply, syndrome calculate, or Gaussian elimination—are responsible for the predominant overall latency. This will decide whether linear-algebra library tuning or the implementation of GPU-based XOR networks will yield the best runtime benefits in embedded environments.

Multivariate Cryptography

Multivariate cryptography relies on the difficulty of solving systems of multivariate polynomial equations over finite fields – a problem that is NP-hard in general. In cryptographic schemes, the typical method is to build a trapdoor function from a system of quadratic equations (that is why it is "multivariate quadratic" or MQ schemes) so that the public key is a system of seemingly random quadratic equations and the private key allows one to solve for the equations easily. The appeal of multivariate schemes, especially for signatures, is that the verification is likely to be extremely fast (merely calculate some polynomials) and the signatures do not have to be extremely long, and the calculations are simple arithmetic in small fields that map very nicely to low-power devices.

An excellent example of multivariate signature schemes is the Unbalanced Oil and Vinegar (UOV) methodology and its variations, of which Rainbow was a notable example (Huang *et al.* 2018). In a UOV scheme, the variables of the polynomial equations are divided into two sets ("oil" and "vinegar"), and a systematic method of generating and solving the equations with

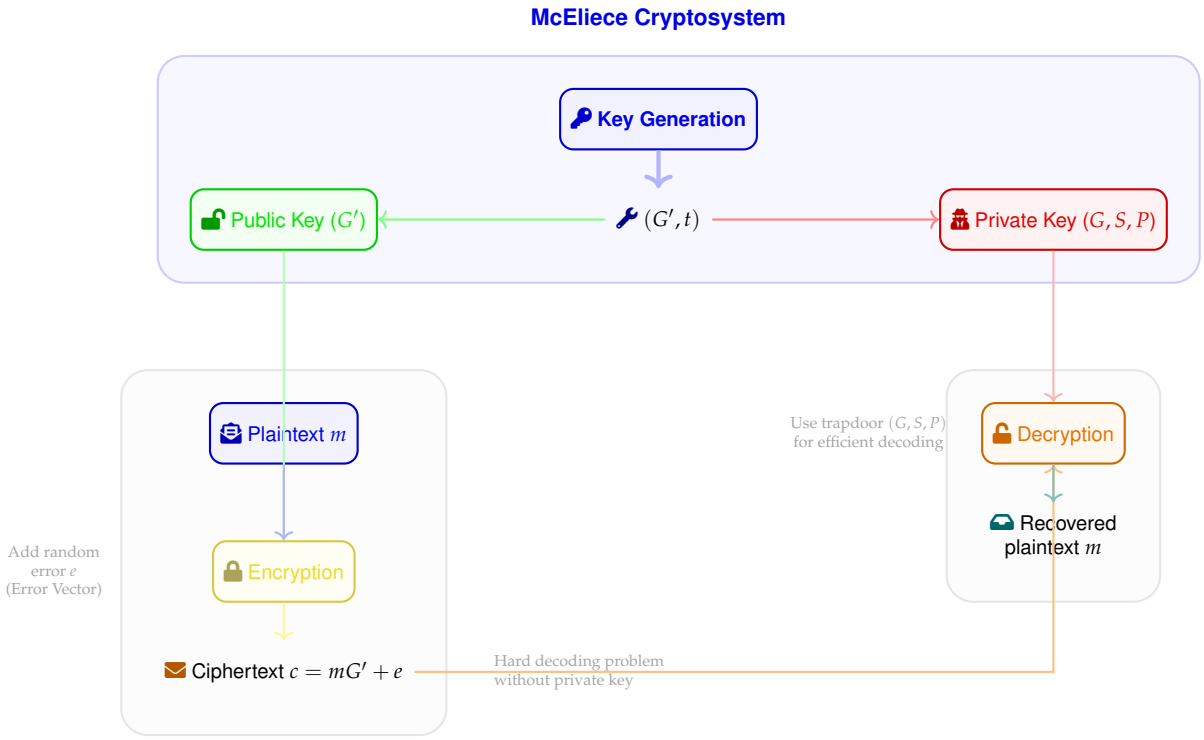

Figure 6 An overview of the McEliece cryptosystem, illustrating key generation, encryption by adding a random error vector, and decryption via the private trapdoor structure for efficient decoding. The security relies on the difficulty of decoding random linear codes.

a secret assignment leads to a signature generation algorithm. Rainbow, a layer generalization of UOV, survived the rounds of the NIST competition and was a top contender due to its small signature sizes and reasonable key sizes. But in 2022 there was a significant cryptanalytic attack that rendered Rainbow de facto broken by solving the underlying equations much better than expected, rendering it ineligible for consideration. This is a common pattern in multivariate cryptography: numerous schemes proposed over the years (such as the popular Sflash and subsequent ones such as TTS, HFE forms, etc.) were later broken or compromised, quite often by algebraic attacks taking advantage of disguised structure in the seemingly random quadratic polynomials.

Multivariate cryptography, though, continues to be an ongoing area of research. Its potential advantages – most notably extremely efficient signature verification and tiny signatures – are attractive for some uses (such as signature verification of lots of signatures on low-power platforms, e.g., smart cards). A number of multivariate schemes (e.g., some UOV variants) are unbroken with some parameter choices, but they may incur trade-offs like very huge public keys or other impracticalities. To date, there has not been a standard multivariate scheme, but efforts continue to make improvements on these designs and set security or find new hard problems for this field (Raju and Kalaiselvi 2023). The problem has been to find a construction that is efficient and resists all known algebraic attack techniques. The fate of Rainbow reminds us that multivariate designs, while theoretically appealing, need more gestation time before it can be fully trusted as a part of the cryptographic toolset in the post-quantum world.

Experiment runs for multivariate designs generate high-dimensional outcome tables: signing time, public-key size, signature size, and resistance estimate to MinRank or Gröbner-basis attack per parameter configuration. Principal component analysis (PCA) then reduces dimension to highlight central trade-off axes—often demonstrating proportionality of halving vinegar variables by 10% in significantly impacting signature size while only marginally impacting resistance estimates. This reveals insight to preclude a large portion of candidate designs from dense algebraic investigation.

In parallel, evidence of algebraic attacks (e.g., field operation count, memory access in solving polynomials) is accumulated. Unsupervised clustering of such evidence can detect parameter sets where attack patterns are unusual, potentially unveiling hidden structure that undermines security (null Chindiyababy *et al.* 2022). Such data-driven alerts induce further cryptanalysis before any new multivariate proposal is adopted.

Hash-Based Cryptography

Hash-based cryptography follows a fundamentally different approach: it relies solely on the assumed collision resistance and preimage resistance of cryptographic hash functions. Cryptographic hash functions like SHA-256 or SHA-3 are used as black-box primitives to construct digital signature schemes. The most widely used hash-based signatures are built on the one-time signature (OTS) principle – schemes that use a set of random values as a private key and publish their hash as a public key, and sign a message by revealing some of the private key values as a function of the message hash. Lamport's one-time signature scheme is a classic example. One-time signatures are highly efficient and secure (their security can be reduced to the security of the hash function directly), but by name, each key can safely sign only

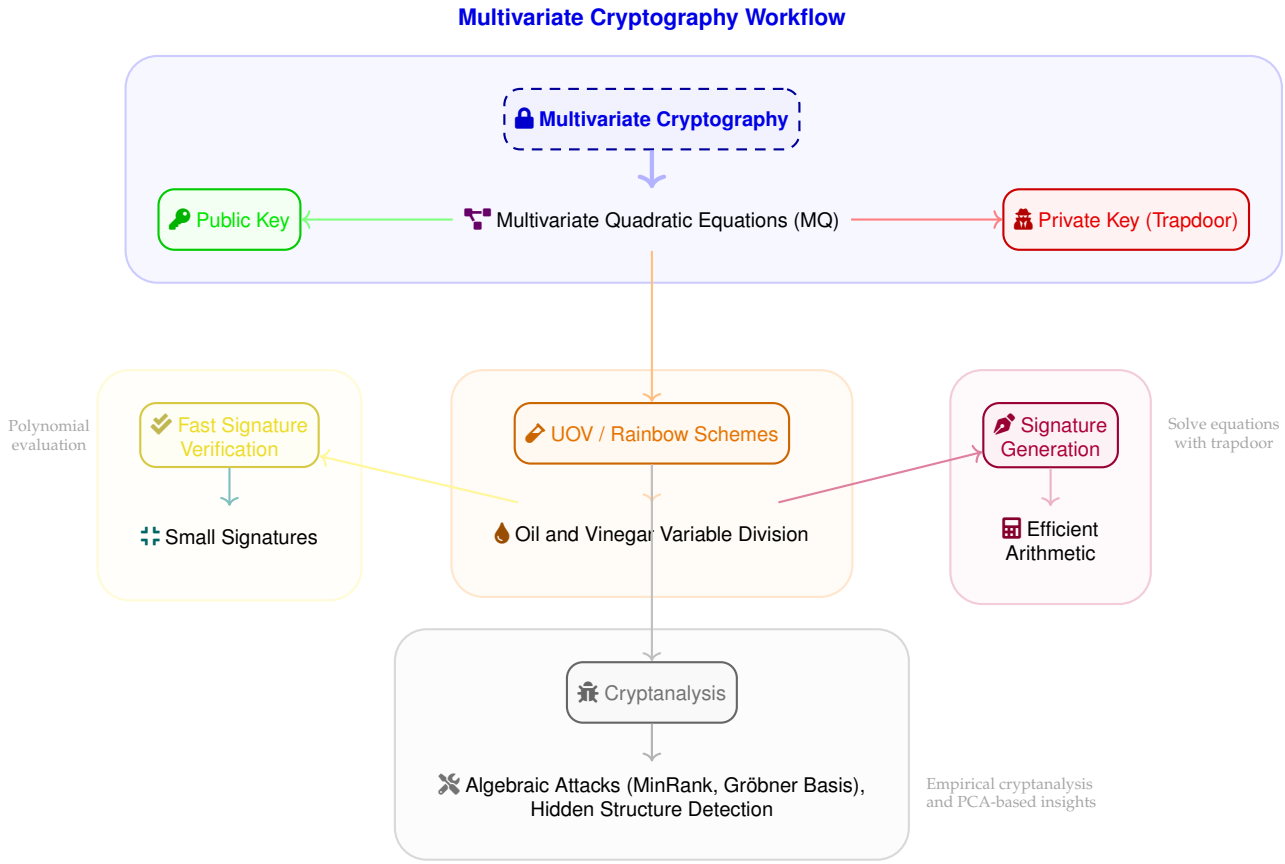

Figure 7 Overview of multivariate cryptography illustrating the MQ problem, the UOV/Rainbow signature generation approach, and signature verification efficiency. It also depicts cryptanalytic attacks leveraging algebraic structures to reveal hidden weaknesses in parameter choices.

a single message. To scale this to many messages, hash-based schemes commit to many one-time public keys simultaneously using a structure (typically a binary Merkle tree). The tree root is a constant public key, and each leaf is used to sign one message (Ghoerad 2022). This is known as the Merkle Signature Scheme.

Recent hash-based signature schemes, including SPHINCS+ (selected to standardize as a backup post-quantum signature), have developed this concept further to be viable and stateless. "Stateless" is the mandate that the signer need not store which one-time keys were exhausted – SPHINCS+ accomplishes this with a sensible tree-of-trees structure in support of in effect unlimited signatures at the price of longer sizes for signatures. A standard SPHINCS+ signature will probably be on the order of tens of kilobytes as it consists of a one-time signature and an avenue down through many Merkle trees. Verification is hashing along such paths, which is slower than verifying an RSA or lattice-based signature but still acceptable (on the order of milliseconds). The public key and private key for hash-based schemes are relatively small (typically a few dozen bytes plus some seeds for pseudo-random number generation).

The overwhelming benefit of hash-based signatures is their strength: their security is based on extremely minimal and well-established assumptions about hash functions. Even in a post-quantum era, when the output size of a hash function may have to be larger to withstand Grover's algorithm (as discussed, doubling the output length would neutralize the square-root speed-up), the design of hash-based signatures remains solid (Zhang *et al.* 2021). In fact, stateful hash-based signature schemes such as XMSS and LMS (which require the signer to keep track on which one-time keys have already been used) have already been standardized for controlled use in individual industries, yet proper handling of state can become error-prone. Hash-based signatures are usually referred to as the "backup plan" – even if all other algebraic post-quantum schemes were somehow compromised, one would resort to hash-based signatures (at the expense of performance and size penalties) due to their inherent security. They have already been standardized in certain regimes (e.g., stateful hash-based signatures such as XMSS and LMS are approved for certain applications by some standards bodies). The price, as said, is space: large signatures and longer verification/signing. This suggests that hash-based signatures are perhaps not suitable for all applications (you wouldn't necessarily want to transmit 40KB signatures over each TLS handshake due to bandwidth, and verification of thousands might be too intense on a low-energy device, e.g.). But for applications like software signing or system updates, where signature size is not quite as big an issue, hash-based signatures are extremely attractive for their simplicity of use and security promise.

It should be observed that hash-based cryptography is generally not employed for key exchange or encryption but is mainly a digital signature solution (Arias *et al.* 2021). Symmetric primitives (e.g., block ciphers and hashes themselves) are, as mentioned earlier, quite safe against quantum attacks with minor changes (longer outputs and keys), so the pressing demand in

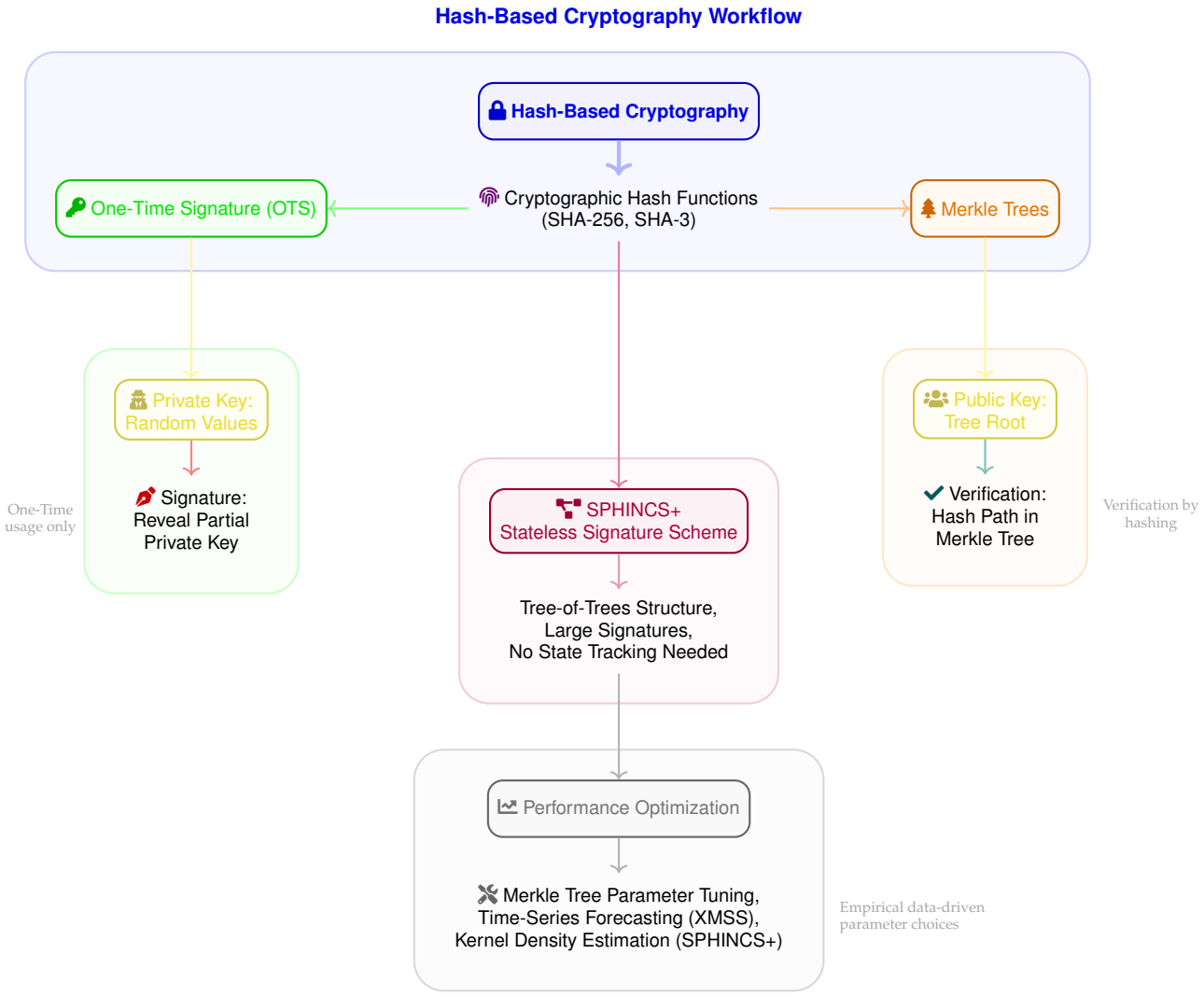

Figure 8 Overview of hash-based cryptography, detailing the use of cryptographic hash functions in one-time signatures, Merkle trees, and SPHINCS+ stateless signatures. Performance optimization involves data-driven parameter tuning and forecasting methods for robust implementation.

PQC comes from asymmetric cryptography. Hash-based signatures satisfy half of that requirement in a very traditional way.

Hash-based signature optimization involves experimenting with Merkle-tree parameters at size. Sets take snapshots of thousands of signing sessions with tree heights of 10 to 20, tracking not only mean signing/verification time but also path-length distributions and memory allocation spikes. Time-series forecasting (e.g., ARIMA models) on one-time key usage in stateful schemes like XMSS provides healthy warnings when remaining leaf nodes drop below a safety threshold, preventing accidental reuse of keys.

In stateless protocols such as SPHINCS+, tree-layer numbers determine signature-size distributions for randomly generated message inputs produced through Monte Carlo simulation. Kernel density estimation subsequently computes a smooth expected signature-size profile, allowing system designers to allocate buffer sizes or storage budgets appropriately for network packets or storage demands. Regression tests also quantify the scaling of performance with tree-layer numbers, in favor of parameter choices that provide trade-offs in signature bulk and realistic throughput (Shah 2022).

Other Approaches and Emerging Schemes

Beyond the large categories mentioned above, researchers have also considered other math problems for post-quantum security. Among them was isogeny-based cryptography, which relies upon the math of elliptic curve isogenies. An isogeny is a specific kind of function (morphism) that maps an elliptic curve to another one. The hard problem used here is to find an isogeny between two given elliptic curves (usually given extra public information about their forms) – an endeavor which, for specially constructed curves, does not appear to have any quantum shortcut. The most widely known isogeny-based scheme was SIKE (Supersingular Isogeny Key Encapsulation), which received high-profile attention since it had extremely small key sizes (public keys consisting of hundreds of bytes, which is much smaller than lattice or code-based keys). SIKE moved on to the next rounds of the PQC competition. But in a dramatic twist of fate, SIKE was found to be compromised by a classical cryptanalysis in 2022 – researchers identified a specific mathematical

attack that was able to recover the secret isogeny much faster than brute force. The attack exploited the particular structure of the problem on which SIKE's security relied, and it made the point that even "post-quantum" cryptosystems can have hidden vulnerabilities (Porambage *et al.* 2023). As a result, isogeny-based cryptography lagged behind; SIKE was no longer on the cards for standardization. There are further isogeny-based proposals (e.g., those utilizing a commutative group operation on isogenies, such as the CSIDH protocol), but they tend to be in an earlier stage of development or are linked to high computational cost.

Another method, also somewhat of interest, uses zero-knowledge proofs and symmetric-key primitives – e.g., the Picnic signature scheme (a PQC competition entry) that used zero-knowledge proofs of possession of a secret key to a block cipher. Picnic used none of number-theoretic problems at all, but just the security of symmetric ciphers and soundness of the zero-knowledge protocol. Despite not winning in competition for Picnic and similar schemes (since performance and size were less competitive when compared to lattice- or hash-based winners), they represent an appropriate way to innovate: building security upon assumptions different from what is conventionally assumed. Such as, maybe, hardness on solving some special combinatorial problems, or soundness on cryptographic proofing. They can become materialized if the faith on alternate assumptions collapses (Fei *et al.* 2018).

In isogeny-based protocols (e.g., CSIDH), researchers benchmark basic operations—curve point multiplications and isogeny walks—across various prime-field sizes. They train random forests that predict end-to-end handshake latency as a function of CPU clock speed and word size, which in turn guides parameter selection to meet sub-10 ms exchange needs. Attack-simulation datasets (measuring success rates of torsion-point attacks) in the meantime power Bayesian risk models that predict confidence in each parameter set over time.

Zero-knowledge–based signatures like Picnic apply data science to proof-circuit optimization. Through the logging of proof-generation and -verification times for different MPC backends and circuit partitionings, groups employ k-means clustering to identify the most optimal circuit decompositions. The clusters inform automated tools that recommend circuit structures likely to optimize the overall proof size and latency for a given block-cipher instantiation.

Practical Challenges in Adopting Post-Quantum Cryptography

The emergence of efficient post-quantum cryptographic schemes is a significant breakthrough in cryptography, but the application of such novel schemes to realistic systems is a gigantic task. There are certain practical issues and problems that must be resolved while the world cryptographic infrastructure transitions from classical to post-quantum schemes (Kumar *et al.* 2023). We identify some of the significant issues and the status quo of activities to fill them in:

One of the primary issues is that most PQC algorithms have longer key and ciphertext sizes than their conventional counterparts. As an example, as noted, lattice-based public keys and signatures are kilobytes in length rather than tens of bytes for ECC. When these larger objects are transmitted in protocols like TLS or are kept in certificates, they can become a bandwidth and storage burden. Early trials that built PQC into protocols have experienced that they suffer from issues such as higher network latency or message splitting of handshakes. For instance, an additional post-quantum key exchange over a TLS handshake can potentially create the handshake above the size boundaries of some middleboxes in a network, and the connections become lost. Real-world experiments by technology companies (e.g., pilot deployments of hybrid post-quantum TLS handshakes by cloud providers and browser manufacturers) have shown that performance overheads are acceptable in the majority of situations, but some legacy network devices find it difficult to handle the increased handshake data, requiring changes to protocol and software patches to address interoperability problems. Optimal encoding and the use of hybrid approaches (transmitting classical and PQ keys but potentially compressing or deduplicating some information) are techniques being investigated to address these issues (Alyami *et al.* 2021). On the computational side, although many PQC operations are fast in software (often faster than RSA signing or key generation), some are more computationally expensive or memory-intensive (e.g., number-theoretic transforms within lattice schemes, or large matrix computation within code schemes). Granting that even constrained devices (microcontrollers, smart cards, IoT sensors) will be able to perform PQC operations in timely manner is a topic of active research. In some cases, hardware acceleration may be required: just as CPUs today include instructions to speed up AES or RSA, hardware tomorrow can offer accelerators for lattice polynomial multiplications or other computationally intensive tasks to enable PQC to run efficiently everywhere.

Merging new algorithms into the vast repository of protocols and standards that currently use RSA/ECC is another challenge. Organizations such as the Internet Engineering Task Force (IETF) are working on extensions to protocols – for example, defining new TLS cipher suites incorporating post-quantum key exchange and new digital signature schemes to be utilized in X.509 certificates. One of the issues here is being compatible. Not everyone will accept new algorithms in a transition phase (Liman and Weber 2023). A web server can have a post-quantum certificate, but an old browser would not be able to understand it, leading to connectivity failure. So as not to disrupt interoperability, there are temporary workarounds like hybrid schemes: a certificate can have two signatures (a classical one and a post-quantum one), or an TLS handshake can negotiate a hybrid key exchange that involves an ECDH key share and a Kyber key share. The security is at least as high as the better of the two; even if one is broken, the other one is still protecting the session. This approach ensures that even if an attacker has a quantum computer, they would still have to compromise the classical part (if the PQC was insecure) or the other way round, thereby protecting both options. But hybrid systems introduce complexity and message size, and require careful standardization so as not to facilitate downgrade attacks (where the attacker might try to trick participants into using just the less secure algorithm).

Organizations and the public trusted RSA and ECC for decades, and it was because of extensive testing and experience in the industry. Post-quantum algorithms are new; entrusting stakeholders with important security responsibilities is a long process. The NIST challenge and such test activity have helped by bringing about open analysis — many teams of cryptanalysts around the world have attempted (and in some instances succeeded) to break candidates, which further reinforces confidence in the survivors (Petrache and Suciu 2020). Yet some degree of caution still remains. Others advise spreading cryptographic use beyond "all eggs in one basket." That is, for example, not

relying on a single family only (only codes or only lattices) to satisfy all requirements, but, e.g., using lattice-based KEMs and a hash-based or code-based signature for especially sensitive systems so that a single breakthrough compromising everything is less likely. This warning is in line with the fact that we are providing security decades in advance with no promises whatsoever. As time goes by and PQC schemes become increasingly well-known, and as they continue to resist attacks, trust will build naturally. In the meantime, conducting pilot projects and small-scale deployments (such as test websites, VPNs or secure messaging apps based on PQC in test mode) has been an unavoidable step in order to demonstrate feasibility and build trust. Actually, by 2025, some technology companies have shown early adoption of post-quantum algorithms in products (e.g., initial deployment in messaging apps and test browser releases), which demonstrates growing confidence in practical readiness.

We have discussed cryptographic agility as a planning requirement; it is also a challenge to implement on existing systems. Most software systems were not designed with a future switch of algorithms in mind (Brijwani *et al.* 2023). Hard-coded presumptions (like a fixed size of a key or an explicit algorithm identifier) may be embedded in code and would fail when a new algorithm emerges. Getting protocols more adaptable usually means introducing more flexible parsing and negotiable algorithm parameters, which can in itself be a tedious endeavor. The effort to make systems flexible is absolutely worth it even independent of the quantum leap: it prepares systems to deal with any future eventual cryptographic change, whether due to quantum progress or new classical attacks. Updating infrastructure to flexibility is an urgent priority in most companies today; that is, modernizing cryptographic libraries, protocols, and even law so that the transition to PQC (or the next crypto) is as smooth as possible. In practice, this might involve putting in place centrally managed cryptography systems where algorithms can be swapped out by configuration, rather than code modification, and where system documentation and security policy do not interfere with applying new cryptographic primitives.

While new algorithms are being deployed at large scale, they must be made as resistant to side-channel attacks as the previous ones, if not more so. Decades of experience have acquired the art of making RSA and ECC deployments resistant (constant-time computation to protect against timing side channels, blinding secrets to defend against power analysis, etc.) (Singh *et al.* 2013). The same degree of scrutiny must be applied to the new post-quantum algorithms. Side-channel weaknesses have already been found in careless implementations of some lattice-based schemes (timing variation in noise sampling or conditional branch patterns depending on secret values, for example). These it must address with the same prudent engineering as previously: constant-time programming, not depending on secret patterns for memory accesses, and introducing suitable protections such as noise in analog emanations. Happily, since most PQC operations are algebraic (such as vector/matrix operations), they can usually be implemented efficiently in streamlined, regular forms amenable to resistance against side channels – but it may affect performance, bringing us back to the question of efficiency. Validation and verification of PQC algorithms in hardware infrastructures (TPMs, smart cards, etc.) for side-channel protection are active areas of research with areas of expertise blending from cryptography, data science (signal processing), and hardware development. The goal is to prevent new attack vectors that could compromise a post-quantum scheme even without a quantum computer (e.g., an attacker with a simple power measurement device shouldn't be able to recover a lattice secret key any more easily than they can an RSA key).

The introduction of new cryptographic algorithms generally entails a key management infrastructure modernization: certificate authorities, key servers, HSMs (Hardware Security Modules), as well as other tools employed in generating, disseminating, and storing keys. Certificate authorities will need to begin issuing certificates including PQC algorithms (and quite possibly dual-algorithm certificates across the transition phase) (Iftikhar *et al.* 2021). This requires certificate format revisions and ensuring that all entities in the chain (servers, clients, browsers) can process them. Significant backup and recovery procedures might need to be updated to support larger keys or other key lifecycle (e.g., hash-based one-time signature systems have unique key usage considerations). Ensuring random number generation is secure is one more facet of key management – some PQC schemes use more random bits (for blinding, for making noise, etc.), so the sources of entropy must be high quality in order to avoid undermining security (a lesson learned from past experience when poor randomness killed cryptography).

Vendors of cryptographic libraries have started to include PQC algorithms (even though labeled as experimental), and industry associations are collaborating on interoperability testing. Experience with past transitions – i.e., the SHA-1 to SHA-256 transition, or the 1024-bit RSA to 2048-bit RSA transition – provides a template for carrying out large-scale cryptographic transitions, although the quantum one is in an even larger scope. It will require coordinated efforts by industry, academia, and government. Data science will be valuable at this phase as well: monitoring the rates of adoption, any falls-off or reverse trends in security, and data-based optimization for performance (Castelvecchi 2018).

Conclusion

Global standards bodies have selected and adopted new algorithms, and organisations have begun testing and rolling them out into real-world applications. The threat of quantum computing has brought the security community to its knees and rebuild the cryptographic underpinnings of the internet age. We have, in this paper, described two principal themes in this activity: the deployment of data science approaches to analyze and manage cryptographic exposures to quantum vulnerabilities within the frame of the quantum threat, and the design of post-quantum schemes meant to supersede those vulnerable primitives. With thorough analysis, we showed how data-driven analytics – from machine learning-based cryptanalysis to cryptographic usage mass-scale auditing – provide actionable intelligence about where our current systems are weakest and under what conditions they will break against quantum-enabled adversaries. Continued analytic vigilance is vital; it allows us to prioritize defense and try and validate the effectiveness of mitigations, rather than have a lack of insight into a developing threat landscape.

Concurrently, post-quantum cryptos coming to maturity are a path to secure security. We have walked through the mainstream types of them, each born out of challenging maths problems that, to date, appear quantum-proof to quantum computing. Lattice-based, code-based encryption, hash-based signatures, and others (multivariate, isogeny-based) altogether constitute a palette out of which secure systems in the future can be built. The diversity of approaches is a strength, with options and fall-

backs in the event of failure of any single assumption. Most importantly, perhaps, the cryptography community's process of standardizing and testing – based on academic peer review, open competition, and vigorous testing – has led to very high confidence in a suite of core algorithms now being put forward for deployment on a large scale.

The interoperability between data science and cryptography that we have described is at the heart of a successful transition to post-quantum. Data science sheds light on the practical realities of deploying new cryptography: measuring performance impacts, finding implementation issues, and ensuring that improved systems behave as intended in practice. Solid cryptographic design provides the secure components that data-driven methods aim to verify and enforce. Together, they enable an anticipatory defense strategy: early detection of threats, their reaction with good solutions, and constant monitoring of the security of those solutions in the long term.

The quantum computing threat, while impressive, is being confronted by an equally impressive response. Through the combination of hard theoretical efforts on post-quantum algorithms and data-driven, empirical study of systems and threats, the information security community is charting a course to protect information decades down the road. Sustained study and global cooperation will be necessary in the future. When staying alert and agile in response to new developments—whether advances in quantum computing or improvements in cryptanalytic techniques—it is possible to keep cryptographic defenses one step ahead of attacks. While this may be the greatest disruption cryptography has experienced in decades, it will be one that brings stronger protections and a cyberinfrastructure secure enough to withstand the test of the quantum era.

References

2023. pp. 270–283 in *Security from Physical Assumptions*. Cambridge University Press.

Ali MZ, Abohmra A, Usman M, Zahid A, Heidari H, Imran MA, Abbasi QH. 2023. Quantum for 6g communication: A perspective. IET Quantum Communication. 4:112–124.

Alyami H, Nadeem M, Alharbi A, Alosaimi W, Ansari MTJ, Pandey D, Kumar R, Khan RA. 2021. The evaluation of software security through quantum computing techniques: A durability perspective. Applied Sciences. 11:11784–11784.

Aoun B, Tarifi M. 2004. Quantum networking.

Arias D, Sanz B, de la Puerta JG, Pastor I, Bringas PG. 2021. pp. 156–166 in *CISIS-ICEUTE - A Repeated Mistake is a Choice: Considering Security Issues and Risks in Quantum Computing from Scratch*. Springer International Publishing.

Badhwar R. 2021. pp. 3–14 in *Are You Ready for Quantum Computing*. Springer International Publishing.

Balakumar A. 2022. Quantum k-means clustering and classical k means clustering for chest pain classification using qiskit. International Journal for Research in Applied Science and Engineering Technology. 10:945–948.

Banafa A. 2023. pp. 131–133 in *Cloud Computing Security*. River Publishers.

Bandi D, Shi Y, Shahriar H, Lo D, Suo K, Chi H, Qian K. 2023. Quantum machine learning for security data analysis. In: . pp. 460–465. IEEE.

Berezin AA. 2007. Quantum computing and security of information systems. In: . volume I. pp. 149–159. WIT Press.

Beri R. 2021. A contemporary study on quantum-computing security mechanisms in 5g networks. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 12:450–455.

Bhushan S. 2022. pp. 193–206 in *Quantum Cryptography*. CRC Press.

Brijwani GN, Ajmire PE, Thawani PV. 2023. pp. 267–298 in *Future of Quantum Computing in Cyber Security*. IGI Global.

Brotherton M, Gupta M. 2023. pp. 1–34 in *A Survey of Quantum Computing for Cloud Security*. CRC Press.

Castelvecchi D. 2018. The quantum internet has arrived (and it hasn't). Nature. 554:289–292.

Das S, Chatterjee A, Ghosh S. 2023. A first order survey of quantum supply dynamics and threat landscapes.

Das S, Ghosh S. 2023. Trojan taxonomy in quantum computing.

Easttom W. 2020. pp. 385–390 in *Quantum Computing and Cryptography*. Springer International Publishing.

Fei Y, Meng X, Gao M, Wang H, Ma Z. 2018. Quantum man-in-the-middle attack on the calibration process of quantum key distribution. Scientific reports. 8:4283–4283.

Ghoerad N. 2022. The future's most advanced computing tool: Quantum computers. Journal of Research in Science and Engineering. 4.

Gupta KD, Nag AK, Rahman L, Mahmud MAP, Sadman N. 2021. Utilizing computational complexity to protect cryptocurrency against quantum threats: A review. IT Professional. 23:50–55.

Gupta S. 2017. Quantum fog cloud model in internet of things with analysis of green computing. International Journal of Research -GRANTHAALAYAH. 5:230–238.

Huang A, Barz S, Andersson E, Makarov V. 2018. Implementation vulnerabilities in general quantum cryptography. New Journal of Physics. 20:103016–.

Iftikhar Z, Iftikhar M, Shah MA. 2021. Icac - quantum safe cloud computing using hash-based digital signatures. In: . pp. 1–6. IEEE.

Khan S, Jain C, Rathi S, Maravi PK, Jhapate A, Joshi D. 2023. Quantum computing in data security.

Khodaiemehr H, Bagheri K, Feng C. 2023a. Navigating the quantum computing threat landscape for blockchains: A comprehensive survey.

Khodaiemehr H, Bagheri K, Feng C. 2023b. Navigating the quantum computing threat landscape for blockchains: A comprehensive survey.

Kumar B, Topno V, Banerjee P, Jha P. 2023. Revolutionizing national security: An analysis of quantum computing's impact on government and defense. In: . pp. 1–9. IEEE.

Liman A, Weber K. 2023. Quantum computing: Bridging the national security–digital sovereignty divide. European Journal of Risk Regulation. 14:476–483.

Liu J, Ren T. 2023. Retracted article: Economic impact of quantum sports technologies on healthcare artificial intelligence based study. Optical and Quantum Electronics. 56.

Liu Z, Xie Q, Zha Y, Dong Y. 2022. Quantum delegated computing ciphertext retrieval scheme. Journal of Applied Physics. 131.

Lyssenko D, Komolafe O. 2023. Leveraging quantum key distribution for securing macsec communications. In: . pp. 57–61. ACM.

Miller JL. 2012a. A blind quantum computer makes its laboratory debut. Physics Today. 65:21–22.

Miller JL. 2012b. A blind quantum computer's laboratory debut. Physics Today. 2012.

Mone G. 2020. The quantum threat. Communications of the ACM. 63:12–14.
Muthumanickam K, Mahesh PCS, Ragab M. 2023. pp. 84–102 in *A Review on Quantum Computing and Security*. IGI Global.
Naik A, Yeniaras E, Hellstern G, Prasad G, Vishwakarma SKLP. 2023. From portfolio optimization to quantum blockchain and security: A systematic review of quantum computing in finance.
null Chindiyababy, Jayaraman R, Kumar M. 2022. pp. 179–192 in *Quantum Cryptography and Quantum Key Distribution*. CRC Press.
Pandya M. 2015. Securing cloud - the quantum way.
Petrache AL, Suciu G. 2020. Security in quantum computing. Annals of Disaster Risk Sciences. 3:0–0.
Porambage P, Liyanage M, Halunen K, Nikula S. 2023. Quantum security and postquantum cryptography.
Raavi M, Wuthier S, Chandramouli P, Balytskyi Y, Zhou X, Chang SY. 2021. pp. 424–447 in *ACNS (2) - Security Comparisons and Performance Analyses of Post-quantum Signature Algorithms*. Springer International Publishing.
Raina P, Kaushal S. 2019. pp. 295–306 in *A Framework for Security Management in Cloud Based on Quantum Cryptography*. Springer Singapore.
Raju R, Kalaiselvi N. 2023. pp. 81–92 in *Optimal Cryptographic Approach Based on Secure Post-Quantum Cryptographic Algorithms and Classical Algorithms*. IGI Global.
Ravi P, Chattopadhyay A, Bhasin S. 2022. Security and quantum computing: An overview. In: . pp. 1–6. IEEE.
Reisner JA, Donkor E. 2000. Protecting software agents from malicious hosts using quantum computing. In: . volume 4047. pp. 50–57. SPIE.
Saki AA, Alam M, Phalak K, Suresh A, Topaloglu RO, Ghosh S. 2021. A survey and tutorial on security and resilience of quantum computing.
Schindler P. 2022. The outlook of german companies on it-security and their readiness for quantum computing. In: . Acavent.
Shah R. 2022. The conventional security of cloud computing and the growing threat to quantum computing. In: . pp. 1–7. IEEE.
Sharma N, Ramachandran RK. 2021. The emerging trends of quantum computing towards data security and key management. Archives of Computational Methods in Engineering. 28:5021–5034.
Singh H, Gupta DL, Singh AK. 2013. Entropy security in quantum cryptography. International Journal of Computer Applications. 81:19–24.
Sreelatha P, Purohit N, Sekaran SC. 2023. pp. 219–230 in *16 RSA security implementation in quantum computing for a higher resilience*. De Gruyter.
Sun S, Huang A. 2022. A review of security evaluation of practical quantum key distribution system. Entropy (Basel, Switzerland). 24:260–260.
Tan J, Xiao L, Qiu D, Luo L, Mateus P. 2022. Distributed quantum algorithm for simon's problem. Physical Review A. 106.
Wang H, Lai X. 2023. Quantum information technology for network security: An application-focused overview. In: . pp. 881–886. IEEE.
Zhang Y, Coles PJ, Winick A, Lin J, Lütkenhaus N. 2021. Security proof of practical quantum key distribution with detection-efficiency mismatch. Physical Review Research. 3:013076–.