WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Cryptographic Discovery and Post-Quantum Migration Risk Assessment: A Review of Approaches and Open Gaps

Aravala Satvik, Avula Yaswanth Sai Charan Reddy, Allamsetti Rohith, Annepu Laxmi Prasanna Rao

GMRIT Deemed to be University

satvikaravala@gmail.com, yaswanthavula879@gmail.com, rohith.allamsetti@gmail.com, a.lprasannarao@gmail.com

ABSTRACT

Moving to post-quantum cryptography (PQC) is hard for organizations that still rely on RSA and elliptic-curve cryptography (ECC), both of which a sufficiently capable quantum computer could break with Shor's algorithm. The Harvest Now, Decrypt Later (HNDL) threat adds urgency, because data collected today may still need protection when such computers arrive. Planning a migration requires knowing where cryptography is used, how exposed it is, and what to replace first. This paper reviews research that addresses these needs, organized into five areas: hybrid and quantum-resistant protocols, cryptographic discovery and software analysis, cryptographic inventories and the Cryptographic Bill of Materials (CBOM), quantum threat and risk assessment, and PQC migration frameworks and standards. We summarize what each area has achieved, compare the reported results side by side, and point out where the pieces do not yet fit together. The main gap we find is the lack of a single, evidence-based workflow that connects discovery, inventory, risk scoring, and migration planning. We outline a conceptual reference pipeline for such a workflow and discuss open challenges, including incomplete discovery, evidence quality, score validation, and evaluation methodology.

Keywords: *Post-Quantum Cryptography (PQC), Harvest Now, Decrypt Later (HNDL), Application Security Posture Management (ASPM), Cryptographic Bill of Materials (CBOM), Quantum Risk Assessment, Cryptographic Discovery, Literature Review.*

INTRODUCTION

Quantum computing is advancing quickly, and it poses a long-term risk to much of the cryptography in use today. Public-key algorithms such as Rivest–Shamir–Adleman (RSA) and elliptic-curve cryptography (ECC) rest on mathematical problems that a sufficiently capable, fault-tolerant quantum computer could solve efficiently with Shor's algorithm. Any organization that relies on these algorithms for secure communication, authentication, digital signatures, or key establishment therefore needs a plan for moving to post-quantum cryptography (PQC). The National Institute of Standards and Technology (NIST) has standardized three main PQC algorithms: ML-KEM for key establishment, ML-DSA for digital signatures, and SLH-DSA for hash-based digital signatures [21].

A second concern is Harvest Now, Decrypt Later (HNDL), where an adversary stores encrypted data today and waits until quantum computers are powerful enough to decrypt it. Defending against it means knowing which cryptographic mechanisms protect long-lived information, how exposed they are to quantum attacks, and which ones to migrate first. All of that depends on having an accurate picture of the cryptographic assets across an organization's infrastructure. Getting that picture is not easy. Cryptography shows up in network services, web applications, source code, software dependencies, configuration files, digital certificates, and executables. Finding it by hand is slow, and it is easy to miss assets that are embedded, called indirectly, or configured outside the main application code. Knowing the algorithm is also not enough to judge the risk, which can depend on the algorithm's role, its configuration, key characteristics, deployment context, the sensitivity of the data it protects, and what depends on it. Machine-readable inventories such as the Cryptographic Bill of Materials (CBOM) defined by the CycloneDX specification [22] give discovery results a common format, but they do not solve the discovery problem on their own.

Research on these topics is spread across several communities. Some work focuses on protocols, some on software analysis, some on inventory formats, some on risk models, and some on organizational migration planning. This paper brings these strands together. Instead of comparing individual algorithms, it asks a practical question: how does an organization find its cryptography, assess its quantum exposure, and decide what to migrate and when? The main contributions of this review are as follows:

1. Structured overview: A review of 20 works and guidance documents, grouped into five areas that follow the path from protocols to discovery, inventory, risk assessment, and migration.

2. Side-by-side comparison: A summary table of the reviewed works that lists each one's focus, contribution, and reported results or scope.
3. Gap analysis: An account of why discovery, inventory, risk assessment, and migration planning are still treated as separate problems.
4. Reference pipeline and agenda: A conceptual workflow that links these stages, together with open challenges and directions for future work.

BACKGROUND

A. Quantum Threat to Cryptographic Primitives

Public-key mechanisms based on integer factorization or discrete logarithms, including RSA and conventional ECC, are vulnerable to sufficiently capable quantum computers running Shor's algorithm. Symmetric encryption and hash-based mechanisms are different, because quantum attacks affect them differently. Grover's algorithm, for example, affects the search complexity of certain symmetric-key and hash-related security tasks, but this does not mean all such mechanisms are broken in the way RSA and ECC are. The standardized PQC algorithms ML-KEM, ML-DSA, and SLH-DSA [21] are the main replacements for vulnerable key-establishment and signature mechanisms. Detecting a PQC algorithm in a system does not, by itself, prove that the whole implementation is secure.

B. HNDL and Migration Urgency

HNDL matters most for information that must stay confidential for years, such as sensitive organizational records and communications. A common way to reason about urgency, often called Mosca-style reasoning, compares how long the information must stay confidential, the estimated time until relevant quantum capabilities arrive, and the time needed to finish migrating. The result depends on the assumptions behind it. Unknown confidentiality lifetimes, uncertain quantum timelines, and missing migration estimates all have to be handled explicitly, so HNDL outputs should be read as risk estimates, not precise predictions of future quantum capability.

SCOPE AND ORGANIZATION OF THE REVIEW

This is a narrative review. It covers 20 works and guidance documents chosen because each addresses at least one stage on the path from discovering cryptography to planning its migration. The reviewed material includes research papers, implementation studies, surveys, and guidance from NIST and CISA. We group it into five areas: hybrid and quantum-resistant protocols, cryptographic discovery and software analysis, cryptographic inventory and CBOM, quantum threat and risk assessment, and PQC migration frameworks and standards. The next section reviews each area in turn. After that, a comparison table puts the works side by side, a gap analysis explains what is missing, and a conceptual reference pipeline shows how the stages could be connected. The paper closes with open challenges, future directions, and a conclusion.

REVIEW OF EXISTING WORK

A. Quantum-Resistant Protocols and Hybrid Cryptography

Work on quantum-resistant communication has mostly looked at combining classical cryptography with post-quantum mechanisms, so that security holds up during the transition to PQC. Rubio García et al. [1] proposed a triple-hybrid design that combines classical cryptography, ML-KEM-1024, and quantum key distribution (QKD) inside TLS 1.3 and IKEv2. They report about 57 ms of extra TLS handshake overhead, a QKD secret-key generation rate of roughly 2.5 kbit/s, and an IKEv2 execution time of about 33.96 ms for the triple-hybrid setup. The results show that several key-establishment mechanisms can be combined, though the approach needs specialized QKD infrastructure.

Kwon et al. [2] studied compact hybrid digital signatures that combine ECDSA P-256 and Falcon-512. Their construction produces smaller signatures than simple concatenation while still allowing selective verification and backward compatibility. Ricci et al. [14] built a hybrid key combiner that brings together classical key exchange, a post-quantum key encapsulation mechanism, and QKD; on their FPGA implementation it reached up to 1,624 keys/s without QKD and 9.2 keys/s with it. Buruaga et al. [16] showed hybrid quantum-safe TLS working in a software-defined networking environment, and Gupta and Mittal [15] proposed a staged hybrid migration strategy for legacy SCADA environments.

Taken together, these studies show that hybrid deployment is feasible and what it costs. Their focus, though, is protocol design, key establishment, implementation performance, or migration architecture. Discovering cryptographic assets automatically across mixed organizational environments is not what they set out to do, which means an organization still has to find its existing deployments before it can choose among these mechanisms.

B. Cryptographic Discovery and Software Analysis

Automated cryptographic discovery has to come first if quantum-vulnerable assets are to be found. The Cybersecurity and Infrastructure Security Agency (CISA) [19] describes a strategy for automated discovery and inventory across networks, file systems, databases, and software packages. It stresses collecting data on algorithms, cryptographic services, key lengths, software packages, and operating systems. It also admits that automated tools can miss embedded cryptography, customized applications, and some operating-system implementations, so manual or vendor-assisted collection is sometimes needed.

Rattanavipanon et al. [3] tackled the problem of finding quantum-vulnerable software executables with their QED toolchain, which combines file-level dependency analysis, API-level analysis, and static call-trace analysis. They report 100% accuracy on a synthetic dataset, an average analysis time under four seconds for real-world executables, and a cut of more than 90% in analyst workload. Static analysis can still struggle with indirect calls, function pointers, embedded cryptography, and statically linked libraries.

These approaches show the value of automated, evidence-based discovery. But network-level configurations, source-code findings, binary dependencies, certificates, and runtime observations each show a different part of an organization's cryptographic posture. What is missing is a single workflow that brings these mixed findings together, keeps track of where each came from, and removes duplicate or inconsistent asset records.

C. Cryptographic Inventory and CBOM

Cryptographic Bills of Materials (CBOMs) give a structured way to describe cryptographic assets and how they relate. Hess [20] examined the CycloneDX CBOM model and tested inventory generation on the Keycloak source code and an nginx container image. The study found 27 algorithms and 37 dependency edges in Keycloak and, after deduplication, 314 unique assets and 2,700 dependency edges in the nginx image. It also notes the limits of static analysis: false positives, false negatives, dynamically loaded components, and cryptographic operations hidden behind abstractions.

CISA's strategy [19] likewise says automated collection can help build a cryptographic inventory and plan migration, but may not give complete visibility by itself. Both works underline why machine-readable inventories, dependency relationships, and evidence for each discovered component matter.

A good inventory format does not, on its own, guarantee that discovery is complete or tell you how risky each component is. Its value depends on the quality of the evidence collected, how consistently algorithms are named, and how much context is available**.** CBOM is thereforbest seen as one stage in a larger discovery and assessment pipeline, not as a standalone output.

D. Quantum Threat and Risk Assessment

Several studies look at what quantum computing means for security across infrastructure, protocols, and organizations. Baseri et al. [4] examined quantum-related threats across applications, data, runtime environments, middleware, operating systems, virtualization, hardware, storage, and networks. They point out that PQC migration brings its own difficulties, including larger cryptographic artifacts, network fragmentation, implementation complexity, and performance overhead.

Baseri et al. [5] turned to network protocols such as TLS, IPsec, SSH, DNSSEC, and PGP, discussing which classical mechanisms are quantum-vulnerable, the hybrid alternatives, interoperability, and the extra communication overhead of larger post-quantum messages. In another paper, Baseri et al. [11] proposed a risk-assessment framework that covers algorithms, certificates, protocols, and migration stages, using STRIDE-based threat analysis and likelihood-impact reasoning.

Kagai et al. [8] treated HNDL as a risk that unfolds over time. Their model relates how long data must stay confidential, the assumed time until an adversary has quantum capability, and the time needed to migrate, in order to estimate residual exposure. Their analysis illustrates that data with a long confidentiality lifetime can remain exposed even after migration has started. Separately, Chethana et al. [6] reviewed block ciphers and discussed what quantum attacks, including Grover's and Simon's algorithms, mean for symmetric cryptographic design.

These studies give theoretical models and assessment methods for understanding quantum threats. Applying them in practice, however, requires reliable information about the cryptography that is actually deployed, the assets it protects, and the context it runs in. A risk model is only as good as the inventory that feeds it.

E. PQC Migration Frameworks and Standards

PQC migration is about more than swapping out individual algorithms. NIST's transition guidance [17] identifies quantum-vulnerable public-key mechanisms and gives a basis for moving to standardized post-quantum alternatives. NIST's key-generation guidance [18] complements it by covering key-generation methods, randomness, security strength, and key management.

Birgin and Celiktas [10] proposed a sector-agnostic operational framework linking discovery, classification, risk assessment, algorithm selection, deployment, monitoring, and governance, with emphasis on automation, auditability, and integration with continuous integration and continuous delivery (CI/CD) processes. El Bizri et al. [13] compared institutional migration strategies and found common priorities: phased implementation, hybrid cryptography, backward compatibility, and crypto-agility.

Other studies deal with particular deployment needs. Wang and Ismail [7] reviewed PQC applications and migration trade-offs in areas such as finance, blockchain, IoT, and networking. González de la Torre et al. [9] investigated a wireless-based Kyber adaptation for resource-constrained devices. Olushola and Meenakshi [12] implemented an authenticated post-quantum session protocol that combines ML-KEM, ML-DSA, key derivation, and symmetric authenticated encryption.These works supply standards, operational guidance, and implementation-specific migration approaches. Still, their requirements have to be applied to an organization's real cryptographic assets and deployment constraints, and that step depends on knowing what those assets are.

COMPARATIVE SUMMARY

Table 1 lists the reviewed works with their main contribution and the results or scope they report. Two patterns stand out. First, almost every work is strong on one stage of the transition and silent on the others: the protocol studies do not discover anything, the discovery tools do not score risk, and the risk models do not say how their inputs are collected. Second, the reported figures cannot be compared directly, because each study measures something different. Handshake latency, key-generation rates, detection accuracy, and asset counts answer different questions, and the works use different test setups.

Table 1. Summary of the reviewed works.

Ref.	Area	Main contribution	Reported result or scope
[1] Rubio García et al.	Hybrid protocols	Triple-hybrid of classical cryptography, ML-KEM-1024, and QKD in TLS 1.3 and IKEv2	About 57 ms extra TLS handshake overhead; QKD key rate about 2.5 kbit/s; IKEv2 time about 33.96 ms
[2] Kwon et al.	Hybrid signatures	Compact hybrid signatures combining ECDSA P-256 and Falcon-512	Smaller than simple concatenation; selective verification; backward compatible
[14] Ricci et al.	Hybrid key combiner	Combines classical key exchange, a post-quantum KEM, and QKD	Up to 1,624 keys/s without QKD and 9.2 keys/s with QKD (FPGA)
[16] Buruaga et al.	Hybrid TLS	Hybrid quantum-safe TLS in a software-defined networking environment	Integration demonstrated
[15] Gupta and Mittal	Migration strategy	Staged hybrid migration for legacy SCADA environments	Strategy proposal
[19] CISA	Discovery strategy	Automated discovery and inventory across networks, file systems, databases, and software packages	Notes that tools may miss embedded cryptography and custom applications
[3] Rattanavipanon et al.	Executable analysis	QED toolchain: file-level dependency, API-level, and static call-trace analysis	100% accuracy on a synthetic dataset; under 4 s average analysis time; over 90% less analyst workload
[20] Hess	CBOM	CycloneDX CBOM inventory generation on Keycloak source code and an nginx container image	Keycloak: 27 algorithms, 37 dependency edges. nginx (after deduplication): 314 unique assets, 2,700 edges
[4] Baseri et al.	Threat analysis	Quantum threats across applications, data, runtime, middleware, operating systems, virtualization, hardware, storage, and networks	Highlights larger artifacts, network fragmentation, implementation complexity, and overhead
[5] Baseri et al.	Protocol risk	Quantum risks in TLS, IPsec, SSH, DNSSEC, and PGP	Discusses hybrid alternatives, interoperability, and communication overhead
[11] Baseri et al.	Risk framework	Risk assessment across algorithms, certificates, protocols, and migration stages using STRIDE and likelihood-impact reasoning	Framework

Ref.	Area	Main contribution	Reported result or scope
[8] Kagai et al.	HNDL	Temporal model relating confidentiality lifetime, quantum horizon, and migration time	Long-lived data can stay exposed even after migration begins
[6] Chethana et al.	Symmetric ciphers	Review of block ciphers under quantum attacks, including Grover's and Simon's algorithms	Review
[17] NIST	Standards	Transition guidance identifying quantum-vulnerable public-key mechanisms	Guidance
[18] NIST	Standards	Key-generation guidance: methods, randomness, security strength, key management	Guidance
[10] Birgin and Celiktas	Operational framework	Sector-agnostic framework from discovery to governance, with CI/CD integration	Framework
[13] El Bizri et al.	Strategy comparison	Comparison of institutional migration strategies	Common priorities: phased rollout, hybrid cryptography, backward compatibility, crypto-agility
[7] Wang and Ismail	Application review	PQC applications and migration trade-offs in finance, blockchain, IoT, and networking	Review
[9] González de la Torre et al.	Constrained devices	Wireless-based Kyber adaptation for resource-constrained devices	Implementation study
[12] Olushola and Meenakshi	Protocol implementation	Authenticated session protocol using ML-KEM, ML-DSA, key derivation, and symmetric authenticated encryption	Implementation

RESEARCH GAPS

The literature shows solid progress in hybrid cryptography, quantum-resistant protocols, executable analysis, cryptographic inventory generation, quantum-risk assessment, and organizational migration planning. Each line of work, though, addresses a different stage of the PQC transition. Protocol studies mostly evaluate secure communication mechanisms, software-analysis tools identify cryptographic dependencies, CBOM research deals with how to represent an inventory, and risk frameworks model quantum-related exposure.

The real challenge is joining these pieces into one consistent, evidence-driven workflow. Cryptographic findings come from network services, certificates, source code, binaries, configuration files, and software dependencies, and each source has its own visibility and confidence. Without normalization and contextual relationships, the findings are a weak basis for assessing quantum risk asset by asset. Risk models and migration frameworks, in turn, need accurate inventory data before they can prioritize anything usefully.

A CONCEPTUAL REFERENCE PIPELINE

To make the gap concrete, this section outlines a conceptual pipeline that connects the stages discussed above. It is a synthesis drawn from the reviewed work, not an implemented or evaluated system, and each stage corresponds to a problem the literature has already identified.

1. Target collection and surface reconnaissance. The pipeline starts from authorized targets such as IP addresses, domains, subdomains, URLs, network services, repositories, executables, configuration files, and dependencies. Reconnaissance expands this set, and validation and deduplication reduce redundant work. The reach of discovery depends on permissions, network accessibility, and the artifacts available.

2. Dual-track discovery. An external track examines what reachable services expose: TLS versions, cipher suites, certificate information, public-key characteristics, and signature algorithms. Services that cannot be reached or observed are recorded as limitations, not assumed to be secure. An internal track examines source code, binaries, configuration files, package manifests, dependency metadata, and certificates for cryptographic APIs, algorithm identifiers, insecure configurations, hardcoded secrets, and vulnerable dependencies. Because static analysis has limits, a missing finding is not treated as proof that cryptography is absent.

3. Normalization and evidence. Findings from both tracks may describe the same algorithm under different names or parameter sets. Normalization maps these to one data model while keeping the originally observed name; for example, aliases of the ML-KEM-768 parameter set can share one identifier. Each asset keeps its evidence and provenance, such as host and port, file path, code location, detection method, timestamp, and confidence. This separates directly observed configurations from inferred dependencies and pattern-based detections, and ambiguous findings keep their uncertainty instead of receiving an unsupported classification.

4. Inventory and dependency analysis. Normalized assets are consolidated into a CycloneDX CBOM that covers algorithms, protocols, certificates, cryptographic materials, libraries, and their links to applications and services. A dependency graph, built only from relationships the evidence supports, ties an algorithm-level finding to the application or service that would need remediation. The CBOM represents what was found within the defined scope; it does not guarantee complete coverage.

5. Quantum-vulnerability and HNDL assessment. Each asset is classified using its algorithm, parameters, role, and configuration context. Integer-factorization and discrete-logarithm mechanisms are marked as vulnerable to Shor's algorithm, symmetric and hash-based mechanisms are analyzed separately, and standardized PQC mechanisms are identified on their own. Where enough information exists, data sensitivity, confidentiality lifetime, exposure, and migration requirements feed a Mosca-style estimate of urgency.

6. Readiness scoring and prioritization. A composite readiness score can summarize preparedness and support comparison across assets or environments. Risk categories such as critical, high, medium, and low are kept separate from the score, so that one aggregate number cannot hide a highly exposed asset or a critical dependency.

7. Recommendation and migration planning. The pipeline uses each asset's role and the applicable standards to suggest candidates. RSA-based or elliptic-curve key establishment may move to ML-KEM or a carefully designed hybrid mechanism, and classical signatures may be assessed for replacement with ML-DSA or SLH-DSA. The choice depends on protocol support, interoperability, security requirements, implementation availability, and organizational standards. The resulting roadmap can separate assets needing immediate investigation, candidates for staged migration, and assets needing compatibility testing or more evidence. These outputs are decision support and need implementation-specific validation before deployment .

OPEN CHALLENGES AND FUTURE DIRECTIONS

A. Incomplete Discovery

Every discovery approach in the reviewed work has blind spots. Indirect function calls, dynamically loaded libraries, custom implementations, obfuscated code, runtime-generated configurations, embedded cryptography, and statically linked libraries can all hide cryptographic use. External scanning cannot see what is not exposed. Combining several sources helps, but the result should still be reported as the cryptography found within a defined scope, not as a complete picture.

B. Evidence Quality and Provenance

An inventory is only as trustworthy as the evidence behind each entry. Recording where a finding came from, how it was detected, and how confident the detector was lets analysts tell observed configurations from inferred ones. Work is still needed on how to combine conflicting evidence from different sources without discarding any of it.

C. Defining and Validating Readiness Scores

A readiness score has to specify its input variables, normalization rules, weighting scheme, aggregation method, and interpretation. Candidate inputs include the proportion of identified quantum-vulnerable assets, inventory coverage, evidence confidence, PQC adoption, dependency information, and migration progress, as long as they are actually measured justified. A scoring model and its classification thresholds should be tested on defined cases before any claim about their effectiveness is made.

D. Uncertainty in HNDL and Migration Estimates

Confidentiality lifetimes are often unknown, quantum timelines are uncertain, and migration effort is hard to estimate. Future work should make these uncertainties explicit in the outputs instead of hiding them behind a single figure.

E. Evaluation Methodology

A fair evaluation needs a controlled test environment with representative network services and software artifacts whose cryptographic characteristics are known. The reference inventory should be built independently of the tool's output so that detected findings can be compared with the expected assets. Useful measures include detection precision and recall, inventory coverage, duplicate findings, false positives, false negatives, processing time, and the correctness of risk classifications. Tests should also compare external scanning, internal analysis, and the two combined, to show whether combining the tracks improves coverage. Recommendations and scores should be checked against predefined cases with documented expected outcomes. Shared test setups and common metrics would also make results from different studies comparable, which is currently difficult.

F. Integration and Crypto-Agility

Operational frameworks such as the one by Birgin and Celiktas [10] point toward continuous discovery and assessment built into CI/CD processes. Together with the emphasis on crypto-agility in migration strategies [13], this suggests that inventories should be kept current as systems change, not produced once as a snapshot.

CONCLUSION

Moving to post-quantum cryptography is not only a matter of choosing new algorithms. Organizations first have to find where their cryptography lives, judge how exposed each piece is, and decide what to migrate first. The literature reviewed here has made real progress on hybrid protocols, executable analysis, CBOM generation, threat and risk models, and migration frameworks and standards. Each of these areas, however, covers only one stage, and the reported results are hard to compare because the studies measure different things.

The clearest gap is a connected, evidence-driven workflow from discovery to migration planning. We outlined a conceptual pipeline for this and listed the open challenges: incomplete discovery, evidence quality, validated scoring, uncertainty in HNDL estimates, and shared evaluation methods. Addressing them would give organizations a more reliable basis for prioritizing their PQC migration and give researchers results they can compare.

REFERENCES

Complete citation details (authors, title, venue, year, DOI) still need to be added for each entry below. The list follows the numbering used in the text.

[1] Rubio García et al. Triple-hybrid key establishment (classical, ML-KEM-1024, QKD) in TLS 1.3 and IKEv2.

[2] Kwon et al. Compact hybrid digital signatures combining ECDSA P-256 and Falcon-512.

[3] Rattanavipanon et al. QED toolchain for identifying quantum-vulnerable software executables.

[4] Baseri et al. Quantum-related threats across infrastructure layers and PQC migration considerations.

[5] Baseri et al. Quantum-security risks across network protocols (TLS, IPsec, SSH, DNSSEC, PGP).

[6] Chethana et al. Review of block ciphers under quantum attacks.

[7] Wang and Ismail. Review of PQC applications and migration trade-offs.

[8] Kagai et al. Harvest Now, Decrypt Later as a temporal security risk.

[9] González de la Torre et al. Wireless-based Kyber adaptation for resource-constrained devices.

[10] Birgin and Celiktas. Sector-agnostic operational framework for PQC migration.

[11] Baseri et al. Risk-assessment framework using STRIDE-based threat analysis.

[12] Olushola and Meenakshi. Authenticated post-quantum session protocol.

[13] El Bizri et al. Comparison of institutional PQC migration strategies.

[14] Ricci et al. Hybrid key combiner with classical key exchange, post-quantum KEM, and QKD.

[15] Gupta and Mittal. Staged hybrid migration strategy for legacy SCADA environments.

[16] Buruaga et al. Hybrid quantum-safe TLS in a software-defined networking environment.

[17] NIST. Transition guidance for quantum-vulnerable public-key mechanisms.

[18] NIST. Guidance on key generation.

[19] CISA. Strategy for automated cryptographic discovery and inventory.

[20] Hess. CycloneDX CBOM evaluation on Keycloak and an nginx container image.

[21] NIST. FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA). New entry: replaces the Introduction's earlier [1].

[22] CycloneDX. Cryptographic Bill of Materials (CBOM) specification. New entry: replaces the Introduction's earlier [2].