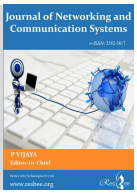
Vol.9 Issue 4 Oct 2026

Journal of Networking and Communication Systems

Received 26 September, Revised 16 October, Accepted 29 October



Quantum-Resistant Security Framework for Next-Generation Digital Banking Networks

Shweta Marigoudar
Dean, Faculty of Computing and IT
GM University, Davangere, Karnataka, India
Email: dean.fcit@gmu.ac.in

Prakruthi S B N
Master of Computer Applications (MCA)
GM University, Davangere, Karnataka, India
Email: prakruthinagaraj41@gmail.com

Bhoomika N
Master of Computer Applications (MCA)
GM University, Davangere, Karnataka, India
Email: bhumika77955@gmail.com

Abstract: The rapid expansion of mobile banking, internet banking, cloud services, digital payments, and interconnected financial platforms has increased the amount of sensitive information that must be protected over long periods. Traditional public-key mechanisms such as RSA and elliptic-curve cryptography remain important for contemporary systems, but their security assumptions are threatened by sufficiently capable quantum computers. Shor's algorithm provides a fundamental reason for treating the future quantum threat as a migration problem rather than a distant theoretical concern. A related concern is the Harvest-Now, Decrypt-Later model, in which adversaries collect encrypted information today with the intention of processing it when stronger computational capabilities become available. This paper presents an expanded Quantum-Resistant Security Framework for next-generation digital banking networks. The framework is centered on NIST-standardized post-quantum cryptographic mechanisms, particularly ML-KEM for quantum-resistant key establishment and ML-DSA for digital authentication and signatures. TLS 1.3 and AES-256-GCM are retained as secure communication and data-protection layers, while a Crypto-Agility Manager separates cryptographic policy from application logic so that algorithms can be changed with limited disruption. A Software-Defined Networking Security Controller provides centralized policy enforcement and controlled network-flow management. The architecture is organized as cooperating modules for authentication, secure communication, cryptographic lifecycle management, network security, and overall quantum-resistant coordination. The study extends the original framework with a detailed threat model, mathematical representation, deployment workflow, migration strategy, performance considerations, security analysis, scalability discussion, implementation roadmap, and limitations. Rather than claiming implementation in a production banking environment, the study proposes a practical architecture to guide the staged transition from classical public-key cryptography to quantum-resistant protection. The proposed approach emphasizes long-term confidentiality, adaptable security policy, centralized control, and continuity of banking services during cryptographic migration.

Keywords: Post-Quantum Cryptography, Digital Banking Security, ML-KEM, ML-DSA, Crypto-Agility, Software-Defined Networking, NIST FIPS 203, NIST FIPS 204, TLS 1.3, AES-256-GCM, Quantum Computing.

1. Introduction

Digital banking has transformed financial services by allowing customers to access accounts, transfer funds, manage cards, receive notifications, and perform payments without visiting a physical branch. The same transformation has increased the dependency of financial institutions on distributed computing, public networks, cloud platforms, application programming interfaces, mobile devices, and third-party services [21][22][23][24]. Consequently, banking security can no longer be treated as a single encryption component. It must be viewed as a layered system in which identity, key establishment, data protection, network control, policy management, and operational continuity work together.

The original research paper identifies the growing mismatch between classical public-key cryptography and the long-term security requirements of digital banking. RSA and ECC depend on mathematical problems that are considered difficult for classical computers. A sufficiently capable quantum computer changes this assumption. Shor's algorithm can efficiently address integer factorization and discrete logarithm problems, creating a future threat to widely deployed public-key mechanisms. Symmetric encryption is affected differently, but public-key establishment and digital signatures require particular attention because they support authentication and secure key exchange across banking infrastructure.

The problem is therefore not simply to select a replacement algorithm. Banks operate large legacy environments in which cryptographic functions are distributed across mobile applications, web gateways, payment services, authentication servers, databases, APIs, transaction processors, and network devices. A migration that changes every cryptographic dependency simultaneously could create operational risk. The proposed framework addresses this issue by combining post-quantum cryptography with crypto-agility and centralized network management [23].

© Resbee Info Technologies Pvt Ltd

The central research question is: How can a digital banking network integrate quantum-resistant key establishment and authentication while preserving secure communication, centralized policy control, scalability, and gradual migration from existing cryptographic mechanisms?

The proposed answer is a modular framework based on ML-KEM, ML-DSA, TLS 1.3, AES-256-GCM, a Crypto-Agility Manager, and an SDN Security Controller. These components are not treated as isolated technologies; their interaction forms the principal contribution of the study.

1.1 Problem Statement and Motivation

Financial records can have long confidentiality requirements. Transaction histories, account information, identity records, internal banking messages, and regulatory records may remain valuable long after the original communication occurs. This creates a special concern under the Harvest-Now, Decrypt-Later threat model. An attacker does not necessarily need to decrypt intercepted traffic immediately. If the captured ciphertext remains useful for many years, future computational advances may turn currently protected information into a target.

The second challenge is cryptographic dependency. A banking application may depend on a certificate, key exchange mechanism, signature algorithm, transport protocol, hardware security module, API gateway, or network policy without exposing that dependency directly to the application developer. If cryptographic choices are embedded throughout software and infrastructure, replacing them becomes expensive and error-prone. This motivates crypto-agility: the ability to change cryptographic mechanisms through controlled policy and configuration rather than rewriting an entire application.

The third challenge is centralized security management. Distributed banking systems require consistent enforcement of access policies and traffic controls. Software-Defined Networking provides a useful management model because network decisions can be coordinated through a logically centralized controller. In the proposed framework, SDN is not presented as a replacement for cryptography. Instead, it provides a control layer through which security policies, segmentation, routing decisions, and communication rules can be coordinated [23].

1.2 Research Objectives

- Develop a modular quantum-resistant security architecture for digital banking networks.
- Use ML-KEM for quantum-resistant key establishment and ML-DSA for digital authentication.
- Retain TLS 1.3 and AES-256-GCM as secure communication and payload-protection layers.
- Introduce a Crypto-Agility Manager capable of policy-driven cryptographic transition.
- Use an SDN Security Controller for centralized network-security policy enforcement.
- Represent the authentication, key establishment, encryption, and secure-session processes mathematically.
- Compare the proposed approach with classical RSA/ECC-oriented security architecture.
- Identify deployment constraints, scalability requirements, operational limitations, and future research directions.

1.3 Contributions Of The Study

The framework contributes an integrated security architecture rather than proposing a new cryptographic primitive. The first contribution is architectural: post-quantum key establishment and digital signatures are positioned inside a complete banking-security workflow. The second contribution is operational: crypto-agility is treated as a first-class component, allowing algorithm selection to be controlled independently from application functions [19]. The third contribution is network-oriented: SDN is incorporated to support centralized policy enforcement and traffic management. The fourth contribution is analytical: the framework is expressed using mathematical relationships that connect authentication, key establishment, encryption, cryptographic policy, and secure-session construction.

The framework also emphasizes staged migration, allowing classical and post-quantum mechanisms to coexist during a controlled transition. This approach avoids the need for immediate infrastructure replacement and provides an opportunity to evaluate interoperability, overhead, reliability, and security before wider deployment.

1.4 Scope of the Paper

The scope is focused on networked digital banking environments, including mobile banking, online banking, ATM-facing services, authentication infrastructure, transaction services, and supporting network components. The paper addresses cryptographic key establishment, digital authentication, secure

transport, payload encryption, policy management, and network security. It does not claim to solve every banking-security issue. Fraud analytics, endpoint compromise, insider threats, physical attacks, social engineering, and all forms of application-layer vulnerability remain broader security concerns [25][26].

1.5 Organization of the Paper

The paper proceeds from the security problem to the proposed architecture and its mathematical representation. The literature review establishes the relationship between post-quantum cryptography, crypto-agility, SDN, and financial-sector readiness. The system architecture then defines the modules and workflow. Subsequent sections discuss implementation requirements, experimental methodology, comparative evaluation, threat analysis, scalability, migration planning, limitations, and future research. The final section summarizes the framework and its practical significance.

2. Literature Review

Research on quantum-resistant banking security sits at the intersection of cryptography, network security, financial technology, and infrastructure migration. Earlier banking systems were designed around classical public-key mechanisms such as RSA and elliptic-curve cryptography. These mechanisms remain relevant to current infrastructure, but the possibility of large-scale quantum computing creates a need to prepare for alternatives [7][18][20]. The literature supplied with the original paper highlights NIST's 2024 standards, research on crypto-agility, post-quantum TLS [5], SDN-based security [6], and financial-sector readiness [8][9][10][11][12].

2.1 Post-Quantum Cryptography

The standardization of post-quantum cryptography provides a foundation for migration. FIPS 203 specifies ML-KEM, a module-lattice-based key encapsulation mechanism intended for secure key establishment. FIPS 204 specifies ML-DSA, a module-lattice-based digital signature standard [1]. FIPS 205 specifies SLH-DSA, a stateless hash-based digital signature standard [3]. These standards are important because they move post- quantum cryptography from experimental research toward standardized mechanisms that organizations can evaluate for practical deployment [7][10].

The research literature also includes the earlier CRYSTALS-Kyber and CRYSTALS-Dilithium work that influenced the standardized ML-KEM and ML-DSA families [13][14]. The relationship between research constructions and standardized mechanisms is relevant to banking migration because an organization needs both algorithmic confidence and stable implementation specifications. Standardization also supports procurement, interoperability testing, compliance planning, and long-term Cryptographic lifecycle management [7][8][20].

2.2 Quantum Threat To Classical Public-Key Systems

Shor's algorithm is central to the quantum threat model considered by this study [18]. Its significance is not that every current quantum computer can immediately break RSA or ECC; rather, it demonstrates that the underlying mathematical assumptions are vulnerable to a sufficiently capable quantum computer. This changes the strategic planning horizon for financial institutions. The security of data with long retention periods must be evaluated against future computational capabilities rather than only current attack capabilities [7][9][10][20].

2.3 Crypto-Agility In Financial Systems

Crypto-agility is repeatedly identified in the supplied literature as a practical requirement for organizations preparing for cryptographic transition [4][7][8]. Its primary value is architectural separation. Applications should not have to understand every cryptographic algorithm supported by the institution. Instead, policy and cryptographic services can determine which mechanism is used for a particular security context [4][8]. This reduces the cost of replacing algorithms and allows a security team to respond to newly identified weaknesses or new standards.

In a banking environment, crypto-agility also supports controlled coexistence. A migration policy may specify that new services use post-quantum mechanisms while selected legacy interfaces continue to use existing mechanisms for compatibility [7][8][11][12]. Over time, the policy can be tightened as interoperability improves. Such a design avoids treating migration as a single event. Instead, migration becomes a managed lifecycle involving discovery, classification, testing, deployment, monitoring, and retirement.

2.4 Post-Quantum Tls

Research on post-quantum TLS 1.3 indicates that secure transport can be adapted to accommodate post-quantum key-establishment mechanisms [5]. This is relevant to banking because TLS already provides an established communication framework. The proposed architecture therefore retains TLS 1.3 while introducing ML-KEM at the key-establishment layer [1][5]. The purpose is to preserve the operational model of secure transport while strengthening the public-key component against future quantum attacks [16].

2.5 Software-Defined Networking

SDN research demonstrates the value of centralized network control, dynamic traffic management, and programmable policy enforcement [6][17]. In the proposed architecture, the SDN Security Controller provides a centralized point for applying network-security policies. This can support segmentation of banking services, controlled communication between security zones, traffic isolation, and rapid response to detected risks [6][15][17]. However, centralization also introduces an important dependency: the controller itself must be strongly protected and highly available.

2.6 Research Gap

The key gap identified in the supplied literature is the limited integration of these technologies into one digital-banking security architecture. Post-quantum algorithms [1][2][3], crypto-agility [4], SDN [6], and secure transport [5][16] have often been studied separately. The proposed work addresses this gap by defining how these elements can cooperate within a single end-to-end banking session. The emphasis is therefore on integration, migration, and operational architecture rather than on inventing a new cryptographic algorithm.

3. System Architecture

The proposed Quantum-Resistant Security Architecture for Digital Banking contains five core functional areas: Authentication Module, Secure Communication Module, Crypto-Agility Manager, SDN Security Controller, and Quantum-Resistant Security Manager. Digital banking services form the application layer that consumes these capabilities. The architecture is designed so that a customer request passes through identity verification, quantum-resistant key establishment, encrypted communication, policy enforcement, and secure session completion.

3.1 Authentication Module

The Authentication Module verifies the identity of a customer before access to banking functions is granted. The original framework combines multi-factor authentication with ML-DSA signature verification and device certificate checks. This creates multiple verification conditions instead of relying on a password alone. The module can be implemented as a gateway service so that the banking application receives a verified security context rather than directly handling every cryptographic operation.

3.2 Secure Communication Module

Once authentication succeeds, the Secure Communication Module establishes a session using ML-KEM and protects subsequent traffic using TLS 1.3 and AES-256-GCM. ML-KEM provides the mechanism for establishing a shared secret, while the symmetric session key is then used for efficient data protection. This separation is important because symmetric encryption is well suited to high-volume transaction traffic, whereas public-key operations are primarily used for establishing trust and keys.

3.3 Crypto-Agility Manager

The Crypto-Agility Manager maintains cryptographic policy, algorithm status, version information, compatibility rules, and migration decisions. It provides an abstraction between banking services and the algorithms they use. For example, an application can request a secure session according to a security profile rather than explicitly embedding a particular algorithm. The manager then selects an approved mechanism according to policy and operational context.

3.4 SDN Security Controller

The SDN Security Controller manages network behavior through programmable rules. The original

framework refers to OpenFlow v1.3 as the networking technology. Within the banking architecture, the controller can define permitted communication paths between authentication services, transaction services, databases, gateways, monitoring systems, and external interfaces. Centralized policy enforcement can reduce inconsistent configuration across distributed network devices.

3.5 Quantum-Resistant Security Manager

The Quantum-Resistant Security Manager coordinates the other security components. Its responsibility is not to replace each module but to ensure that their security decisions form one coherent session. It can maintain the state of a transaction security context, determine whether the required authentication factors are complete, request key establishment, apply cryptographic policy, and coordinate network rules. This component also provides a natural location for audit events associated with quantum-resistant security operations.

3.6 Digital Banking Services

The application layer includes customer-facing services such as account access, money transfers, payments, account information, alerts, and notifications. These services should remain insulated from low-level cryptographic details. Such separation reduces application changes during migration and makes it easier to apply common security controls consistently across multiple channels such as mobile applications, web banking, and ATM services.

3.7 Architectural Data Flow

A typical request begins when a customer accesses a banking channel. Credentials, a device context, and an authentication message are passed to the Authentication Module. Successful verification creates an authenticated security context. The Secure Communication Module then performs ML-KEM-based key establishment. TLS 1.3 protects the communication channel, and AES-256-GCM protects application payloads. In parallel, the Crypto-Agility Manager verifies that the selected mechanisms comply with current policy, while the SDN controller enforces permitted network flows. The security manager coordinates the overall process until the session is terminated.

Fig. 1 depicts the Quantum-Resistant Security Architecture with Digital Banking.

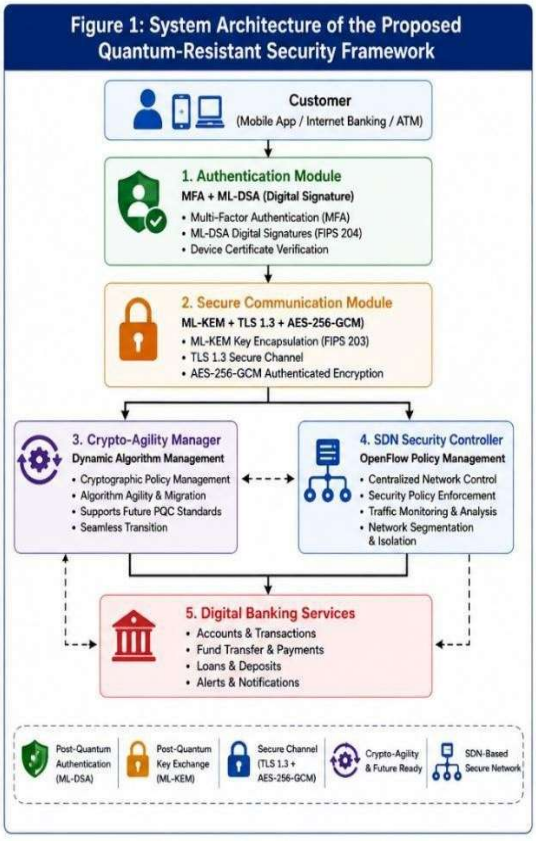


Fig. 1. *Quantum-Resistant Security Architecture with Digital Banking*

3.7.1 End-To-End Workflow

Table 1 presents the nine-stage workflow, from customer access and authentication to secure communication, SDN policy enforcement, banking transaction processing, and session termination.

Table 1. *End-to-End Workflow of the Proposed Quantum-Resistant Security Framework*

Stage	Operation
Step 1 – Customer Access	The user accesses mobile banking, online banking, or an ATM-facing service. The request is received through a protected entry point.
Step 2 – Multi-Factor Authentication	User credentials and additional authentication factors are checked. The registered device context is also evaluated.
Step 3 – ML-DSA Verification	The authentication message is associated with an ML-DSA signature and verified against the appropriate public key.
Step 4 – ML-KEM Key Establishment	The communication endpoint performs ML-KEM encapsulation and decapsulation to derive a shared session secret.
Step 5 – TLS 1.3 Channel	The secure transport channel is established, and the session uses the negotiated security parameters.
Step 6 – AES-256-GCM Protection	Banking messages are encrypted and authenticated using the session key.
Step 7 – SDN Security Enforcement	Network policies are applied to the relevant flows and services.
Step 8 – Banking Transaction	The requested banking service is executed within the authenticated and protected session.
Step 9 – Session Completion	The session is closed according to policy, temporary keys are retired, and relevant security events are recorded.

3.8 Security Boundaries

The architecture can be divided into several trust boundaries. The first is between the customer device and the bank's external gateway. The second is between the gateway and authentication services. The third is between security services and internal transaction-processing systems. The fourth concerns management traffic associated with the SDN controller and cryptographic policy services. Each boundary should be protected using least privilege, authenticated communication, monitoring, and carefully scoped network access.

3.9 Design Principles

- Layered security: no single mechanism is expected to provide every security property.
- Cryptographic separation: key establishment, signatures, transport, and payload protection have distinct responsibilities.
- Policy-driven agility: algorithm selection is separated from application logic.
- Least privilege: network and service permissions are restricted to required communication paths.
- Migration readiness: classical and post-quantum mechanisms can coexist during transition.
- Operational continuity: security improvements should not require unnecessary interruption of banking services.
- Auditability: security decisions should generate sufficient records for investigation and compliance.

4. Proposed Quantum-Resistant Security Framework

4.1 Framework Components

4.1.1 Authentication Module

Authentication is treated as a multi-condition decision. Passwords or other user credentials establish the first identity signal, additional factors strengthen the decision, and device verification helps establish whether the request originates from an expected environment. ML-DSA provides the digital-signature component in the proposed model. The combination reduces dependence on a single credential and prepares the signature layer for a post-quantum environment.

4.1.2 Secure Communication Module

The Secure Communication Module begins operation after the identity checks are successful. ML-KEM is used for key establishment. The resulting shared secret is used to support the secure session, while TLS

1.3 provides the communication protocol and AES-256-GCM provides authenticated encryption for protected data. This division allows the architecture to combine a post-quantum key-establishment mechanism with a high-throughput symmetric encryption layer.

4.1.3 Crypto-Agility Manager

Cryptographic agility is implemented as a lifecycle service. The manager maintains approved algorithms, security profiles, key lifetimes, compatibility rules, and transition states. It should also support algorithm deprecation. If a mechanism becomes unacceptable because of a newly discovered weakness, an organization can change policy and direct new sessions toward an approved alternative without rewriting every banking service.

4.1.4 SDN Security Controller

The SDN controller provides centralized control over network paths and segmentation. Security policies can distinguish authentication traffic, transaction traffic, management traffic, monitoring traffic, and external service traffic. The controller can support isolation of sensitive services and rapid modification of forwarding behavior when a security event is detected. Because it is a central control component, redundancy, access control, secure management channels, and fail-safe behavior are essential.

4.1.5 Quantum-Resistant Security Manager

The Quantum-Resistant Security Manager coordinates the framework and provides an integrated security context. It can verify that authentication is complete before allowing key establishment, verify that the selected cryptographic profile satisfies policy, and coordinate network access with the SDN controller. During migration, it can also identify whether a session belongs to a legacy, transitional, or post-quantum security profile.

4.2 Cryptographic and Session Management

4.2.1 Cryptographic Policy Model

A banking institution requires more than a list of algorithms. It needs policy rules describing where algorithms may be used, which combinations are acceptable, how keys are generated and retired, and what actions are taken when an algorithm is deprecated. The Crypto-Agility Manager therefore interprets a security policy as a set of constraints. The selected algorithm must satisfy the required security level, interoperability conditions, application requirements, and migration state.

4.2.2 Key Lifecycle Management

Key lifecycle management includes generation, distribution or encapsulation, activation, use, rotation, revocation, archival where required, and destruction. ML-KEM establishes shared session material, but operational security still depends on protecting private keys and controlling where cryptographic operations occur. In a banking environment, hardware-backed protection and restricted administrative access can strengthen the implementation. The framework treats key lifecycle management as an operational responsibility rather than an automatic property of the algorithm.

4.2.3 Session Termination

When a banking session ends, temporary session material should no longer be usable. The system should record security-relevant events, terminate active connections, and apply key-retirement rules. Session termination is especially important for banking because a user may perform multiple sensitive operations in a short period. Proper closure limits the useful lifetime of session secrets and reduces the consequences of session-state compromise.

4.3 Threat Model and Security Analysis

The threat model considers adversaries who may intercept network traffic, attempt to impersonate customers or services, compromise endpoints, exploit weak configuration, or collect encrypted data for future decryption. The model also considers operational threats associated with incorrect cryptographic migration. A quantum-resistant design is useful only if surrounding authentication, policy, network, and implementation controls are also addressed.

4.3.1 Harvest-Now, Decrypt-Later

Under the HNDL model, an adversary records encrypted traffic today and attempts to decrypt it later after quantum capabilities become stronger. This threat is particularly relevant to information with long

confidentiality lifetimes. The proposed use of ML-KEM for key establishment is intended to reduce dependence on classical public-key assumptions for future confidentiality.

4.3.2 Impersonation Threat

An attacker may attempt to impersonate a legitimate customer or service by stealing credentials, replaying messages, or forging authentication material. The framework combines multi-factor authentication, device verification, and ML-DSA signature validation. These controls operate together; ML-DSA does not replace the need for secure credential management or endpoint protection.

4.3.3 Man-In-The-Middle Threat

A network attacker may attempt to position themselves between a customer and a banking service. Secure authenticated transport, key establishment, and digital signatures are intended to prevent unauthorized modification or interception from becoming a valid banking session. Proper certificate and key validation remain necessary because post-quantum algorithms cannot compensate for incorrect trust configuration.

4.3.4 Replay Threat

A replay attack occurs when an attacker captures a valid message and attempts to submit it again. Banking protocols should therefore associate security messages with fresh session state, nonces, timestamps, transaction identifiers, or other anti-replay controls as appropriate. The proposed architecture establishes a fresh secure session and treats transaction authorization as separate from simple message encryption.

4.3.5 Network Compromise

A compromised network segment can expose traffic patterns, attempt unauthorized connections, or interfere with service availability. SDN-based segmentation can limit the paths available to an attacker. The controller can apply rules that restrict which services are allowed to communicate. However, the controller must itself be treated as a high-value security asset and protected against unauthorized modification.

4.3.6 Cryptographic Misconfiguration

Migration creates a new class of risk: an institution may deploy a technically strong algorithm incorrectly. Examples include inconsistent parameter selection, unsupported combinations, expired certificates, unprotected private keys, or an unintended fallback to a weaker mechanism. The Crypto-Agility Manager is intended to reduce these risks by centralizing policy and maintaining approved configurations.

4.3.7 Insider And Administrative Threats

Administrative privileges can affect cryptographic policy, network routing, and security configuration. Strong role separation is therefore required. Cryptographic administrators, network administrators, application operators, and auditors should have clearly defined permissions. Changes to security policy should be logged and, for high-impact operations, subject to approval and change-management procedures.

4.3.8 Availability Threat

Banking security must preserve availability as well as confidentiality and integrity. A security controller, authentication service, or cryptographic service that becomes unavailable could prevent legitimate transactions. The framework should therefore use redundancy, health checks, failover mechanisms, and carefully defined degraded-operation policies. Security should not be designed as a single point of failure.

4.3.9 Threat-To-Control Mapping

Table 2 maps the major security threats to their primary and supporting controls, demonstrating how cryptographic mechanisms, access controls, SDN policies, auditing, and redundancy work together to strengthen digital banking security.

Table 2. *Threat-to-Control Mapping for the Proposed Quantum-Resistant Security Framework*

Threat	Primary Control	Supporting Control
HNDL	ML-KEM	Crypto-agility and long-term migration
Impersonation	MFA + ML-DSA	Device verification
MITM	TLS 1.3 + authenticated key establishment	ML-DSA
Replay	Fresh session state	Transaction-level controls
Network compromise	SDN policy enforcement	Segmentation and monitoring
Misconfiguration	Crypto-Agility Manager	Audit and change control
Administrative abuse	Role separation	Centralized logging
Availability attack	Redundancy	Failover and monitoring

4.4 Mathematical Model

The framework can be represented through a set of functions describing authentication, key establishment, encryption, policy selection, and session composition. The equations do not constitute a new cryptographic primitive. Their purpose is to express the interaction of the existing security components in a formal and understandable manner.

4.4.1 Authentication Model

Auth(U, D, σ) = 1 if (U ∧ D ∧ VerifyML-DSA(pk, σ, m)) is true; Auth(U, D, σ) = 0 otherwise,
where U represents successful user-credential verification, D represents device-certificate verification, σ represents the ML-DSA signature, pk is the relevant public key, and m is the authentication message. Authentication succeeds only when the required conditions are satisfied. This logical structure prevents a valid signature from being treated as sufficient if the associated user or device checks fail.

4.4.2 Key Establishment Model

$$(K, C) = \text{Encaps}(pk)\ K = \text{Decaps}(sk, C)$$

Here pk is the public key, sk is the private key, C is the encapsulated ciphertext, and K is the shared session secret. The important security property is that both communication endpoints derive the same session secret without transmitting the secret directly as plaintext.

4.4.3 Secure Data Encryption

$$C = E_k(M)\ M = D_k(C)$$

M denotes the plaintext message, C denotes ciphertext, K denotes the session key, E is the encryption operation, and D is the decryption operation. AES-256-GCM provides authenticated encryption, meaning confidentiality and integrity protection are provided together for the protected data. In a transaction system, additional application-level validation remains necessary.

4.4.4 Crypto-Agility Model

A = f(P, S)
P represents the cryptographic policy, S represents the applicable security standard or security context, and A represents the selected cryptographic algorithm or algorithm profile. The expression captures the central idea that algorithm selection is policy-driven. In practice, the function can consider service type, security classification, interoperability, algorithm lifecycle state, regulatory requirements, and migration stage.

4.4.5 Secure Session Model

SS = Auth ∘ ML-KEM ∘ TLS 1.3 ∘ AES-256-GCM
The composition represents the complete secure-session process. Auth provides identity verification through the proposed MFA and ML-DSA combination. ML-KEM provides quantum-resistant key establishment. TLS
1.3 provides secure transport. AES-256-GCM protects the banking payload. The order is conceptual rather than an implementation prescription for every protocol stack.

4.4.6 Security Success Condition

Here Auth indicates successful identity verification, KeyOK indicates successful ML-KEM key establishment, ChannelOK indicates that the secure communication channel is valid, PolicyOK indicates that the cryptographic configuration complies with current policy, and NetworkOK indicates that the SDN security policy permits the required communication. A session should proceed only when all required

conditions are satisfied.

4.4.7 Migration State Model

A Legacy service uses the existing approved classical profile. A Transitional service supports coexistence and interoperability during migration. A Quantum-Resistant service uses the target post-quantum profile. The Crypto-Agility Manager can use this state to determine which policy is applied to each service.

4.5 Implementation Requirements

The original experimental setup specifies Ubuntu 22.04 LTS, an Intel Xeon 64-bit 8-core processor, 32 GB DDR4 memory, ML-KEM, ML-DSA, TLS 1.3, AES-256-GCM, crypto-agility management, and Software-Defined Networking using OpenFlow v1.3. The expanded implementation model retains these choices as a baseline for evaluating the architecture.

4.5.1 Cryptographic Services

A prototype should expose ML-KEM and ML-DSA through well-defined service interfaces. The application should request a security operation rather than directly manipulating private cryptographic material. This approach allows the Crypto-Agility Manager to select an approved implementation and makes later algorithm changes less disruptive.

4.5.2 Key Protection

Private keys and long-lived credentials require strong protection. In a production banking environment, cryptographic key material would normally be managed using dedicated key-management infrastructure or hardware-backed mechanisms appropriate to the institution. Access to private keys should be restricted, audited, and separated from ordinary application privileges.

4.5.3 Transport Layer

TLS 1.3 remains the transport framework in the proposed architecture. The post-quantum component is integrated at the key-establishment stage rather than replacing TLS as a whole. The implementation must ensure that protocol negotiation does not accidentally downgrade a session to an unapproved security configuration.

4.5.4 Payload Protection

AES-256-GCM is used for authenticated encryption of application data. Banking transactions can generate substantial traffic, so symmetric encryption is appropriate for protecting payloads after a session key has been established. Correct nonce management and authenticated decryption are essential implementation requirements.

4.5.5 SDN Deployment

The SDN layer can be organized around a logically centralized controller with protected communication to network devices. Banking services can be separated into security zones, such as external access, authentication, application services, transaction processing, databases, management, and monitoring. Policies should specify permitted communication paths rather than relying on broad network reachability.

4.5.6 Monitoring And Logging

A production-oriented implementation requires security telemetry. Events such as authentication failures, unusual key-establishment errors, policy changes, certificate failures, abnormal network flows, and repeated transaction anomalies should be recorded. Logs should support both real-time response and later investigation. Because the framework is intended for banking, auditability is a core operational requirement.

4.5.7 Error Handling

Cryptographic failures should be handled explicitly. A failed key-establishment operation should not silently fall back to an insecure mechanism. Similarly, an invalid signature, expired certificate, or policy mismatch should result in a controlled failure path. Error responses should reveal enough information for administrators while avoiding disclosure of sensitive internal security details to attackers.

4.5.8 API And Service Integration

Modern banking systems commonly use service-oriented or API-driven architectures. The security framework can therefore be exposed through internal security services. Each service can request

authentication validation, session establishment, cryptographic policy evaluation, or network authorization. This modular design makes the framework applicable across several customer channels without duplicating cryptographic logic.

4.5.9 Prototype Software Stack

Table 3 summarizes the baseline technologies used across the prototype layers and their respective purposes, including post-quantum key establishment, digital signatures, secure communication, network control, and cryptographic policy management.

Table 3. *Prototype Software Stack for the Proposed Quantum-Resistant Security Framework*

Layer	Baseline Technology	Purpose
Operating system	Ubuntu 22.04 LTS	Prototype execution environment
Key establishment	ML-KEM / FIPS 203	Quantum-resistant shared secret
Digital signature	ML-DSA / FIPS 204	Authentication and integrity
Transport	TLS 1.3	Secure network communication
Payload	AES-256-GCM	Confidentiality and integrity
Network control	SDN / OpenFlow v1.3	Centralized traffic policy
Policy	Crypto-Agility Manager	Algorithm lifecycle and migration

4.6 Experimental Setup And Methodology

The evaluation follows the comparative approach described in the original paper. The proposed framework is compared conceptually with a conventional banking security architecture based on RSA/ECC-oriented public-key mechanisms. The objective is to assess quantum resistance, secure key establishment, authentication, crypto-agility, and long-term security rather than to claim measured production-scale performance.

4.6.1 Experimental Parameters

Table 4 presents the cryptographic mechanisms, networking technologies, hardware and software environment, and evaluation approach defined for assessing the proposed quantum-resistant security framework.

Table 4. *Experimental Parameters and Baseline Configuration*

Parameter	Definition
Key Establishment	ML-KEM (FIPS 203)
Digital Signature	ML-DSA (FIPS 204)
Secure Messaging	TLS 1.3
Data Encryption	AES-256-GCM
Crypto-Agility	Management of cryptographic policies
Networking	Software-Defined Networking (OpenFlow v1.3)
Operating System	Ubuntu 22.04 LTS
Processor	Intel Xeon, 64-bit, 8-core
Memory	32 GB DDR4
Evaluation Approach	Comparative security analysis

4.6.2 Evaluation Dimensions

- Quantum resistance: whether the public-key security assumption is designed for the post-quantum threat model.
- Authentication: whether the architecture supports digital identity verification using a post-quantum signature mechanism.
- Key establishment: whether a post-quantum KEM is incorporated into session creation.
- Communication security: whether secure transport and authenticated encryption remain part of the architecture.
- Crypto-agility: whether algorithms can be changed through policy rather than widespread application modification.
- Network control: whether security policies can be centralized and dynamically enforced.
- Long-term security: whether the design addresses the need to protect information whose confidentiality extends into the future.

4.6.3 Comparison Baseline

The classical baseline uses RSA or ECDH for key establishment and RSA or ECDSA for digital signatures,

with TLS 1.3 and AES-256 as the communication and data-protection layers. The proposed framework substitutes ML-KEM for the key-establishment function and ML-DSA for the digital-signature function while adding explicit crypto-agility and SDN security management.

4.6.4 Experimental Scenarios

Scenario A evaluates a normal authenticated banking session. A customer is successfully authenticated, a secure session is created, and a transaction message is exchanged. The expected outcome is successful establishment of the secure session and protected transaction communication.
Scenario B evaluates an invalid authentication event. One of the required authentication conditions is intentionally unsuccessful. The expected outcome is that the secure banking session is not established. This demonstrates that key establishment is dependent on successful authentication rather than being available to any unauthenticated request.

Scenario C evaluates a cryptographic-policy transition. A service initially operates under one approved profile, and the Crypto-Agility Manager changes its policy to a different approved profile. The expected outcome is that new sessions use the updated policy without requiring a redesign of the banking application.

Scenario D evaluates a network-policy event. A suspicious communication path is detected, and the SDN controller changes the relevant network policy. The expected outcome is controlled isolation or restriction of the affected flow while legitimate services remain available where permitted.

Scenario E evaluates migration coexistence. A legacy service and a post-quantum-ready service operate simultaneously. The Crypto-Agility Manager determines the permitted security profile for each service. This scenario reflects the staged transition required in a large financial institution.

4.6.5 Measurement Considerations

A full implementation study should measure key-establishment latency, signature-generation latency, signature-verification latency, handshake size, CPU utilization, memory usage, transaction throughput, network bandwidth, and failure rates. The original paper focuses primarily on comparative security analysis; these measurements are identified here as necessary extensions for a production-oriented evaluation.

4.6.6 Experimental Validity

The proposed evaluation should distinguish between architectural claims and measured claims. A framework diagram can demonstrate how components interact, but it cannot by itself establish production performance. Performance conclusions should therefore be supported by repeatable experiments, multiple workloads, controlled hardware conditions, and clearly defined measurement procedures.

5. Results And Comparative Review

5.1 Classical Versus Quantum-Resistant Security

Table 5 compares classical RSA/ECC-based security with the proposed framework in terms of quantum resistance, cryptographic foundations, key establishment, digital authentication, standards alignment, crypto-agility, network policy management, and long-term security preparedness.

Table 5. *Classical Versus Quantum-Resistant Security Comparison*

Security Feature	RSA / ECC-Oriented Baseline	Proposed Framework
Quantum resistance	No	Designed for post-quantum threat model
Security basis	Integer factorization / discrete logarithm	Module-lattice assumptions
Key establishment	RSA / ECDH	ML-KEM
Digital authentication	RSA / ECDSA	ML-DSA
NIST PQ standard alignment	Not a PQ mechanism	FIPS 203 / FIPS 204
Crypto-agility	Limited when embedded in applications	Explicitly supported
Network policy	Distributed/traditional management	SDN-based centralized control
Long-term protection	Limited against future quantum threat	Improved preparation for quantum threat

The comparison shows that the primary difference is the security assumption used by the public-key layer. RSA and ECC are designed around mathematical problems that are vulnerable to Shor's algorithm on sufficiently capable quantum computers. ML-KEM and ML-DSA are designed within the post-quantum cryptography framework and are intended to resist known quantum algorithmic threats of the type

considered in this study.

5.2 Security Architecture Comparison

The proposed framework also differs at the architectural level. Instead of treating cryptography as a fixed library choice, it introduces a Crypto-Agility Manager. Instead of treating network configuration as an unrelated infrastructure concern, it incorporates an SDN Security Controller. These additions are significant because quantum migration is not solely an algorithm-selection problem; it is also an operational and network-management problem.

5.3 Communication Security

Table 6 compares the classical and proposed frameworks in terms of transport security, key establishment, payload encryption, digital signatures, cryptographic policy management, and network control.

Table 6. *Communication Security Comparison Between Classical and Proposed Frameworks*

Criterion	Classical Framework	Proposed Framework
Transport	TLS 1.3	TLS 1.3
Key establishment	RSA / ECDH	ML-KEM
Payload encryption	AES-256	AES-256-GCM
Digital signature	RSA / ECDSA	ML-DSA
Cryptographic policy	Often distributed	Centralized through crypto-agility
Network policy	Traditional controls	SDN-assisted control

The proposed approach does not replace every existing security layer. Instead, it retains mature communication and symmetric encryption components while strengthening the public-key functions that face the clearest quantum risk. This layered approach reduces unnecessary architectural change. At the same time, larger post-quantum keys, ciphertexts, and signatures can create bandwidth, storage, and processing overhead that must be measured before deployment.

5.4 Performance Overview

The original paper characterizes the framework as a promising and scalable security architecture but does not present a full production benchmark. Therefore, performance should be interpreted as an evaluation requirement rather than a proven numerical outcome. A complete study should compare the classical and post-quantum configurations under equivalent workloads and report the additional communication and computational costs introduced by the post-quantum mechanisms.

5.5 Expected Performance Trade-Offs

- Post-quantum public keys and ciphertexts may increase network traffic.
- Digital signatures may require more storage than classical signatures.
- Handshake processing may increase latency depending on implementation and workload.
- Centralized policy management can reduce configuration inconsistency but adds controller dependencies.
- Crypto-agility can reduce future migration effort but requires additional policy and lifecycle infrastructure.
- Symmetric payload encryption remains suitable for high-volume data once session keys are established.

5.6 Security Analysis

5.6.1 Confidentiality

Confidentiality is strengthened through the combination of quantum-resistant key establishment and authenticated encryption. ML-KEM is intended to reduce dependence on classical public-key assumptions during session creation. AES-256-GCM then protects the application payload. The architecture is particularly motivated by long-term confidentiality and the HNDL threat model.

5.6.2 Integrity

Integrity is addressed at multiple layers. ML-DSA supports digital authentication, TLS 1.3 protects transport communication, and AES-256-GCM provides authenticated encryption for application payloads. Layered integrity checks are valuable because an attacker may target different stages of the

communication process.

5.6.3 Authenticity

Authenticity depends on the combined authentication process. User credentials establish identity information, device verification provides an additional context signal, and ML-DSA supports signature verification. The architecture therefore avoids relying on a single authentication factor. Nevertheless, endpoint compromise and stolen credentials remain broader risks that require additional controls.

5.6.4 Availability

Availability is supported by centralized policy management only if the management infrastructure is itself resilient. The SDN controller should be deployed with redundancy, controlled failover, and protected management access. Authentication and cryptographic services should also avoid becoming single points of failure. A quantum-resistant system that prevents legitimate users from accessing banking services would not satisfy the practical requirements of a financial institution.

5.6.5 Forward Security Considerations

The framework's use of fresh session keys is intended to reduce the impact of compromise of one session on unrelated sessions. A complete implementation should define key lifetimes and rotation rules and should ensure that session secrets are not reused beyond their intended scope. Forward-security properties also depend on the complete protocol implementation, not solely on the choice of ML-KEM.

5.6.6 Zero-Trust Alignment

The architecture is compatible with a zero-trust style of operation because authentication and authorization can be evaluated before access to each protected service. The SDN layer can enforce service-to-service restrictions, while the authentication layer establishes the identity context. The design should not assume that an internal network is automatically trustworthy.

5.6.7 Policy Enforcement

The Crypto-Agility Manager and SDN Security Controller provide two complementary forms of policy enforcement. The first controls cryptographic behavior; the second controls network behavior. Separating these responsibilities makes the architecture easier to reason about. A cryptographic policy may specify which algorithms are permitted, while a network policy specifies which endpoints may communicate.

5.6.8 Auditability

Financial systems require strong traceability. The framework can generate audit records for authentication attempts, signature-verification results, key-establishment events, policy changes, network-policy changes, and session termination. Audit records should be protected from unauthorized modification and should be correlated with transaction identifiers where appropriate.

5.6.9 Security Failure Modes

Table 7 summarizes potential authentication, cryptographic, policy, and network failures and specifies the corresponding responses to maintain secure session establishment, enforce access restrictions, and support reliable banking operations.

Table 7. *Security Failure Modes and Expected Responses in the Proposed Framework*

Failure	Expected Response
Invalid credentials	Reject authentication
Invalid ML-DSA signature	Reject authentication/session.
Device verification failure	Require additional validation or reject
ML-KEM failure	Do not establish the secure session.
Policy mismatch	Select approved profile or reject
Unauthorized network flow	Block/restrict through SDN policy
Controller failure	Fail over according to resilient deployment.
Expired cryptographic material	Rotate/revoke and prevent unsafe use.

5.7 Crypto-Agility And Migration Strategy

The transition to post-quantum cryptography should be treated as a lifecycle rather than a one-time replacement project. The first stage is discovery: an institution must identify where public-key cryptography is used, which systems depend on certificates or key exchange, and which information requires long-term confidentiality. The second stage is classification: services are grouped according to

business criticality, data sensitivity, compatibility, and migration urgency.

5.7.1 Phase 1 – Cryptographic Inventory

A cryptographic inventory should record algorithms, key sizes, certificate dependencies, protocol versions, libraries, hardware dependencies, and data-retention requirements. The inventory provides the baseline required for migration planning. Without it, an institution may replace visible components while leaving hidden classical dependencies in APIs, legacy applications, backup systems, or internal communication channels.

5.7.2 Phase 2 – Risk Prioritization

Systems containing information with long confidentiality periods should receive early attention. Critical external interfaces and high-value authentication services should also be prioritized because their compromise can affect large portions of the banking ecosystem. Risk ranking should combine data sensitivity, exposure, business criticality, and difficulty of future migration.

5.7.3 Phase 3 – Pilot Deployment

A limited pilot can introduce ML-KEM and ML-DSA into a controlled environment. The pilot should evaluate interoperability, latency, message size, operational logging, certificate handling, and failure behavior. It should also verify that legacy services can continue to operate without unexpected disruptions.

5.7.4 Phase 4 – Transitional Coexistence

During coexistence, the Crypto-Agility Manager determines the permitted profile for each service. New applications can be designed with post-quantum support, while selected legacy applications remain on existing mechanisms until replacement or upgrade. Policy should prevent accidental use of unapproved combinations.

5.7.5 Phase 5 – Broader Deployment

After successful pilot and coexistence testing, post-quantum security can be expanded to additional banking services. Deployment should be staged by service criticality. Monitoring should compare security events and operational metrics before and after each migration step. Rollback procedures should be defined for implementation defects, interoperability failures, or unexpected operational impact.

5.7.6 Phase 6 – Deprecation Of Legacy Mechanisms

Once post-quantum alternatives are widely supported, the institution can reduce and eventually retire selected classical public-key mechanisms according to risk and policy. Deprecation should be controlled rather than abrupt. The Crypto-Agility Manager can mark legacy algorithms as restricted, prohibit their use for new sessions, and later remove them from approved profiles.

5.7.7 Migration Governance

Migration requires cooperation among security teams, network teams, application teams, infrastructure administrators, risk managers, compliance personnel, and business owners. A governance process should define ownership, approval requirements, testing standards, change windows, incident response, and evidence collection. The framework therefore supports not only technical migration but also organizational coordination.

5.7.8 Migration Checklist

- Identify all public-key cryptographic dependencies.
- Classify data according to confidentiality lifetime and business sensitivity.
- Map external and internal communication protocols.
- Select approved post-quantum algorithms and implementation profiles.
- Introduce a crypto-agility abstraction layer.
- Test interoperability with legacy systems.
- Measure performance and message-size overhead.
- Deploy monitoring and audit controls.
- Introduce staged production migration.
- Restrict and later retire obsolete cryptographic profiles.

5.8 Scalability and Deployment Considerations

Digital banking infrastructure must support large numbers of concurrent sessions and high transaction volumes. A security framework should therefore scale horizontally where possible. Authentication services, cryptographic gateways, policy services, and transaction services can be distributed across multiple instances. The SDN controller should be deployed using an architecture that avoids a single point of failure.

5.8.1 Cloud And Distributed Infrastructure

The proposed architecture is compatible with cloud and distributed infrastructure because the security functions can be exposed as controlled services. However, distribution increases the importance of service identity, secure inter-service communication, key management, and consistent policy propagation. Crypto-agility becomes particularly valuable when different services operate on different technology stacks.

5.8.2 Database And Data-Protection Considerations

The framework primarily addresses communication and cryptographic session security. Databases containing financial information require additional protection at rest, access control, backup protection, monitoring, and retention management. A complete deployment should therefore integrate the proposed communication framework with existing data-security controls rather than treating it as a complete database-security solution.

5.8.3 High-Volume Transactions

Transaction workloads may be sensitive to handshake latency and cryptographic overhead. One practical design is to minimize expensive public-key operations while maintaining secure session lifetimes that are appropriate to the application. Once a secure session is established, AES-256-GCM can protect multiple application messages efficiently. Exact session policies should be determined by security requirements and workload measurements.

5.8.4 Multi-Channel Banking

The same security architecture can support mobile applications, web applications, and ATM-related services, but the device and network context differs across channels. The Authentication Module can therefore use channel-specific device validation while the underlying cryptographic policy remains centralized. This provides consistency without forcing every channel to use an identical user experience.

5.8.5 Resilience And Disaster Recovery

Security services must be included in disaster-recovery planning. Cryptographic policy databases, key-management dependencies, authentication services, and SDN controllers require backup and recovery procedures. Recovery processes should preserve security properties and prevent the accidental restoration of deprecated cryptographic settings.

5.8.6 Interoperability

Interoperability is a major challenge because banking ecosystems involve internal applications, external payment networks, vendors, cloud services, and partner institutions. The migration framework should therefore define supported security profiles and transition mechanisms. Crypto-agility allows the institution to maintain more than one approved profile when necessary, but coexistence should be temporary and tightly controlled.

5.8.7 Regulatory And Compliance Considerations

Financial institutions operate under security, privacy, audit, and operational requirements. The proposed framework does not replace those requirements; instead, it provides technical mechanisms that can support them. Cryptographic policy, access control, logging, key lifecycle management, and network segmentation should be mapped to the organization's applicable compliance obligations.

5.8.8 Cost Considerations

Migration costs include software changes, infrastructure upgrades, testing, monitoring, staff training, cryptographic inventory, and possible hardware changes. Post-quantum mechanisms can also increase bandwidth and storage requirements because of larger keys, ciphertexts, or signatures. These costs should be compared with the potential cost of delaying migration for systems containing long-lived sensitive information.

5.8.9 Deployment Readiness Levels

Table 8 outlines six deployment readiness levels, from the absence of a cryptographic inventory to the adoption of post-quantum mechanisms as the default for eligible services and the gradual deprecation of legacy cryptographic methods.

Table 8. *Deployment Readiness Levels for the Proposed Quantum-Resistant Security Framework*

Level	Description
Level 0	No cryptographic inventory or migration policy
Level 1	Inventory completed and high-risk dependencies identified.
Level 2	Crypto-agility layer and laboratory prototype available
Level 3	Controlled pilot using ML-KEM / ML-DSA
Level 4	Transitional production deployment with monitoring
Level 5	Post-quantum default for eligible services and legacy deprecation

6. Limitations and Discussion

The proposed framework provides an integrated architectural model, but several limitations must be recognized. First, the paper does not represent a full-scale production banking deployment. The experimental environment described in the original paper is a prototype-oriented setup, and the comparative evaluation is primarily architectural and security-focused. Therefore, numerical performance claims should not be inferred where direct measurements are not provided.

Post-Quantum Overhead

Post-quantum mechanisms can require larger public keys, ciphertexts, or signatures than some classical alternatives. In high-volume financial networks, this can increase network bandwidth, storage, memory usage, and potentially handshake latency. The impact will depend on implementation quality, hardware, protocol integration, and workload characteristics.

6.1 Implementation Risk

Using a standardized algorithm does not automatically guarantee a secure implementation. Software vulnerabilities, incorrect parameter handling, poor key storage, unsafe random-number generation, protocol errors, and configuration mistakes can undermine the security architecture. Secure coding, implementation validation, testing, patch management, and operational monitoring remain essential.

6.2 SDN Controller Dependency

Centralized control improves policy consistency but also creates a high-value management component. If the controller is compromised, an attacker could potentially manipulate network behavior. Redundant controllers, strong authentication, protected management channels, role separation, and change auditing are therefore necessary.

6.3 Legacy System Constraints

Older banking applications may not support post-quantum protocols or may rely on fixed cryptographic libraries. Such systems may require vendor upgrades, gateway-based migration, or eventual replacement. Crypto-agility can reduce application changes, but it cannot eliminate fundamental hardware or protocol incompatibility.

6.4 Discussion Of Practical Value

The strongest practical value of the framework is its combination of security modernization and migration readiness. A bank does not need to wait for a large-scale quantum computer before preparing. By cataloguing cryptographic dependencies, introducing an abstraction layer, testing post-quantum algorithms, and applying centralized policy, an institution can reduce future migration risk.

6.5 Relation To The Original Framework

The expanded paper preserves the original framework's principal design: ML-KEM for key establishment, ML-DSA for authentication, TLS 1.3 and AES-256-GCM for protected communication, crypto-agility for cryptographic transition, and SDN for centralized network security management. The additional material develops the same points in greater depth through threat modeling, mathematical formulation, migration stages, deployment requirements, and operational considerations.

6.6 Future Research Directions

- Benchmark ML-KEM and ML-DSA under realistic banking transaction workloads.
- Evaluate hybrid classical/post-quantum handshake configurations during migration.
- Study the effect of larger post-quantum messages on high-volume financial networks.
- Evaluate SLH-DSA and other standardized or emerging post-quantum signature options.
- Investigate Zero Trust Architecture integration across banking service boundaries.
- Evaluate quantum key distribution for selected high-value inter-bank communication paths.
- Study FPGA and ASIC acceleration for post-quantum cryptographic operations.
- Develop automated cryptographic inventory and policy-validation tools.
- Evaluate multi-controller SDN resilience for large distributed banking environments.
- Create formal verification and security-testing methods for the complete session workflow.

7. Conclusion And Future Work

The proposed Quantum-Resistant Security Framework provides a practical approach for improving the security of digital banking systems against the possible risks associated with future quantum computing. The framework combines ML-KEM for secure key establishment, ML-DSA for digital authentication, TLS 1.3 and AES-256-GCM for protected communication, crypto-agility for flexible cryptographic management, and SDN for centralized network security. Instead of depending on a single security mechanism, the framework connects these technologies into a layered security architecture.

One of the main advantages of the proposed approach is its focus on **long-term security and gradual migration**. Traditional cryptographic methods may continue to be useful in existing banking environments, but financial institutions need to prepare for a future in which quantum computers could threaten current public-key security mechanisms. The use of ML-KEM and ML-DSA provides a foundation for moving important cryptographic operations toward post-quantum protection. At the same time, the Crypto-Agility Manager allows security administrators to change cryptographic configurations as standards and security requirements evolve.

The framework also considers the operational requirements of modern banking networks. The integration of an SDN Security Controller allows security policies and network communication rules to be managed centrally. This can make it easier to control communication between different banking services and respond to changes in the security environment. The combination of centralized network management and flexible cryptographic policies makes the framework suitable for distributed digital banking environments.

However, the proposed architecture should be considered a **security framework and research model rather than a completed production banking implementation**. Post-quantum algorithms may introduce additional communication and storage requirements because of their larger keys, ciphertexts, and signatures. Real-world deployment would therefore require detailed performance testing, interoperability testing, security validation, and compatibility checks with existing banking infrastructure.

7.1 Future Work

Future research can focus on implementing the proposed framework in a controlled banking environment and measuring its practical performance. Important measurements can include authentication time, ML-KEM key-establishment latency, ML-DSA signature processing time, network overhead, memory consumption, transaction throughput, and scalability under multiple simultaneous users.

Further work can also investigate the use of other post-quantum algorithms, including **SLH-DSA**, and compare their suitability for different banking applications. Hybrid cryptographic approaches that combine classical and post-quantum mechanisms during the migration period can also be studied.

Another important direction is the integration of **Zero Trust Architecture** with the proposed framework. This could provide continuous verification of users, devices, and services rather than relying only on initial authentication. Research can also examine the use of hardware acceleration through FPGA or ASIC technologies to reduce the computational overhead of post-quantum cryptographic operations.

Finally, future studies can develop a complete prototype and evaluate it using realistic banking workloads. Such research would help determine whether the proposed architecture can provide quantum-resistant protection while maintaining the performance, availability, scalability, and operational requirements expected from modern digital banking systems.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

Refernces

[1] National Institute of Standards and Technology (NIST), "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)," Gaithersburg, MD, USA, Aug. 2024.

[2] National Institute of Standards and Technology (NIST), "FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)," Gaithersburg, MD, USA, Aug. 2024.

[3] National Institute of Standards and Technology (NIST), "FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)," Gaithersburg, MD, USA, Aug. 2024.

[4] A. Alnahawi, A. Alshamrani, and M. Alzahrani, "On the State of Crypto-Agility," IEEE Access, vol. 10, pp. 111313–111330, 2022.

[5] M. Sosnowski, F. Wiedner, E. Hauser, L. Steger, D. Schoinianakis, S. Gallenmüller, and G. Carle, "The Performance of Post-Quantum TLS 1.3," Proceedings of the ACM CoNEXT Companion, 2023.

[6] X. Guo, H. Xian, T. Feng, Y. Jiang, D. Zhang, and J. Fang, "An Intelligent Zero Trust Secure Framework for Software Defined Networking," PeerJ Computer Science, vol. 9, e1674, 2023.

[7] CISA, NSA, and NIST, "Preparing for a Post-Quantum Cryptography World: Guidance for Organizations," 2023.

[8] Financial Services Information Sharing and Analysis Center (FS-ISAC), "Building Cryptographic Agility in Financial Services," White Paper, 2024.

[9] A. Bishwas and J. Sen, "A Strategic Roadmap for Quantum-Resistant Industry Security," IEEE Access, vol. 12, 2024.

[10] S. Sadeghi, V. Chouhan, M. Aldarwbi, A. Ghorbani, A. Chow, and R. Burko, "Securing Financial Sector Applications in the Quantum Era: A Comprehensive Evaluation of NIST's Recommended Algorithms Through Use-Case Analysis," Expert Systems with Applications, vol. 288, Art. no. 128243, 2025.

[11] T. Le, H. Do, T. Dinh, and M. Pham, "Enterprise Quantum-Safe Readiness: A SWOT Analysis," 2025.

[12] T. Muzenda and M. Ndlovu, "Post-Quantum Cryptography Readiness in the Financial Sector: A Systematic Review," 2026.

[13] J. Bos et al., "CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based Key Encapsulation Mechanism," IEEE European Symposium on Security and Privacy, 2018, pp. 353–367.

[14] L. Ducas et al., "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2018, no. 1, pp. 238–268, 2018.

[15] NIST, "Special Publication 800-207: Zero Trust Architecture," 2020.

[16] Frontiers in Physics, "Design and Implementation of an Authenticated Post-Quantum Session Protocol Using ML-KEM (Kyber), ML-DSA (Dilithium), and AES-256-GCM," vol. 13, 2025.

[17] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

[18] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM Journal on Computing, vol. 26, no. 5, pp. 1484–1509, 1997.

[19] Open Quantum Safe (OQS) Project, "Open-Source Tools for Quantum-Resistant Cryptography."

[20] ETSI, "Quantum-Safe Cryptography and Security: An Introduction, Benefits, Enablers and Challenges," ETSI White Paper No. 8, 2015.

[21] R. T. Gadhave et al., "SmartHeart: A Cloud and Machine Learning Framework for Early Cardiovascular Disease Prediction," International Journal of Environmental Sciences, vol. 11, no. 5, 2025.

[22] L. I. Sakri et al., "AI-Enabled Transformation of Online Learning through Personalization," Journal of Engineering Education Transformations, vol. 39, Special Issue 1, pp. 26–33, 2025.

[23] S. Marigoudar and B. M. V., "Internet of Things (IoT) Security Enhancement Using Blockchain Technology," International Journal of Engineering and Techniques, vol. 11, no. 4, 2025.

[24] S. Marigoudar and Harshita, "Predicting Student Engagement in Online Courses Using AI," International Journal of Scientific Research in Engineering and Management, vol. 9, no. 11, 2025.

[25] S. Marigoudar and S. Attar, "Intelligent Video Analytics for Abnormal Event Detection," International Journal of Scientific Research in Engineering and Management, vol. 8, no. 9, 2024.

[26] S. Marigoudar, "Machine Learning Regression Techniques to Detect the Disease in Cotton Leaf," TIJER – International Research Journal, vol. 10, no. 7, pp. 740–748, 2023.