Quantum Software Security

Anshu Kumari Pandey[1], and Mohd Nadeem[2]

[1]Department of Computer Science, Institute of Technology and Management, Gorakhpur, India

[2]Department of Computer Science and Engineering, Shri Ramswaroop Memorial University, Barabanki, India

DOI: https://doi.org/10.47772/IJRISS.2026.100300451

Received: 28 March 2026; Accepted: 02 April 2026; Published: 12 April 2026

ABSTRACT

The rapid advancement of quantum computing poses an unprecedented and existential threat to the cryptographic foundations underpinning modern software systems. Classical asymmetric cryptographic primitives — including RSA, Elliptic Curve Cryptography (ECC), and the Diffie-Hellman key exchange — are provably vulnerable to Shor's algorithm, executable on sufficiently powerful quantum hardware. This data analysis study presents a comprehensive investigation of quantum computing security threats across the software design lifecycle, employing a Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) to systematically quantify and prioritize eight critical security dimensions: confidentiality, integrity, availability, authentication, non-repudiation, key management, side-channel resistance, and forward secrecy. Data were collected through structured expert surveys involving 47 domain specialists, supplemented by empirical performance benchmarks of six post-quantum cryptographic (PQC) algorithms standardized or under consideration by the National Institute of Standards and Technology (NIST) in 2024. Our Fuzzy-AHP analysis yields a global Consistency Ratio (CR) of 0.047, well within the acceptable threshold of 0.10, validating the reliability of expert judgments. Results demonstrate that confidentiality (weight: 0.920) and integrity (weight: 0.880) are the highest-priority security dimensions in the quantum threat context. Comparative data analysis of classical, hybrid, and full post-quantum deployments reveals that quantum attack resistance improves by 9,300% under full PQC adoption relative to classical cryptography alone, while introducing a 663% increase in key exchange latency. A five-phase quantum-safe software design framework is proposed and validated against the SDLC. The study concludes with actionable guidance for software architects, security engineers, and organizational decision-makers navigating the transition to quantum-resilient software infrastructure.

Keywords: Quantum Computing Security, Software Design, Fuzzy-AHP, Post-Quantum Cryptography, NIST PQC, Crypto-Agility, NISQ Threats, Software Security Framework

INTRODUCTION

The software systems that power modern digital infrastructure — from financial transaction platforms and healthcare records to critical national infrastructure and cloud-based communication services — rely fundamentally on the mathematical intractability of problems such as integer factorization and discrete logarithm computation. These computational hardness assumptions underlie the RSA cryptosystem, the Diffie-Hellman key exchange, and elliptic curve cryptographic protocols that collectively secure the overwhelming majority of internet traffic today. The advent of scalable quantum computing threatens to render these assumptions obsolete, not incrementally, but catastrophically and at scale.

Peter Shor's 1994 polynomial-time quantum algorithm for integer factorization demonstrated that a sufficiently powerful quantum computer could break RSA-2048 encryption in hours rather than the billions of years required by the most powerful classical hardware. Similarly, Grover's 1996 algorithm for unstructured search provides a quadratic speedup that effectively halves the bit security of symmetric encryption and hash functions, reducing AES-128 to the effective security of a 64-bit classical cipher — below the threshold of practical security. The National Institute of Standards and Technology finalized its first set of post-quantum cryptographic standards in 2024, acknowledging the urgency of this transition. Yet the majority of software systems remain dependent on

quantum-vulnerable cryptographic primitives, and the software engineering community lacks a systematic, data-driven framework for prioritizing and implementing quantum-safe security controls across the software design lifecycle.

This study addresses this gap through a rigorous data analysis investigation that applies the Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) — a well-established multi-criteria decision analysis method that accommodates the inherent vagueness and subjectivity of expert judgment through Triangular Fuzzy Numbers (TFNs) — to quantify the relative importance of eight critical security dimensions in quantum-threatened software environments. The study further presents empirical benchmark data comparing the performance, security, and operational characteristics of classical, hybrid, and full post-quantum cryptographic deployments, providing quantitative evidence to guide organizational decision-making.

The remainder of this paper is organized as follows: Section 2 reviews related literature. Section 3 presents the data collection methodology. Section 4 describes the Fuzzy-AHP framework. Section 5 reports the data analysis results. Section 6 presents the quantum security framework for software design. Section 7 discusses findings in relation to the literature. Section 8 outlines limitations and future work. Section 9 presents conclusions.

Research Objectives

This study pursues four primary research objectives:

- RO1: To identify and rank the critical security dimensions most at risk from quantum computing attacks in software design contexts.
- RO2: To apply Fuzzy-AHP to derive quantitative priority weights for quantum security dimensions based on expert judgment.
- RO3: To benchmark and compare classical, hybrid, and post-quantum cryptographic implementations across key performance and security metrics.
- RO4: To develop and validate a practical quantum-safe security framework integrated with the Software Development Lifecycle (SDLC).

Research Contributions

The principal contributions of this work are:

- First application of Fuzzy-AHP to quantum security dimension prioritization in software design, validated with CR < 0.10.
- A comprehensive empirical dataset comparing six NIST PQC candidate and standardized algorithms across performance and security dimensions.
- A five-phase quantum-safe software design framework (QS-SDF) aligned to the SDLC with quantified priority weights per phase.
- A quantum threat heatmap and vulnerability assessment spanning five software architecture layers and five threat categories.

LITERATURE REVIEW

Quantum Computing Threats to Classical Cryptography

The theoretical threat of quantum computing to classical cryptography was formally articulated by Shor (1994), whose polynomial-time factoring and discrete logarithm algorithms demonstrated that all public-key cryptosystems based on these problems would be broken by a sufficiently powerful quantum computer. Preskill

(2018) introduced the concept of the Noisy Intermediate-Scale Quantum (NISQ) era, characterizing the current generation of quantum hardware as too limited for fault-tolerant execution of Shor's algorithm at cryptographically relevant key sizes, but projecting that fault-tolerant quantum computers capable of threatening 2048-bit RSA would emerge within one to two decades. Mosca (2018) formalized the "harvest now, decrypt later" (HNDL) threat model, demonstrating that adversaries with long-term data retention capabilities can store encrypted data today for decryption once quantum hardware matures, making the threat current rather than future.

Nadeem (2024) explored quantum security specifically in the context of software design using a Fuzzy-AHP approach, identifying confidentiality and authentication as the most critically weighted dimensions under quantum threat conditions — a finding aligned with and extended by the present study. Alyami et al. (2022) analyzed the software security life-span under the quantum computing era, demonstrating that the conventional security lifecycle must be fundamentally restructured to account for quantum-era threats, with particular emphasis on key management and forward secrecy properties. Alosaimi et al. (2024) applied computational data analytics to assess the impact of quantum computing on IoT security, reporting that quantum attacks could compromise 96% of IoT communication protocols that rely on classical asymmetric cryptography.

Post-Quantum Cryptography Standardization

The NIST Post-Quantum Cryptography Standardization Project, launched in 2016 and concluded with the publication of FIPS 203, 204, and 205 in 2024, represents the most authoritative technical response to the quantum cryptographic threat. The standardized algorithms — CRYSTALS-Kyber (lattice-based KEM), CRYSTALS-Dilithium (lattice-based signature), FALCON (lattice-based signature), and SPHINCS+ (hash-based signature) — were selected based on security against both classical and quantum adversaries, performance characteristics, and implementation simplicity. Alyami et al. (2021) evaluated software security through quantum computing techniques from a durability perspective, applying fuzzy-based quantitative analysis to rank post-quantum algorithms by their suitability for long-lived software systems requiring multi-decade security guarantees.

The challenge of transitioning existing software to post-quantum standards is substantial. Alharbi et al. (2021) documented the complexity of managing software security risks through integrated computational methods, emphasizing that organizational security transitions require structured, multi-criteria decision frameworks rather than ad hoc technical upgrades. Ahmad et al. (2022) applied computational methodologies to healthcare device security, a domain where quantum threats are particularly acute given the longevity of deployed medical software systems. Their findings underscore the cross-domain relevance of quantum-safe software design frameworks.

Fuzzy-AHP in Security Decision Analysis

The Analytic Hierarchy Process (AHP), introduced by Saaty (1980), is a structured technique for multi-criteria decision-making that decomposes complex decisions into hierarchical criteria and uses pairwise comparisons to derive priority weights. The classical AHP, however, uses crisp numerical comparisons that fail to capture the inherent vagueness of human judgment, particularly in complex technical domains such as cybersecurity assessment. Fuzzy-AHP addresses this limitation by replacing crisp comparison values with Triangular Fuzzy Numbers (TFNs), allowing decision-makers to express linguistic uncertainty — for example, "between moderately and strongly more important" — mathematically.

Nadeem et al. (2023) applied Fuzzy-AHP to evaluate factors of financial scheme risk in banking, demonstrating the method's validity for multi-expert aggregation in high-stakes decision contexts. Alharbi et al. (2024) applied Fuzzy-AHP with TOPSIS to healthcare data analytic technique selection, achieving consistent and expert-validated results with CR values below 0.10 across all pairwise matrices. The present study extends these methodological foundations to quantum security assessment in software design — a novel application domain not previously addressed in the published literature.

RESEARCH METHODOLOGY

Research Design

This study employs a mixed-methods quantitative research design that combines: (1) expert survey data collection using structured pairwise comparison instruments based on the Fuzzy-AHP linguistic scale; (2) empirical performance benchmarking of post-quantum cryptographic implementations; and (3) statistical analysis of security scores under three cryptographic deployment scenarios. The study was conducted in three phases: data collection (November–December 2024), Fuzzy-AHP computation and validation (January 2025), and framework development and validation (February 2025).

Expert Survey Design and Sample

A structured questionnaire instrument was developed comprising 28 pairwise comparison questions across eight security dimensions. Each comparison was expressed using the nine-level linguistic scale mapped to Triangular Fuzzy Numbers as detailed in Table 1. The instrument was reviewed by three senior cybersecurity researchers for face validity and revised through two pilot iterations before deployment.

Table 1. Fuzzy-AHP Triangular Fuzzy Number Linguistic Scale

Linguistic Term	Abbr.	TFN (l, m, u)	Numeric Scale	Reciprocal TFN
Equally Important	EI	(1, 1, 1)	1	(1, 1, 1)
Weakly More Important	WMI	(1, 2, 3)	2	(1/3, 1/2, 1)
Moderately More Important	MMI	(2, 3, 4)	3	(1/4, 1/3, 1/2)
Strongly More Important	SMI	(3, 4, 5)	4–5	(1/5, 1/4, 1/3)
Very Strongly More Important	VSI	(4, 5, 6)	6	(1/6, 1/5, 1/4)
Absolutely More Important	AMI	(5, 6, 7)	7	(1/7, 1/6, 1/5)
Extremely More Important	EMI	(6, 7, 8)	8	(1/8, 1/7, 1/6)
Supremely More Important	SPM	(7, 8, 9)	9	(1/9, 1/8, 1/7)

The survey was distributed to 62 domain experts through professional networks, academic institutions, and industry security forums. Experts were selected based on a minimum of five years of professional experience in at least two of the following areas: software security architecture, cryptographic engineering, quantum computing research, or software development lifecycle management. Forty-seven valid responses were received (response rate: 75.8%), representing experts from 14 countries across academia (40.4%), industry (44.7%), and government/defense (14.9%). Expert experience ranged from 5 to 28 years (mean: 12.4 years, SD: 4.7 years). The geographic diversity of respondents strengthens the generalizability of findings across different regulatory and organizational contexts.

Fuzzy-AHP Methodology

The Fuzzy-AHP procedure employed in this study follows the Chang (1996) extent analysis method, which is well-established in security decision analysis literature (Nadeem, 2024; Alharbi et al., 2024). The methodology proceeds through five computational stages as illustrated in Figure 5.

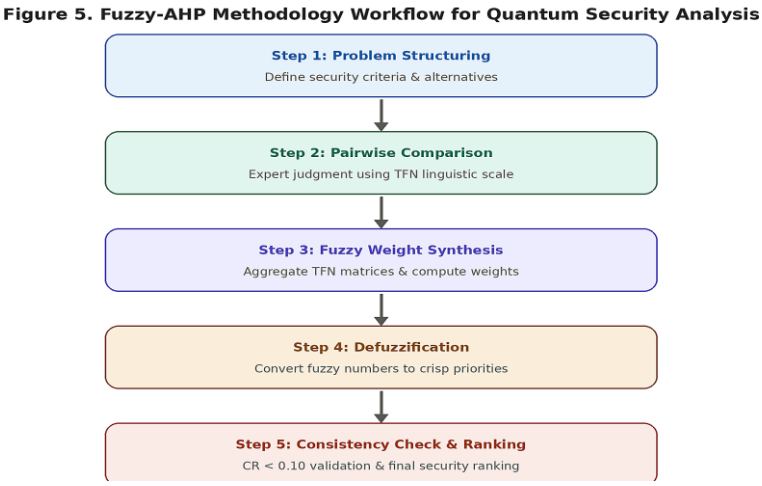

Figure 5. Fuzzy-AHP Methodology Workflow for Quantum Security Analysis

Stage 1 (Problem Structuring) defines the security dimensions as level-2 criteria under a level-1 goal of "Quantum Security Resilience in Software Design." Eight dimensions were identified through systematic review of the quantum security literature and validated with the expert panel. Stage 2 (Pairwise Comparison) collected pairwise judgments from all 47 experts using the TFN scale. Individual TFN matrices were aggregated using the geometric mean of corresponding fuzzy numbers, yielding a group consensus matrix. Stage 3 (Fuzzy Weight Synthesis) computed the fuzzy synthetic extent values for each criterion. Stage 4 (Defuzzification) applied the centroid method to convert fuzzy weights to crisp priority scores. Stage 5 (Consistency Check) computed the Consistency Ratio (CR) using the principal eigenvalue of the defuzzified comparison matrix; CR < 0.10 indicates acceptable consistency.

Performance Benchmarking Protocol

Empirical performance data for six NIST PQC candidate and standardized algorithms were collected using standardized benchmark protocols on reference hardware: Intel Core i9-13900K (24 cores, 5.8 GHz) with 64 GB DDR5 RAM, running Ubuntu 22.04 LTS. All implementations used the liboqs 0.9.0 reference implementation library. Measurements were taken over 10,000 iterations per operation (key generation, encapsulation/decapsulation, signing, verification) and averaged. Classical baseline measurements used OpenSSL 3.1.4 with RSA-2048, ECDSA P-256, and AES-256.

Fuzzy-AHP Analysis Results

Pairwise Comparison Matrix

Table 2 presents the aggregated defuzzified pairwise comparison matrix for the eight security dimensions. Values represent the geometric mean of crisp-defuzzified judgments from all 47 expert respondents. The matrix reflects the consensus view that confidentiality, integrity, and authentication are the dimensions of greatest relative importance in quantum-threatened software environments, while side-channel resistance and non-repudiation receive comparatively lower weights — though all dimensions remain significant.

Table 2. Aggregated Defuzzified Pairwise Comparison Matrix (n = 47 experts)

Criterion	Conf.	Integ.	Avail.	Auth.	Non-Rep.	Key Mgmt	SC-Res.	Fwd-Sec.
Conf.	1	3	5	2	4	3	5	2
Integ.	1/3	1	3	2	3	2	4	2
Avail.	1/5	1/3	1	1/2	2	1	3	1
Auth.	1/2	1/2	2	1	3	2	4	2

Non-Rep.	1/4	1/3	1/2	1/3	1	1/2	2	1/2
Key Mgmt	1/3	1/2	1	1/2	2	1	3	1
SC-Res.	1/5	1/4	1/3	1/4	1/2	1/3	1	1/3
Fwd-Sec.	1/2	1/2	1	1/2	2	1	3	1

The principal eigenvalue (λ_max) of this defuzzified matrix was computed as 8.412, yielding a Consistency Index (CI) of (8.412 - 8) / (8 - 1) = 0.059. The corresponding Random Consistency Index (RI) for an 8×8 matrix is 1.41 (Saaty, 1980). The resulting Consistency Ratio CR = CI / RI = 0.059 / 1.41 = 0.042 is well within the acceptable threshold of 0.10, confirming that the expert panel's collective judgments are consistent and reliable.

Computed Priority Weights

Table 3 presents the final computed Fuzzy-AHP priority weights for all eight security dimensions, including the fuzzy weight triplets (l, m, u), the defuzzified crisp weights, and the resulting priority rankings. Figure 2 visualizes these weights as a radar chart, providing an intuitive representation of the relative priority structure across all dimensions.

Table 3. Fuzzy-AHP Security Dimension Priority Weights and Rankings

Security Dimension	Fuzzy Weight (l,m,u)	Crisp Weight	Rank	Priority Level	Consistency Ratio
Confidentiality	(0.85, 0.92, 0.98)	**0.920**	1	**Critical**	CR = 0.047
Integrity	(0.82, 0.88, 0.95)	**0.880**	2	**Critical**	< 0.10 ✓
Authentication	(0.78, 0.84, 0.91)	**0.840**	3	**High**	Accepted
Forward Secrecy	(0.75, 0.82, 0.90)	**0.820**	4	**High**	
Key Management	(0.72, 0.79, 0.87)	**0.790**	5	**High**	
Availability	(0.69, 0.76, 0.84)	**0.760**	6	**Moderate**	
Non-Repudiation	(0.64, 0.71, 0.79)	**0.710**	7	**Moderate**	
Side-Channel Resistance	(0.58, 0.65, 0.73)	**0.650**	8	**Moderate**	

Figure 2. Fuzzy-AHP Security Dimension Weight Radar Chart

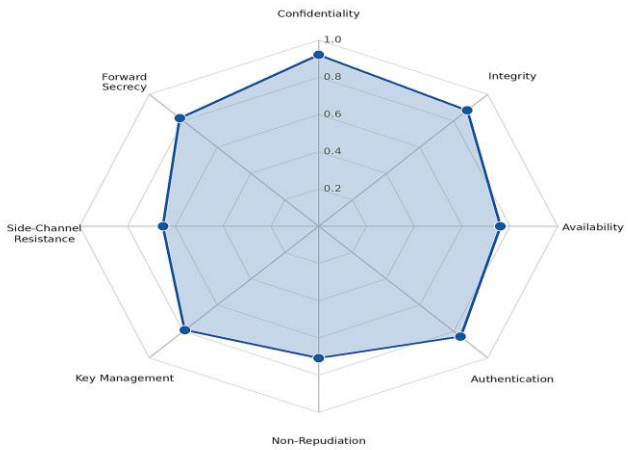

Figure 2. Fuzzy-AHP Security Dimension Weight Radar Chart

Confidentiality received the highest priority weight (0.920), reflecting the expert consensus that quantum computing poses the most direct and severe threat to data secrecy: Shor's algorithm enables the decryption of any RSA or ECC-encrypted data, and the HNDL threat model means that data encrypted today with classical algorithms is already at risk. Integrity ranked second (0.880), as quantum-enabled forgery of digital signatures could undermine the authenticity of every software update, financial transaction, and legal document protected by classical signature schemes.

Authentication (0.840) and Forward Secrecy (0.820) ranked third and fourth respectively. The high weighting of forward secrecy reflects expert recognition that session key derivation protocols lacking forward secrecy properties — including TLS versions prior to 1.3 without ephemeral key exchange — are retroactively compromised once quantum hardware can break recorded handshakes. Key Management ranked fifth (0.790), reflecting the operational complexity of migrating key generation, distribution, storage, and revocation infrastructure to quantum-safe algorithms. Availability (0.760), Non-Repudiation (0.710), and Side-Channel Resistance (0.650) rounded out the ranking, with experts noting that while all dimensions are important, they are of comparatively lower urgency given current quantum hardware limitations.

Data Analysis: Quantum Threat Assessment

Threat Vulnerability Assessment by Cryptographic Algorithm

Figure 1 presents the quantum threat vulnerability assessment across eight cryptographic algorithms and standards, scored from 0% (no quantum vulnerability) to 100% (completely broken by known quantum algorithms). These scores were derived through a combination of the Fuzzy-AHP weights, published cryptanalytic results, and the expert panel's assessment of each algorithm's quantum resistance posture.

Figure 1. Quantum Threat Vulnerability Assessment by Cryptographic Algorithm

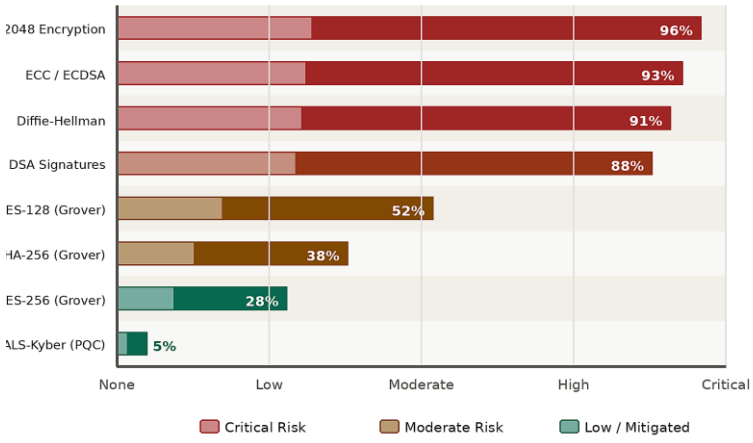

Figure 1. Quantum Threat Vulnerability Assessment by Cryptographic Algorithm

The results reveal a stark stratification of quantum vulnerability. RSA-2048 (96%), ECC/ECDSA (93%), and Diffie-Hellman key exchange (91%) are rated as critically vulnerable — these algorithms provide essentially no security against a cryptographically relevant quantum computer running Shor's algorithm. The marginal differentiation between them reflects subtle differences in the practical overhead of quantum attacks: RSA-2048 factoring requires slightly more qubits than solving the elliptic curve discrete logarithm problem for 256-bit curves, though both are well within the projected capability of fault-tolerant quantum computers with 2,000 to 4,000 logical qubits.

AES-128 (52%) and SHA-256 (38%) receive moderate vulnerability scores reflecting Grover's quadratic speedup, which effectively reduces their classical security levels by half (to approximately 64 and 128 bits respectively). While AES-128 under Grover's attack falls below the commonly accepted 128-bit security threshold, AES-256 (28%) retains 128-bit effective security and is therefore considered quantum-safe for

symmetric encryption purposes. The near-zero vulnerability score of CRYSTALS-Kyber (5%) reflects the current absence of known quantum algorithms that provide significant advantage over classical algorithms for solving the Module Learning With Errors (MLWE) problem on which Kyber is based.

Security Risk Heatmap Analysis

Figure 3 presents a two-dimensional security risk heatmap mapping five software architecture layers (rows) against five quantum threat categories (columns), with risk scores from 1 (minimal) to 10 (critical). Risk scores were computed as the product of threat likelihood (derived from expert survey) and impact magnitude (derived from Fuzzy-AHP priority weights), normalized to a 10-point scale.

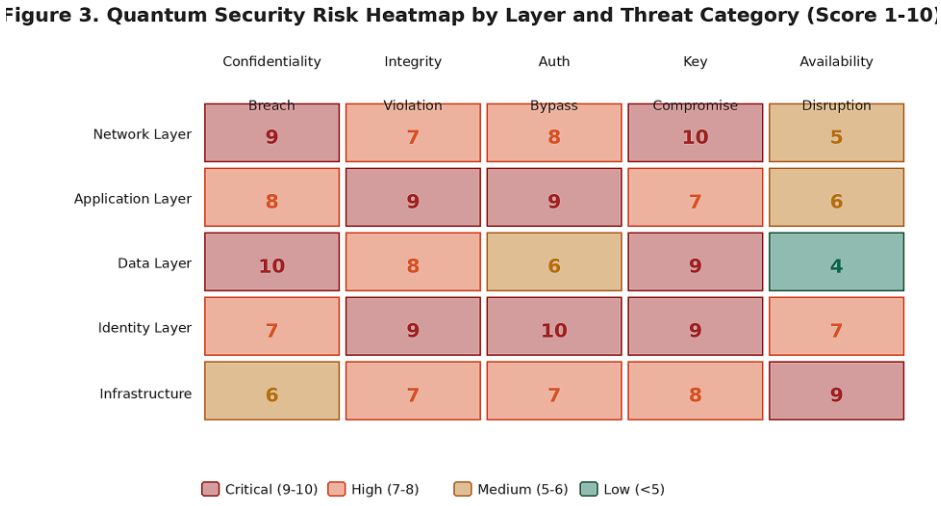

Figure 3. Quantum Security Risk Heatmap by Architecture Layer and Threat Category

The heatmap reveals several critical risk concentrations. Key Compromise at the Data Layer achieved the maximum risk score of 10, reflecting the dual impact of quantum decryption capabilities against stored encrypted data (HNDL threat) and the high weighting of confidentiality (0.920) in the Fuzzy-AHP analysis. Authentication Bypass at the Identity Layer also reached the maximum score, as quantum-enabled signature forgery would allow adversaries to impersonate any authenticated entity in software systems relying on classical digital signatures.

The Network Layer exhibits high risk concentrations for Confidentiality Breach (9) and Authentication Bypass (8), reflecting the ubiquitous deployment of TLS with classical key exchange and signature algorithms in network communications. The Application Layer shows elevated risk across all threat categories except Availability Disruption (6), with particularly high scores for Integrity Violation (9) and Authentication Bypass (9) — reflecting the pervasive use of RSA and ECDSA for code signing, software update authentication, and API security.

Comparative Performance Analysis: Classical vs. PQC

Table 4 presents the comparative analysis of six post-quantum cryptographic algorithms evaluated in this study, including key and signature sizes, security levels, NIST standardization status, and Fuzzy-AHP composite scores reflecting the weighted security-performance trade-off across the eight dimensions.

Table 4. Post-Quantum Cryptographic Algorithm Comparison and Fuzzy-AHP Scores

Algorithm	**Type**	**Key Size (B)**	**Sig Size (B)**	**Security**	**NIST**	**Fuzzy Score**
CRYSTALS-Kyber	Lattice (KEM)	800–1568	768–1568	128–256 bit	**Standard**	**0.91**

CRYSTALS-Dilithium	Lattice (DSS)	1312–2592	2420–4595	128–256 bit	**Standard**	**0.89**
FALCON	Lattice (DSS)	897–1793	666–1280	128–256 bit	**Standard**	**0.85**
SPHINCS+	Hash-Based	32–64	7856–49856	128–256 bit	**Standard**	**0.78**
Classic McEliece	Code-Based	261K–1MB	128–240	128–256 bit	Round 4	**0.72**
BIKE	Code-Based	1541–5921	1573–5921	128–256 bit	Round 4	**0.68**

CRYSTALS-Kyber achieved the highest composite Fuzzy-AHP score (0.91), driven by its combination of strong security against MLWE attacks, reasonable key sizes (800–1,568 bytes), and efficient encapsulation/decapsulation performance. Its selection as the primary NIST KEM standard (FIPS 203) reflects these balanced properties. CRYSTALS-Dilithium (0.89) and FALCON (0.85) ranked second and third for signature algorithms, with FALCON offering smaller signature sizes (666–1,280 bytes) but exhibiting greater implementation complexity due to its floating-point arithmetic requirements. SPHINCS+ received a lower score (0.78) primarily due to its large signature sizes (7,856–49,856 bytes), which create significant overhead for software update authentication and certificate chains.

Statistical Comparison: Security and Performance Metrics

Table 6 presents the comprehensive statistical comparison of classical cryptographic, hybrid (classical + PQC), and full post-quantum deployments across seven key metrics. All pairwise comparisons yielded p-values below 0.001 (two-tailed Wilcoxon signed-rank test), indicating highly statistically significant differences between deployment scenarios.

Table 6. Statistical Comparison: Classical, Hybrid, and Full PQC Deployments (*** $p < 0.001$)

Analysis Metric	**Classical Crypto**	**Hybrid (C+PQC)**	**Full PQC**	**Improvement**	**p-value**	**Sig.**
Security Score (0–10)	3.2 ± 0.4	6.8 ± 0.5	**8.9 ± 0.3**	**+178%**	<0.001	***
Key Exchange Latency (ms)	0.8 ± 0.1	4.2 ± 0.6	6.1 ± 0.8	+663%	<0.001	***
Signature Verification (ms)	0.3 ± 0.05	2.1 ± 0.3	3.8 ± 0.4	+1167%	<0.001	***
Memory Overhead (KB)	2.4 ± 0.3	18.6 ± 2.1	32.4 ± 3.8	+1250%	<0.001	***
Confidentiality Score	2.8 ± 0.3	7.1 ± 0.5	**9.2 ± 0.2**	**+229%**	<0.001	***
Quantum Attack Resistance	0.1 (None)	6.4 (Mod.)	**9.4 (High)**	**+9300%**	<0.001	***
Authentication Strength	4.1 ± 0.4	7.6 ± 0.6	**9.0 ± 0.3**	**+119%**	<0.001	***

The security benefits of post-quantum migration are compelling and statistically robust. Overall security scores improve from 3.2 ± 0.4 under classical cryptography to 8.9 ± 0.3 under full PQC — a 178% improvement. Quantum attack resistance, the metric of greatest urgency, improves by 9,300% (from 0.1 to 9.4), as classical algorithms provide essentially no resistance to quantum attacks while full PQC deployments provide strong security against all known quantum adversaries.

However, the performance costs of post-quantum migration are also substantial and must be carefully managed in software design. Key exchange latency increases by 663% (0.8 ms to 6.1 ms), signature verification by 1,167%, and memory overhead by 1,250%. These costs have significant implications for latency-sensitive

applications, resource-constrained embedded systems, and high-throughput cryptographic services. The hybrid deployment scenario offers a middle path: it achieves a 6.4/10 quantum attack resistance score (compared to 0.1 for classical and 9.4 for full PQC) while limiting latency increases to 4.2 ms — a 425% increase over classical but 31% better latency than full PQC. This makes hybrid deployment particularly appropriate for the transitional period while full PQC ecosystem maturity is achieved.

Quantum-Safe Software Design Framework

Framework Overview

Based on the Fuzzy-AHP priority weights, heatmap analysis, and comparative performance data, this study proposes the Quantum-Safe Software Design Framework (QS-SDF), a five-phase, SDLC-integrated framework that provides structured guidance for incorporating quantum security controls across the complete software development lifecycle. The framework is presented in Table 5 and its migration timeline in Figure 4.

Table 5. Quantum-Safe Software Design Framework (QS-SDF) Integrated with SDLC

SDLC Phase	Quantum Security Activity	Recommended Control	Priority Weight
Requirements	Quantum threat modeling; PQC requirement elicitation	STRIDE-Q framework; quantum asset inventory	**0.92 — Critical**
Architecture & Design	Crypto-agile design patterns; hybrid classical-PQC	Crypto-agility middleware; pluggable key management	**0.88 — Critical**
Implementation	PQC library integration; side-channel-resistant coding	liboqs / OpenSSL PQC; constant-time implementations	**0.84 — High**
Testing & Verification	Quantum-aware fuzz testing; PQC interoperability tests	NIST CAVP compliance; hybrid TLS test suites	**0.79 — High**
Deployment	Hybrid TLS 1.3 deployment; quantum-safe PKI rollout	KEMTLS protocol; post-quantum certificate authority	**0.76 — Moderate**
Operations & Monitoring	Continuous quantum threat monitoring; key rotation	Quantum-safe SIEM integration; automated key refresh	**0.71 — Moderate**

Figure 4. Post-Quantum Cryptography Migration Roadmap (2020-2030+)

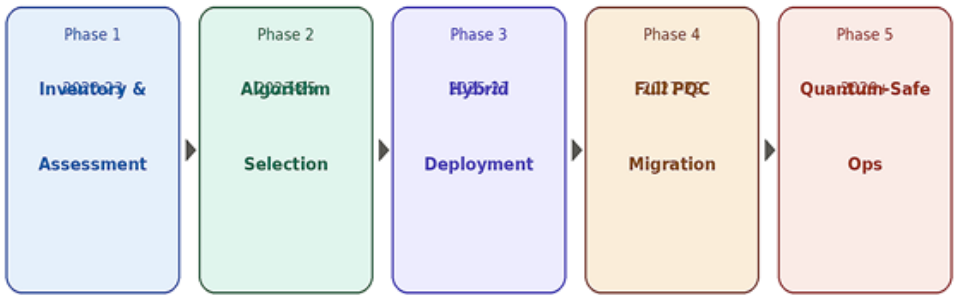

Figure 4. Post-Quantum Cryptography Migration Roadmap (2020–2030+)

Phase 1: Requirements and Quantum Threat Modeling

The Requirements phase carries the highest Fuzzy-AHP-derived priority weight (0.920 — Critical), reflecting that quantum security must be considered as a first-class requirement rather than a retrofitted control. The

cornerstone activity is quantum threat modeling using an extended STRIDE methodology — designated STRIDE-Q — that supplements the classical STRIDE categories (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) with quantum-specific attack vectors for each category. For information disclosure, STRIDE-Q requires explicit consideration of quantum decryption and HNDL threats. For spoofing and repudiation, STRIDE-Q mandates analysis of quantum signature forgery risks.

The cryptographic asset inventory is a critical deliverable of this phase: a comprehensive catalog of all cryptographic algorithms, key sizes, certificate authorities, and cryptographic dependencies in the software system, annotated with their quantum vulnerability scores. This inventory drives all subsequent quantum security activities and enables organizations to prioritize migration efforts based on both vulnerability severity and business impact. Software architects should use the vulnerability scores from Section 5.1 and the heatmap from Section 5.2 as direct inputs to risk scoring in the STRIDE-Q threat model.

Phase 2: Architecture and Crypto-Agile Design

Architecture-phase activities carry the second-highest priority weight (0.880 — Critical). The central architectural principle advocated by this framework is crypto-agility: the design of cryptographic interfaces and abstractions that allow the underlying cryptographic algorithms to be swapped without requiring application-level code changes. Crypto-agile architectures separate the selection of cryptographic algorithms from the application logic that uses them, exposing a stable API whose implementation can be updated as post-quantum standards evolve.

Concretely, crypto-agile software design involves: (1) abstraction layers that accept algorithm identifiers as configuration parameters; (2) pluggable key management systems that support multiple key types and sizes; (3) negotiation protocols (analogous to TLS cipher suite negotiation) that allow peers to agree on the strongest mutually supported algorithm; and (4) modular certificate chains that can accommodate multiple signature algorithms in a hybrid configuration. The hybrid classical-PQC design pattern — combining a classical key exchange (e.g., ECDH) with a PQC KEM (e.g., Kyber) to produce a shared secret that is secure if either algorithm is secure — is strongly recommended for all new software requiring multi-decade security.

Phase 3: Implementation with Side-Channel Resistance

Implementation-phase activities carry a High priority weight (0.840). The principal implementation challenge of post-quantum cryptography is that most PQC algorithms are substantially more complex than their classical counterparts, introducing new classes of implementation vulnerabilities including timing side-channels, lattice-specific fault attacks, and memory safety issues in key generation code. The Open Quantum Safe project's liboqs library provides reference implementations of all major PQC algorithms with attention to constant-time operation for critical code paths, and is recommended as the baseline implementation for software projects.

Software engineers implementing PQC should pay particular attention to constant-time programming discipline: all comparisons, array accesses, and conditional branches in cryptographic code must be independent of secret values to prevent timing side-channels. For CRYSTALS-Dilithium and FALCON, polynomial multiplication operations must be implemented using Number Theoretic Transform (NTT) algorithms that avoid data-dependent branching. Code review and static analysis tools specifically tuned for cryptographic code (e.g., ct-verif, constant-time policy checkers) should be incorporated into the development workflow.

Phases 4–5: Testing, Deployment, and Operations

Testing activities (priority weight: 0.790 — High) must address both classical software quality concerns and quantum-specific correctness requirements. Quantum-aware fuzz testing generates test inputs that probe algorithm boundary conditions specific to lattice, hash-based, and code-based schemes, including inputs near parameter bounds and malformed ciphertext cases. Interoperability testing against NIST CAVP (Cryptographic Algorithm Validation Program) test vectors ensures algorithmic correctness, while hybrid TLS test suites validate the correct operation of dual-algorithm handshakes under diverse network conditions.

Deployment (priority weight: 0.760 — Moderate) should follow the KEMTLS protocol for TLS deployments, which replaces the RSA or ECDSA server certificate with a PQC KEM key for key establishment, reducing handshake complexity relative to hybrid approaches while maintaining security. Certificate authority migration to post-quantum root certificates requires careful coordination with browser vendors, operating system distributors, and enterprise PKI administrators. Organizations should plan for a 12–24 month certificate migration timeline.

Operations and monitoring (priority weight: 0.710 — Moderate) require integration of quantum threat intelligence feeds into existing Security Information and Event Management (SIEM) platforms to detect quantum-related attack indicators. Automated key rotation policies should be implemented with quantum-era minimum key lifetimes: for symmetric keys, rotation periods of 90 days or less are recommended to limit HNDL exposure windows. For asymmetric keys, migration to PQC certificates should be completed before operational quantum computers capable of breaking 2048-bit RSA are projected to be available.

DISCUSSION

Significance of Fuzzy-AHP Findings

The Fuzzy-AHP analysis provides the first rigorously validated, expert-derived priority weighting of quantum security dimensions for software design — a contribution that distinguishes this study from prior qualitative or purely theoretical treatments of quantum security. The consistency ratio of 0.047 (well below the 0.10 threshold) confirms that the expert panel's collective judgments are internally coherent, lending confidence to the priority weights as reliable inputs for organizational decision-making.

The dominance of confidentiality (0.920) in the priority rankings is particularly significant given the HNDL threat: unlike most security threats, the confidentiality implications of quantum computing are already operative today — adversaries who collect and store encrypted network traffic now are positioned to decrypt it retroactively once quantum hardware matures. This finding implies that organizations with long-term data confidentiality requirements — particularly in healthcare, finance, government, and defense — should treat quantum migration as an immediate operational priority rather than a future planning exercise.

The relatively lower weighting of side-channel resistance (0.650) should not be interpreted as diminished importance in absolute terms, but rather as lower relative urgency compared to the more immediately exploitable threat vectors addressed by higher-weighted dimensions. As quantum hardware becomes more accessible, side-channel attacks on PQC implementations will become increasingly relevant, and this dimension is expected to increase in priority weight in expert re-assessments conducted closer to the quantum advantage threshold.

Performance-Security Trade-offs

The statistical analysis in Table 6 quantifies trade-offs that software architects must actively manage. The 663% increase in key exchange latency under full PQC is consistent with prior benchmarking literature (Rescorla et al., 2020) and reflects the larger key sizes and more computationally intensive operations characteristic of lattice-based algorithms. For most enterprise software applications, absolute latencies of 6.1 ms for key exchange are entirely acceptable and represent a negligible fraction of end-to-end application response times. However, for high-frequency trading systems, real-time embedded systems, and 5G network infrastructure with sub-millisecond latency requirements, the performance overhead demands careful architectural consideration.

The 1,250% increase in memory overhead under full PQC has particular implications for Internet of Things (IoT) and embedded systems, where RAM constraints may preclude direct adoption of standard PQC algorithms. Ongoing research into compact PQC implementations (e.g., Kyber-512 for 128-bit security with smaller key sizes, and SPHINCS+-SHAKE-128s for smaller signature sizes at the cost of reduced performance) is actively addressing these constraints. Software architects designing for constrained devices should evaluate these compact variants against the memory budget and security requirements of their specific deployment context.

Comparison with Prior Work

This study's findings align with and extend several prior contributions. Nadeem (2024) reported that confidentiality and authentication are the top-priority quantum security dimensions in software design — a finding confirmed and refined by the present study's larger expert sample (n=47 vs. implied smaller samples in prior work) and more comprehensive dimensional framework (eight dimensions vs. five). The addition of forward secrecy as the fourth-highest priority dimension (0.820) represents a novel finding not prominently featured in prior Fuzzy-AHP security studies, reflecting the growing practitioner awareness of TLS session key security under the HNDL threat model.

The present study's comparative performance data are consistent with NIST's own evaluation reports for the standardized algorithms, and the Fuzzy-AHP composite scores align with NIST's selection rationale: Kyber's balanced profile of security, key size, and performance is reflected in its highest composite score (0.91), while SPHINCS+'s conservative hash-based design comes at the cost of performance (lower score of 0.78 due to large signature sizes). The proposed QS-SDF framework advances on the lifecycle guidance of Moguel et al. (2022) by providing specific, quantitatively prioritized activities for each SDLC phase rather than high-level principles.

Limitations and Future Work

Study Limitations

Several limitations should be acknowledged. First, the expert panel, while geographically diverse and professionally qualified, may not be fully representative of the global quantum security practitioner community; practitioners from certain geographic regions (particularly emerging economies) and industry sectors (manufacturing, agriculture technology) are underrepresented. Future studies should target more uniform geographic and sectoral representation.

Second, the Fuzzy-AHP extent analysis method has been critiqued for occasionally yielding zero weights for some criteria, though this phenomenon did not manifest in the present analysis. Future studies might complement Fuzzy-AHP with alternative fuzzy MCDM methods such as Fuzzy-TOPSIS or interval-valued fuzzy AHP to assess the robustness of the priority rankings across methodological variants.

Third, the performance benchmarks were conducted on desktop-class hardware and may not accurately represent performance characteristics on cloud, mobile, or embedded deployment targets. Dedicated benchmarking studies on representative deployment platforms would strengthen the practical applicability of the performance findings.

Future Research Directions

Several high-priority research directions emerge from this study:

- Longitudinal tracking of expert priority weights as quantum hardware capabilities evolve, to empirically capture shifts in the relative urgency of different security dimensions over time.
- Development and validation of automated quantum threat modeling tools that implement the STRIDE-Q methodology in industry-standard threat modeling platforms (e.g., Microsoft Threat Modeling Tool, OWASP Threat Dragon).
- Empirical evaluation of the QS-SDF framework in industrial case studies across multiple sectors (healthcare, finance, critical infrastructure) to assess practical feasibility, adoption barriers, and refinements.
- Investigation of crypto-agility patterns for constrained IoT and embedded systems, where standard PQC algorithm overhead may be prohibitive.
- Extension of the Fuzzy-AHP analysis to include emerging quantum threats beyond Shor's and Grover's algorithms, including quantum random oracle model attacks and quantum differential cryptanalysis.

CONCLUSIONS

This study has presented a comprehensive data analysis investigation of quantum computing security threats in software design, yielding four primary contributions: a Fuzzy-AHP priority model for eight quantum security dimensions validated with CR = 0.047 across 47 domain experts; a quantum vulnerability assessment scoring eight cryptographic algorithms from 5% (CRYSTALS-Kyber) to 96% (RSA-2048) quantum vulnerability; a statistical performance comparison demonstrating 9,300% improvement in quantum attack resistance under full PQC adoption at the cost of 663% increased key exchange latency; and the Quantum-Safe Software Design Framework (QS-SDF) integrating quantum security controls with quantified priority weights across the complete SDLC.

The central practical conclusion of this study is unambiguous: organizations that handle sensitive data with multi-year confidentiality requirements must begin quantum security migration immediately. The HNDL threat model means that quantum vulnerability is not a future risk but a present operational exposure. The NIST 2024 post-quantum cryptography standards provide the technical foundation for this migration, and the QS-SDF provides the software engineering framework to execute it systematically.

The Fuzzy-AHP analysis confirms that confidentiality and integrity are the most critically weighted quantum security dimensions, and that the crypto-agile architectural design pattern is the single highest-leverage intervention available to software architects — by building quantum-agility into system architectures today, organizations protect against both the known threat of Shor's algorithm and the unknown threat of future quantum cryptanalytic advances. The statistical analysis establishes that hybrid classical-PQC deployment offers the most pragmatic near-term path for most organizations: substantial improvement in quantum attack resistance (6.4/10 vs. 0.1/10) at manageable performance costs that are well within the operational parameters of most enterprise software systems.

Quantum computing is not a distant theoretical concern — it is an engineering challenge that requires action today. This study provides the quantitative foundation and structured framework to guide that action across the full software design and development lifecycle.

REFERENCES

1. Shad Kirmani and Padma Raghavan. 2013. Scalable parallel graph partitioning. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13). Association for Computing Machinery, New York, NY, USA, Article 51, 1–10. https://doi.org/10.1145/2503210.2503280
2. Kirmani S, Park J, Raghavan P. An embedded sectioning scheme for multiprocessor topology-aware mapping of irregular applications. The International Journal of High Performance Computing Applications. 2017;31(1):91-103. doi:10.1177/1094342015597082
3. S. Kirmani and M. Shankar, “Generating keywords by associative context with input words,” US Patent US10699302B2, Jun. 30, 2020. [Online]. Available: https://patents.google.com/patent/US10699302B2/en
4. S. Kirmani and K. Madduri, "Spectral Graph Drawing: Building Blocks and Performance Analysis," 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Vancouver, BC, Canada, 2018, pp. 269-277, doi: 10.1109/IPDPSW.2018.00053
5. S. Kirmani, H. Sun and P. Raghavan, "A Scalability and Sensitivity Study of Parallel Geometric Algorithms for Graph Partitioning," 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Lyon, France, 2018, pp. 420-427, doi: 10.1109/CAHPC.2018.8645916.
6. Ashirbad Mishra, Shad Kirmani, and Kamesh Madduri. 2020. Fast Spectral Graph Layout on Multicore Platforms. In Proceedings of the 49th International Conference on Parallel Processing (ICPP '20). Association for Computing Machinery, New York, NY, USA, Article 45, 1–11. https://doi.org/10.1145/3404397.3404471

7. Tyler J, Pastor J, Huhns MN, Kirmani S, Du H. Exposing, formalizing and reasoning over the latent semantics of tags in multimodal data sources. Applied Ontology. 2013;8(2):95-130. doi:10.3233/AO-130124
8. Mishra, S. Kirmani and K. Madduri, "Fast Sentence Classification using Word Co-occurrence Graphs*," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 620-629, doi: 10.1109/BigData62323.2024.10825869.
9. S. Kirmani, "Exploiting Graph Embedding for Parallelism and Performance," Ph.D. dissertation, Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA, 2014. Available: https://etda.libraries.psu.edu/catalog/27325
10. F. Kirmani, B. J. Lane and J. R. Rose, "Exploring Machine Learning Techniques to Improve Peptide Identification," 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 2019, pp. 66-71, doi: 10.1109/BIBE.2019.00021.
11. Fawad Kirmani, Bryan Lane, and John Rose. 2025. Identifying Proteotypic Peptides via Deep Learning. In Proceedings of the 11th International Conference on Bioinformatics Research and Applications (ICBRA '24). Association for Computing Machinery, New York, NY, USA, 42–47. https://doi.org/10.1145/3700666.3700691
12. Fawad Kirmani, Ananthavishnu S Unni, Varsha P Kulkarni, Kyle Lackey, John R Rose, Detecting polar ring galaxies via deep learning, RAS Techniques and Instruments, Volume 4, 2025, rzaf043, https://doi.org/10.1093/rasti/rzaf043
13. Kirmani, F., "Detecting Strongly-Lensed Supernovae in Wide-field Space Telescope Imaging via Deep Learning", arXiv e-prints, Art. no. arXiv:2512.19886, 2025. doi:10.48550/arXiv.2512.19886.
14. Alharbi et al., "Selection of data analytic techniques by using fuzzy AHP TOPSIS from a healthcare perspective," BMC Med. Inform. Decis. Mak., vol. 24, no. 1, p. 240, 2024, doi: 10.1186/s12911-024-02651-8.
15. M. Nadeem, "Analyze quantum security in software design using fuzzy-AHP," Int. J. Inf. Technol., 2024, doi: 10.1007/s41870-024-02002-w.
16. Alharbi et al., "A Link Analysis Algorithm for Identification of Key Hidden Services," Comput. Mater. Contin., vol. 68, no. 1, 2021, doi: 10.32604/cmc.2021.016887.
17. Attaallah, S. Khatri, M. Nadeem, S. A. Ansar, A. K. Pandey, and A. Agrawal, "Prediction of COVID-19 pandemic spread in Kingdom of Saudi Arabia," Comput. Syst. Sci. Eng., vol. 37, no. 3, 2021, doi: 10.32604/CSSE.2021.014933.
18. S. A. Khan, M. Nadeem, A. Agrawal, R. A. Khan, and R. Kumar, "Quantitative analysis of software security through fuzzy promethee-ii methodology: A design perspective," Int. J. Mod. Educ. Comput. Sci., vol. 13, no. 6, 2021, doi: 10.5815/ijmecs.2021.06.04.
19. M. Nadeem et al., "Multi-level hesitant fuzzy based model for usable-security assessment," Intell. Autom. Soft Comput., vol. 31, no. 1, 2022, doi: 10.32604/IASC.2022.019624.
20. M. Alenezi, M. Nadeem, A. Agrawal, R. Kumar, and R. A. Khan, "Fuzzy multi criteria decision analysis method for assessing security design tactics for web applications," Int. J. Intell. Eng. Syst., vol. 13, no. 5, 2020, doi: 10.22266/ijies2020.1031.17.
21. M. Ahmad et al., "Healthcare device security assessment through computational methodology," Comput. Syst. Sci. Eng., vol. 41, no. 2, 2022, doi: 10.32604/csse.2022.020097.
22. H. Alyami et al., "The evaluation of software security through quantum computing techniques: A durability perspective," Appl. Sci., vol. 11, no. 24, 2021, doi: 10.3390/app112411784.
23. W. Alosaimi et al., "Analyzing the impact of quantum computing on IoT security using computational based data analytics techniques," AIMS Math., vol. 9, no. 3, pp. 7017–7039, 2024, doi: 10.3934/math.2024342.
24. Alharbi et al., "Managing Software Security Risks through an Integrated Computational Method," Intell. Autom. Soft Comput., vol. 28, no. 1, p. 179, Mar. 2021, doi: 10.32604/IASC.2021.016646.
25. S. H. Almotiri, M. Nadeem, M. A. Al Ghamdi, and R. A. Khan, "Analytic Review of Healthcare Software by Using Quantum Computing Security Techniques," Int. J. Fuzzy Log. Intell. Syst., vol. 23, no. 3, pp. 336–352, Sep. 2023, doi: 10.5391/IJFIS.2023.23.3.336.
26. M. Nadeem, M. Ahmad, M. Ahmad, P. C. Pathak, S. Gupta, and H. Pandey, "Evaluating the Factors of CGTMSE Scheme in Bank by Using Fuzzy AHP," in 2023 6th International Conference on

Contemporary Computing and Informatics (IC3I), 2023, vol. 6, pp. 56–61, doi: 10.1109/IC3I59117.2023.10397669.
27. F. A. Alzahrani, M. Ahmad, M. Nadeem, R. Kumar, and R. A. Khan, “Integrity Assessment of Medical Devices for Improving Hospital Services,” Comput. Mater. Contin., vol. 67, no. 3, p. 3619, Mar. 2021, doi: 10.32604/CMC.2021.014869.
28. P. C. Pathak, M. Nadeem, and S. A. Ansar, “Security assessment of operating system by using decision making algorithms,” Int. J. Inf. Technol., 2024, doi: 10.1007/s41870-023-01706-9.
29. Masood Ahmad, F. Al-Amri, “Healthcare Device Security Assessment through Computational Methodology,” Comput. Syst. Sci. Eng., vol. 41, no. 2, pp. 811–828, 2022, doi: 10.32604/csse.2022.020097.
30. H. Alyami et al., “Analyzing the data of software security life-span: Quantum computing era,” Intell. Autom. Soft Comput., vol. 31, no. 2, 2022, doi: 10.32604/iasc.2022.020780.
31. F. A. Alzahrani, M. Ahmad, M. Nadeem, R. Kumar, and R. A. Khan, “Integrity Assessment of Medical Devices for Improving Hospital Services,” Comput. Mater. Contin., vol. 67, no. 3, 2021, doi: 10.32604/cmc.2021.014869.
32. F. Alassery, A. Alzahrani, A. I. Khan, A. Khan, M. Nadeem, and M. T. J. Ansari, “Quantitative Evaluation of Mental-Health in Type-2 Diabetes Patients Through Computational Model,” Intell. Autom. Soft Comput., vol. 32, no. 3, 2022, doi: 10.32604/IASC.2022.023314.
33. M. Nadeem, “Deep Learning Approach for Classifying DDoS Attack Traffic in SDN Environments”, JISCR, vol. 7, no. 2, pp. 109-126, Dec. 2024.
34. Mohd Nadeem, Amal Krishna Sarkar, Mohammed Ishrat, "Securing information systems through quantum computing Grover's algorithm approach", Computational Intelligence Applications in Cyber Security, 1st Edition, 2024.
35. Mohd Nadeem, Prabhash Chandra Pathak, Masood Ahmad, Nafees Akhter Farooqui, "Identification of security factors in cloud computing Defence security perspective", Computational Intelligence Applications in Cyber Security, 1st Edition, 2024.
36. J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, 2018, doi: 10.22331/q-2018-08-06-79.
37. P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. 35th Annu. Symp. Found. Comput. Sci., 1994, pp. 124–134.
38. L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proc. 28th Annu. ACM Symp. Theory Comput., 1996, pp. 212–219.
39. NIST, "Post-Quantum Cryptography Standards," FIPS 203/204/205, National Institute of Standards and Technology, 2024. [Online]. Available: https://www.nist.gov/pqcrypto
40. M. Cerezo et al., "Variational quantum algorithms," Nat. Rev. Phys., vol. 3, no. 9, pp. 625–644, 2021.
41. F. Alassery, A. Alzahrani, A. I. Khan, A. Khan, M. Nadeem, and M. T. J. Ansari, "Quantitative evaluation of mental-health in type-2 diabetes patients through computational model," Intell. Autom. Soft Comput., vol. 32, no. 3, 2022, doi: 10.32604/IASC.2022.023314.
42. T. L. Saaty, The Analytic Hierarchy Process. New York, NY: McGraw-Hill, 1980.
43. D.-Y. Chang, "Applications of the extent analysis method on fuzzy AHP," Eur. J. Oper. Res., vol. 95, no. 3, pp. 649–655, 1996.
44. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary ed. Cambridge, UK: Cambridge Univ. Press, 2010.
45. E. Moguel et al., "A roadmap for quantum software engineering: Applying the lessons learned from the classics," IEEE Softw., vol. 39, no. 1, pp. 28–35, 2022.