Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quantum-QRIS: A Quantum-Safe Trust Architecture for Merchant-Bound Static and Dynamic QR Payments

H.A Danang Rimbawa

hadr71@gmail.com

Indonesia Defense University

Danny Setyowati

Indonesia Defense University

Achmad Farid Wadjdi

National Research and Innovation Agency

Article

Keywords:

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11637831/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Quantum-QRIS: A Quantum-Safe Trust Architecture for Merchant-Bound Static and Dynamic QR Payments

H.A. Danang Rimbawa · Danny Setyowati · Achmad Farid Wadjdi

Abstract QR-based payment systems offer low-cost deployment and broad interoperability, but visual QR artifacts do not inherently prove that the encoded recipient is the intended merchant or that displayed payment evidence reflects an authoritative transaction. This creates a trust gap that persists even when the payment rail itself functions correctly, while financial infrastructures must also prepare for migration to quantum-resistant cryptography. This study proposes and experimentally evaluates Quantum-QRIS, a merchant-bound post-quantum trust architecture for static and dynamic QR payments. Its novelty lies in decoupling compact customer-facing QR artifacts from online trust resolution so that merchant identity, transaction freshness, lifecycle state, post-quantum authentication, quantum-safe key establishment, payment completion, and fraud evidence are jointly resolved through a reusable Quantum-Safe Security as a Service (QSSaaS) backend. The architecture integrates ML-DSA-65, ML-KEM-768, hybrid QRNG/OS-CSPRNG freshness, replay and expiry controls, merchant lifecycle validation, and authoritative payment-state verification. A design-science methodology was used to implement and evaluate the prototype through functional/security validation and three benchmark runs. In the main same-network benchmark (n=300), all core signing, verification, nonce, merchant, static QR, HTTP static, and EDC creation operations achieved 100% success. Mean service-path latency was 515.79 ms for ML-DSA signing, 497.85 ms for verification, and 2366.76 ms for EDC creation. Full-payment completion reached 76% for static payments and 69% for EDC payments. The results show that quantum-safe QR-payment security is a systems problem requiring cryptographic assurance, merchant and transaction-state binding, lifecycle enforcement, and explicit treatment of physical-context limitations.

Keywords QR payment security · post-quantum cryptography · ML-DSA · ML-KEM · merchant authentication · QSSaaS

H.A. Danang Rimbawa (Corresponding Author)
Cyber Defense Engineering, Faculty of Defense Science and Technology, Republic of Indonesia Defense University, Bogor, Indonesia
E-mail: hadr71@gmail.com

Danny Setyowati
Cyber Defense Engineering, Faculty of Defense Science and Technology, Republic of Indonesia Defense University, Bogor, Indonesia

Achmad Farid Wadjdi
Research Center for Artificial Intelligence and Cybersecurity, BRIN, Bandung, Indonesia

1 Introduction

QR-based payments are now a foundational mechanism for low-cost digital commerce because they reduce merchant hardware requirements while maintaining interoperability across mobile payment applications. This same simplicity produces a trust asymmetry: users see a compact visual symbol, but the symbol itself does not independently prove that the encoded recipient is the merchant physically standing in front of them. QRIS is the Indonesian interoperable QR-payment standard, while merchant-presented QR operation is aligned with established payment-QR and QR-symbol specifications [1], [2], [3]. Recent systematic work on QR security similarly shows that QR attacks exploit hidden destinations, user trust, tampering, phishing, and the limited ability of users to inspect the code before interaction [4].

The Indonesian QRIS ecosystem provides a particularly relevant setting because static merchant-presented

QR codes are widely used by micro and small merchants. The security problem is not hypothetical. Bank Indonesia has explicitly warned the public and merchants about QRIS misuse [5]. In April 2023, Indonesian police reported that an attacker placed fraudulent QRIS labels at approximately 38 locations, including places of worship and public facilities, collecting more than IDR 13 million [6]. In October 2025, merchants around Telkom University in Bandung discovered that their original QRIS labels had been physically overlaid by different codes, redirecting customer payments away from the intended merchants [7], [8]. In May 2026, police in Surabaya reported a different trust failure: manipulated QRIS payment evidence was shown to a store employee to obtain cash despite the lack of authoritative merchant-side confirmation [9]. These cases expose two distinct classes of failure: merchant identity binding failure and transaction confirmation trust failure.

These incidents are important because the underlying payment rail can function exactly as designed while the economic outcome is still wrong. A substituted QR may point to a technically valid attacker-controlled payment account; similarly, a manipulated screenshot may look convincing without corresponding to a confirmed transaction. Transport encryption alone cannot solve either problem because the attack may occur before trusted backend validation or at the human decision layer.

A second, forward-looking problem is cryptographic longevity. The financial sector is beginning to plan for migration from RSA/ECC-based trust toward quantum-resistant mechanisms. The Bank for International Settlements emphasizes that quantum readiness is not a simple algorithm swap but a coordinated migration involving inventory, cryptographic agility, performance assessment, and defense in depth [10]. Recent financial-sector research also evaluates NIST-standardized PQC algorithms as practical candidates for financial applications [11], while regulatory scholarship identifies quantum resilience as an emerging requirement for digital-payment security and governance [12].

This paper argues that these present-day and future risks should be treated as one systems-security problem: payment trust must bind the intended merchant, the transaction context, freshness, cryptographic evidence, and authoritative completion status, while remaining migratable to quantum-safe primitives.

Novelty statement. Quantum-QRIS differs from prior QR-security, fraud-detection, and post-quantum financial studies by treating trust resolution as a unified payment-system function rather than as an isolated QR-authentication or cryptographic primitive problem. The proposed design keeps the customer-facing QR compact while shifting trust decisions to a reusable backend that jointly resolves merchant identity, signed transaction state, freshness, revocation, payment completion, and fraud evidence. This architecture therefore targets the gap between cryptographic authenticity and contextual payment trust, including the explicit identification of valid-but-misplaced QR as a residual physical-context attack that cryptography alone cannot solve.

1.1 Research Questions

1. RQ1: Can standardized post-quantum cryptography be integrated into static and dynamic QR payment workflows without replacing the low-friction printed-QR usability model?
2. RQ2: Can the architecture enforce merchant authenticity, transaction integrity, expiry, replay resistance, revocation, and fraud evidence in a unified workflow?
3. RQ3: What latency, tail-latency, success, and operational behavior emerge when the prototype is executed repeatedly under controlled same-network conditions?
4. RQ4: Which real-world QRIS attack classes are directly mitigated by the proposed architecture, and which still require physical-context or organizational controls?

1.2 Contributions

1. A merchant-bound trust model that formalizes the distinction between cryptographic validity and contextual payment validity, including the residual valid-but-misplaced QR problem.
2. A three-mode architecture for static printed QR, dynamic EDC/POS, and enterprise/API payments in which compact QR artifacts are decoupled from online trust resolution through a reusable QSSaaS backend.
3. A unified control plane combining ML-DSA-65 authentication, ML-KEM-768 quantum-safe key establishment, hybrid QRNG/OS-CSPRNG freshness, merchant/key lifecycle, replay/expiry state, authoritative payment confirmation, audit evidence, and fraud-memory.
4. An incident-driven security validation showing acceptance of genuine flows and expected rejection of replay, tampering, merchant mismatch, expiry, revocation, and forged/unregistered QR scenarios.

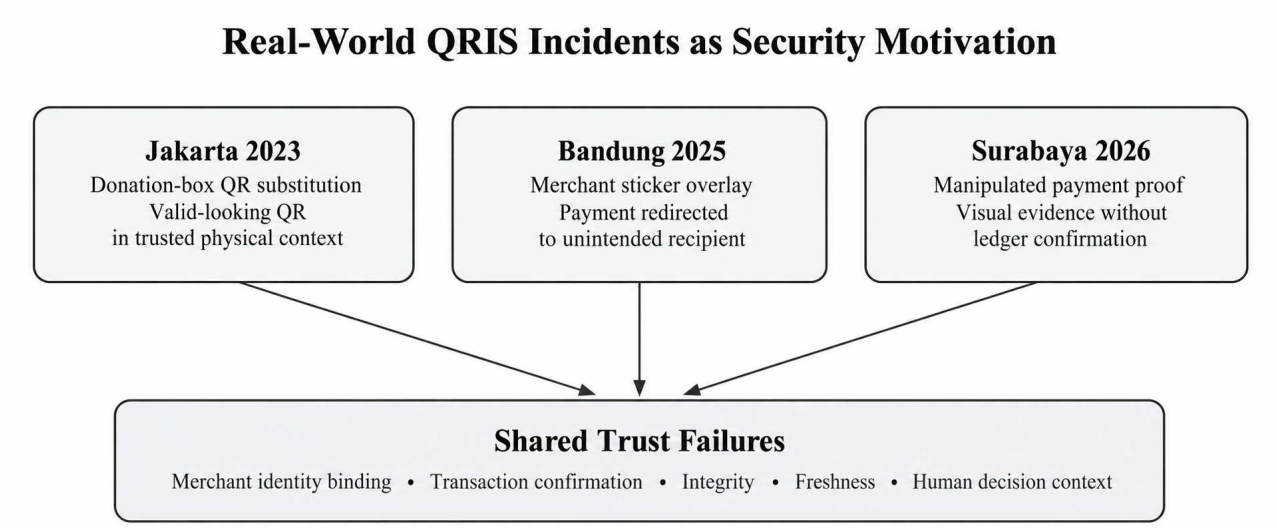

Fig. 1 Real-world QRIS incidents motivating the Quantum-QRIS trust model (AI-assisted explanatory schematic; content verified by the authors)

5. An empirical system-level evaluation across Pilot n=20, Main n=300, and Full-Payment n=100 runs that reports service-path latency, tail latency, completion, rejection causes, and cross-run consistency while explicitly separating cryptographic reliability from application/network dependencies.

2 Related Work and Research Gap

2.1 QR Security, Substitution, and Quishing

Recent QR-security research confirms that substitution, counterfeit QR artifacts, hidden redirection, phishing, and human trust are persistent concerns. Njuguna and Ndia reviewed 50 QR-security studies and found a broad attack surface spanning QR-to-QR manipulation, payment attacks, counterfeiting, information leakage, phishing, and reader/application attacks [4]. Alsuhibany proposed watermarking plus a neural-network authenticator for tamper-resistant QR verification and evaluated the approach on 5000 QR samples [13]. Yalda et al. developed B-PhishQR, a service-oriented framework combining encryption, revocation, and blockchain-inspired integrity checks for QR phishing defense [14]. Recent on-device work also demonstrates multi-stage static/dynamic QR-phishing detection in mobile WebView environments [15], while QR-based browser-in-the-browser attacks illustrate the continuing evolution of QR-mediated social engineering [16]. These studies strengthen QR authenticity and attack detection, but they do not address post-quantum merchant trust, dynamic transaction lifecycle, and payment-state confirmation in a unified fiat-payment architecture.

2.2 Fraud Detection and Transaction Context

Fraud detection in mobile payments increasingly combines behavioral modeling with cryptographic or distributed evidence. Ranganatha and Mustafa integrated deep sequence modeling with blockchain-oriented controls for mobile-transaction fraud detection [17]. This line of work is complementary to Quantum-QRIS: behavioral detection can identify anomalous patterns, but transaction signatures, merchant provenance, replay state, and key lifecycle provide stronger attribution and enforceable rejection mechanisms. Quantum-QRIS therefore uses fraud-memory as a stateful evidence layer rather than claiming that cryptography alone eliminates fraud.

2.3 Post-Quantum Security for Financial and Transaction Systems

Schöffel et al. showed that post-quantum deployments in constrained systems are shaped strongly by communication and system-level bottlenecks rather than cryptographic computation alone [18]. Hu et al. demonstrated practical post-quantum wallet constructions and emphasized the interaction between key management, signature efficiency, and transaction throughput [19]. Sadeghi et al. recently evaluated NIST-recommended PQC algorithms for financial-sector use cases, highlighting the need to match algorithm properties to operational requirements [11].

Wang et al. proposed a quantum secure digital-payment protocol based on measurement-device-independent quantum methods, demonstrating a different path toward quantum-secure payment but with infrastructure assumptions distinct from mass-market QR payments [20]. Related research on post-quantum signatures has investigated forward-secure identity-based constructions, including the work of Shaw and Dutta (2022), published in the Journal of Information Security and Applications [21]. This research addresses cryptographic authentication but does not specifically examine QR-payment merchant binding, transaction freshness, or authoritative payment-state verification.

The literature therefore supports three separate observations: QR artifacts need stronger authenticity and context verification; fraud detection benefits from richer transaction evidence; and financial infrastructures need quantum-safe cryptographic migration. The gap lies in experimentally combining these requirements while preserving static printed QR usability and supporting dynamic payments through a common backend trust service.

2.4 Research Gap Matrix

Comparative positioning. The closest prior approaches each cover only a subset of the trust chain: QR-authentication studies strengthen artifact integrity or phishing resistance but do not provide post-quantum merchant and payment-state lifecycle; post-quantum financial studies evaluate algorithms, wallets, or migration but do not address QR substitution and merchant-context binding; and behavioral fraud studies detect anomalies without providing cryptographically enforceable transaction provenance. Quantum-QRIS is differentiated by evaluating these functions as one deployable workflow. The comparison is architectural rather than a claim of algorithmic superiority: this study does not report a classical RSA/ECC baseline or claim lower primitive latency than prior PQC implementations.

2.5 Security-Property Comparison Criteria

The comparison in Table 1 is architectural, rather than a controlled performance or attack-success comparison. To make the comparative claim falsifiable, five distinct properties should be examined: (i) *recipient binding*, whether the verified identity identifies the merchant intended by the customer; (ii) *message integrity*, whether protected transaction fields can be modified without detection; (iii) *freshness and single use*, whether previously accepted or expired requests can be reused; (iv) *payment-state authenticity*, whether acceptance by the merchant is based on an authoritative transaction state rather than a visual receipt; and (v) *key and identity lifecycle*, whether retired or revoked identities cease to authorize new operations. These properties are related but not interchangeable: a cryptographically correct signature proves authenticity relative to a public key, but the association between that key and the intended recipient must be established separately. This distinction is consistent with identity-centric authorization and key-management principles [29, 28].

For this work, evidence consists of the implemented control scenarios in Table 4, the observed service-path measurements in Tables 7–11, and the incident-to-control analysis in Table 13. The first two categories demonstrate prototype behavior under the described conditions; the third is an analytical mapping, not a measured prevention rate for field fraud. Accordingly, *security coverage*, *operational completion*, and *performance advantage over alternatives* are treated as separate claims. A direct comparison with conventional QR payment controls or a controlled classical/PQC baseline remains necessary to establish incremental efficacy or overhead.

3 Threat Model and Security Requirements

The adversary may obtain physical access to displayed QR artifacts, present a technically valid attacker-controlled QR in a trusted merchant context, manipulate transaction fields, replay previously valid intents, present forged visual payment evidence, exploit weak freshness, or benefit from future quantum capabilities against classical public-key cryptography. We assume the adversary does not directly compromise the QSSaaS host or steal active private keys; key-compromise response is instead modeled through rotation and revocation.

3.1 Trust Boundaries and Security Objectives

The model separates four trust boundaries: (B1) the physical merchant location and displayed QR artifact, which an adversary may replace; (B2) the customer-facing application or scanner, which observes the displayed artifact but cannot infer physical provenance from its bytes alone; (B3) the merchant application and its local transaction state; and (B4) the QSSaaS registry, signing/key services, and transaction-verification functions, which form the trusted service boundary in

Table 1 Research gap matrix and distinctive contribution of Quantum-QRIS

Study	Primary focus	Strength	Unresolved gap	Distinctive contribution relative to this work
Schöffel et al. [18]	PQC performance in IoT/TLS	System/network bottlenecks	No QR payment, merchant binding, or fraud lifecycle	Uses service-path benchmarking in a QR-payment PoC
Hu et al. [19]	Post-quantum deterministic wallets	Quantum-safe key/signature design	Cryptocurrency wallet context; no QR merchant trust	Applies PQC to merchant/transaction trust and payment state
Wang et al. [20]	Quantum secure digital payment	Quantum-protocol authentication	Specialized quantum communication assumptions	Deployable PQC backend for conventional QR workflows
Njuguna & Ndia [4]	QR attack systematic review	Threat taxonomy and countermeasures	Review only; no integrated payment prototype	Implements and benchmarks integrated controls
Alsuhibany [13]	Tamper-proof QR + neural verification	Printed-QR authenticity	No PQC, payment lifecycle, replay, or key lifecycle	Adds PQC, merchant lifecycle, replay/expiry, payment-state validation
Sadeghi et al. [11]	PQC selection for finance	Financial use-case evaluation	No QRIS, merchant substitution, or fraud context	Evaluates standardized PQC in QR payment architecture
Yalda et al. [14]	QR phishing verification service	Revocation, encryption, integrity	No post-quantum primitives or payment-intent semantics	ML-DSA/ML-KEM + merchant/payment lifecycle
Bush [12]	Quantum computing and digital-payment regulation	Quantum risk and regulatory gaps	Conceptual/legal; no prototype benchmark	Provides implemented technical evidence and operational limitations
This work	Quantum-QRIS integrated trust architecture	Static/dynamic/API + PQC + freshness + lifecycle + fraud evidence	Prototype, not production QRIS integration	Incident-driven design plus functional/security and performance evaluation

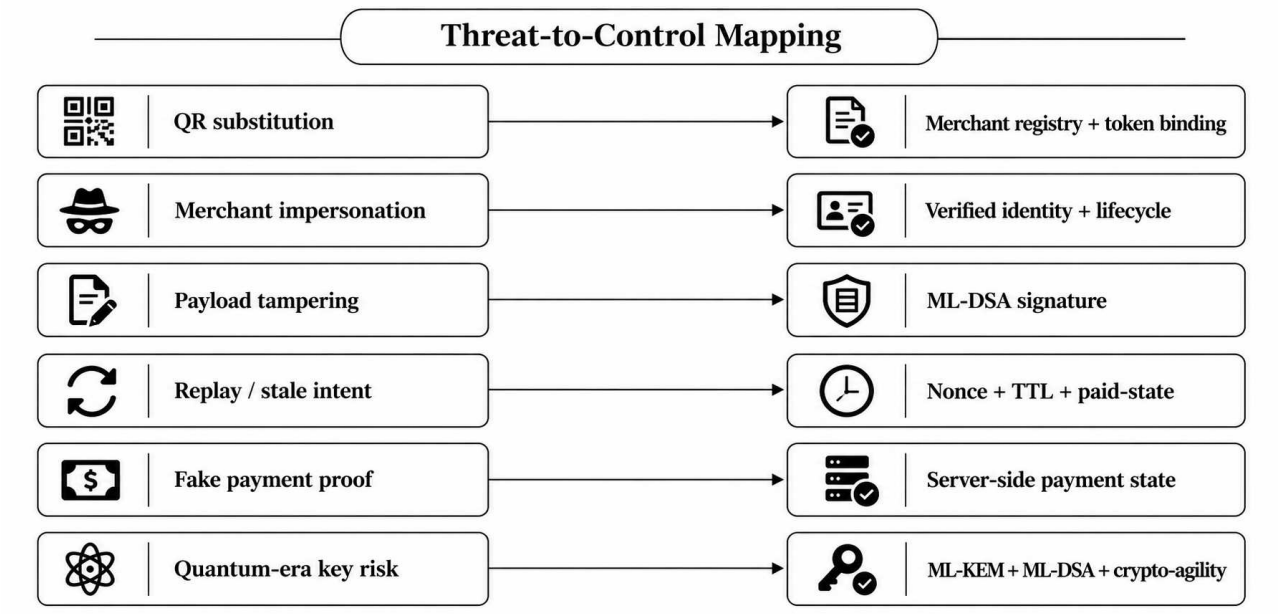

Fig. 2 Mapping of primary threats to Quantum-QRIS controls (AI-assisted explanatory schematic; content verified by the authors)

Table 2 Threat model and primary Quantum-QRIS security controls

Threat	Attack mechanism	Primary control
QR substitution	Legitimate physical QR is covered/replaced	Merchant registry, token/project binding, verified merchant display; future location binding
Merchant impersonation	Attacker-controlled account presented as intended merchant	Merchant lifecycle and trusted identity resolution
Payload tampering	Merchant, amount, intent, expiry, or nonce modified	Canonical message + ML-DSA verification
Replay	Previously valid EDC intent/payment reused	One-time token, TTL, transaction state, replay/paid check
Fake payment proof	Manipulated screenshot/visual receipt	Authoritative server-side transaction state; merchant-side confirmation
Key compromise	Active signing/KEM key becomes untrusted	ACTIVE/RETIRED/REVOKED lifecycle, rotation, audit
Quantum-era public-key risk	Future quantum attack on classical asymmetric primitives	ML-DSA-65, ML-KEM-768, crypto-agility
Weak freshness	Predictable nonce/token	Hybrid QRNG + OS CSPRNG conditioned entropy

this prototype. Communication between components is a separate exposure: possession of an authentic QR or of an encrypted channel does not by itself establish the customer's intended payee. The trusted-backend and uncompromised-active-private-key assumptions declared above are necessary limits on the claimed protection.

The intended security objectives are: *G1*, detect modification of signed dynamic intent fields; *G2*, reject expired and replayed dynamic intents; *G3*, reject requests whose merchant, project, token, or registry state does not satisfy the configured checks; *G4*, require authoritative transaction-state inspection before the merchant treats a payment as complete; and *G5*, enable traceable rejection, rotation, and revocation events. These objectives do not imply protection against a QR that is validly issued to another legitimate merchant and then placed at the victim's premises. Such a QR can satisfy digital authenticity checks while violating the customer's physical-context expectation, unless an independently trusted merchant-context binding is supplied.

The evaluation therefore distinguishes *digital validity* from *physical-context validity*. The latter would require a separately verified location, terminal, or merchant-confirmation signal; it is not established by a signature or by QR symbol conformance alone. The subsequent prototype results are interpreted within these boundaries, rather than as a formal security proof or evidence of production-system fraud prevention.

4 Quantum-QRIS Architecture

4.1 Deployment Modes

Mode A targets micro and small merchants using printed static QR. The printed artifact remains simple; trust is added through registered merchant identity, project/token binding, backend verification, and a fresh payment context. The design treats the QR symbol and merchant-presented payment artifact as compatible with the established QR/merchant-presented ecosystem rather than replacing the underlying QR representation [2], [3]. Mode B targets dynamic EDC/POS payments, where amount, merchant, transaction identifier, nonce, creation time, and expiry are bound into a short-lived intent. Mode C exposes equivalent controls through authenticated APIs for payment platforms, aggregators, banks, or enterprise integration.

4.2 QSSaaS Backend

QSSaaS runs as a centralized cryptographic service on Ubuntu and separates project identities through API-level controls. It provides ML-DSA signing/verification, ML-KEM key establishment, entropy services, lifecycle state, audit logging, backup/recovery, and usage management. Centralizing private-key operations simplifies merchant clients and allows rapid revocation, while introducing an availability dependency that must be addressed by production-grade redundancy.

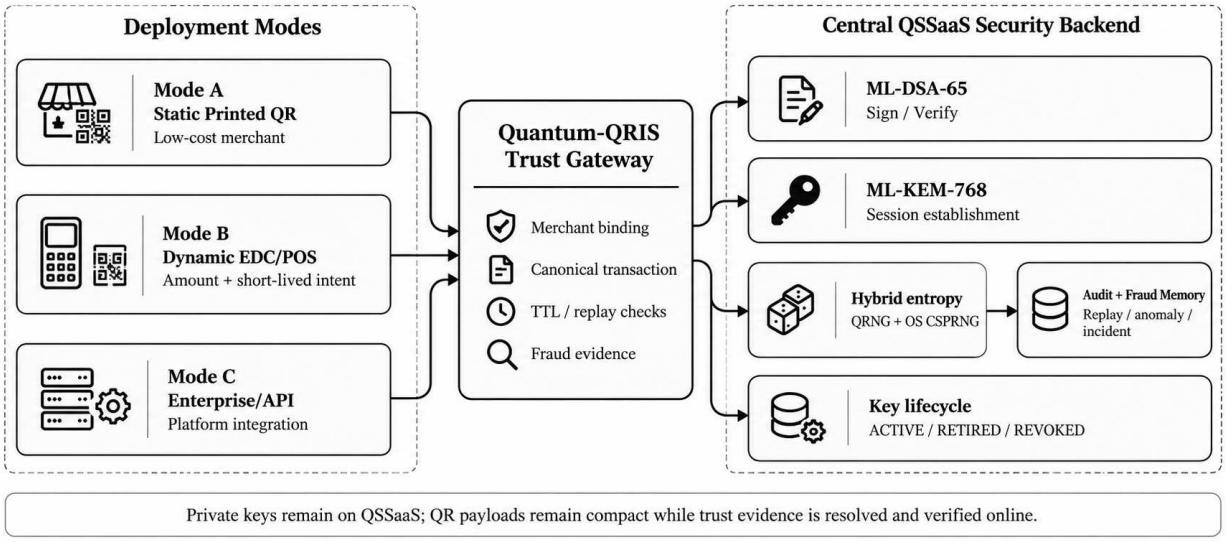

Fig. 3 Quantum-QRIS architecture and reusable QSSaaS backend (AI-assisted explanatory schematic; content verified by the authors)

4.3 Cryptographic and Freshness Design

ML-DSA-65, standardized in NIST FIPS 204 [22], is the baseline signature mechanism for merchant and transaction authenticity. ML-KEM-768, standardized in NIST FIPS 203 [23], is used for quantum-safe shared-secret establishment in the integrated payment/session path. Freshness material is generated through a hybrid entropy design that combines available quantum-derived bytes with the Linux CSPRNG and conditions the result before use. The entropy-source and random-bit-generation design follows the principles of NIST SP 800-90B and SP 800-90C [24], [25]. Where key derivation is required, the architecture uses HKDF consistent with RFC 5869 [26]. The prototype records entropy provenance and permits CSPRNG-only fallback so payment availability does not depend on the external QRNG service.

The implemented static payload is compact and references project, merchant, and QR token identifiers; the EDC intent binds transaction-specific fields including transaction ID, merchant identity, amount, currency, nonce, creation/expiry time, canonical message, and ML-DSA signature. This design deliberately avoids embedding large post-quantum public keys or signatures directly in the printed QR artifact.

4.4 Trust-Resolution Sequence and Binding Semantics

The architecture can be described in terms of the following *logical verification sequence*; the sequence specifies the intended trust decisions and does not claim a separately verified implementation of every step in all three deployment modes. For a static printed QR, the displayed payload supplies compact project, merchant, and token identifiers. The application resolves those identifiers against the backend registry, checks token and merchant status, and establishes a fresh transaction context. Confirmation of payment is distinct from verification of the QR: a QR token that resolves correctly is not evidence that a payment has been settled. The prototype's reported static-token and merchant-mismatch scenarios support the corresponding digital checks, whereas physical-location provenance remains outside their scope.

For a dynamic EDC/POS intent, the signed context includes the transaction and merchant identifiers, amount and currency, nonce, creation and expiry times, and canonical message. The verifier must associate the signing public key with a trusted merchant/project identity, verify the signature over the same canonical representation, enforce the validity interval and one-time/replay state, and inspect the transaction status before presenting a payment as completed. The order and atomicity of the state transitions matter: a verification decision and a state update that can race may permit duplicate acceptance even if signature validation is correct. The study's functional tests exercise selected lifecycle violations, but do not constitute a concurrency or formal state-machine proof.

This separation is essential for interoperability claims. ISO/IEC 18004 specifies the QR symbol, and

Table 3 Logical trust decisions and limits of the reported prototype evidence

Decision point	Required trust evidence	Evidence boundary in this study
Static QR resolution	Registered project, merchant, and token; active registry state	Positive and negative token/identity scenarios; not a field test of sticker attacks
Dynamic intent verification	Valid canonical signature and merchant association	ML-DSA service success and functional tamper/mismatch tests; no formal protocol proof
Freshness and replay	Nonce/expiry and transaction-state checks	Expiry/replay rejections observed; batch scheduling affected EDC verification outcomes
Payment completion	Backend transaction status independent of screenshots	Simulated end-to-end payment workflow; not proof of production acquirer/QRIS settlement
Physical merchant context	Independent confirmation of intended physical recipient	Not established for valid-but-misplaced QR substitution

the EMVCo merchant-presented specification addresses payment-QR data representation; neither fact establishes that the custom prototype payload is accepted unchanged by commercial QRIS readers or payment acquirers. Hence, compact-QR feasibility is an architectural and prototype finding, not a demonstrated certification or backward-compatibility result for the national payment rail.

4.5 Key Establishment, Trust Bootstrap, and Entropy Assurance

ML-KEM-768 provides a standardized key-encapsulation primitive, but a secure payment-session protocol additionally requires authenticated association of the recipient's encapsulation public key with the intended service identity, appropriate key derivation and context separation, ciphertext validation, and protected management of active and retired key material [27, 28]. A shared secret obtained from encapsulation should not be interpreted as merchant authentication in the absence of this binding. The current benchmark isolates ML-DSA service-path operations but does not report standalone ML-KEM encapsulation/decapsulation timings, authenticated key-establishment transcripts, or protocol-level downgrade tests. It therefore supports the stated integrated-system feasibility claim only within this explicitly limited evidence scope.

Similarly, the use of externally supplied QRNG bytes together with an operating-system CSPRNG must be distinguished from certification of an entropy source or random-bit generator. NIST SP 800-90B and SP 800-90C provide design and validation criteria; citing them does not establish compliance without the corresponding source characterization, conditioning analysis, and validation evidence [24, 25]. The reported CSPRNG-only fallback is an availability measure and should not be treated as a proven quantitative entropy improvement. These boundaries permit the prototype to be evaluated without overstating cryptographic or entropy-source assurances.

5 Methodology

5.1 Research Design

The study follows a design-science research approach. First, real QRIS incidents and recent literature were analyzed to identify concrete trust failures. Second, security requirements were translated into an architecture combining cryptographic authenticity, lifecycle state, freshness, and fraud evidence. Third, the architecture was instantiated as a proof of concept with separate merchant and QSSaaS systems. Fourth, functional/security scenarios were executed. Fifth, repeated performance benchmarks quantified service-path behavior and end-to-end payment latency.

5.2 Prototype Environment

The merchant application ran on Ubuntu with the active service bound to localhost and exposed to the user flow through the local application layer. The central QSSaaS ran on a separate physical Ubuntu host reachable through a secure network path. Merchant-side components used a SQLite transaction/merchant database and a shared Python environment. Across the benchmark runs, the recorded source hashes for the main merchant application, merchant registry, EDC service, QSS client, and nonce service remained identical, supporting version consistency.

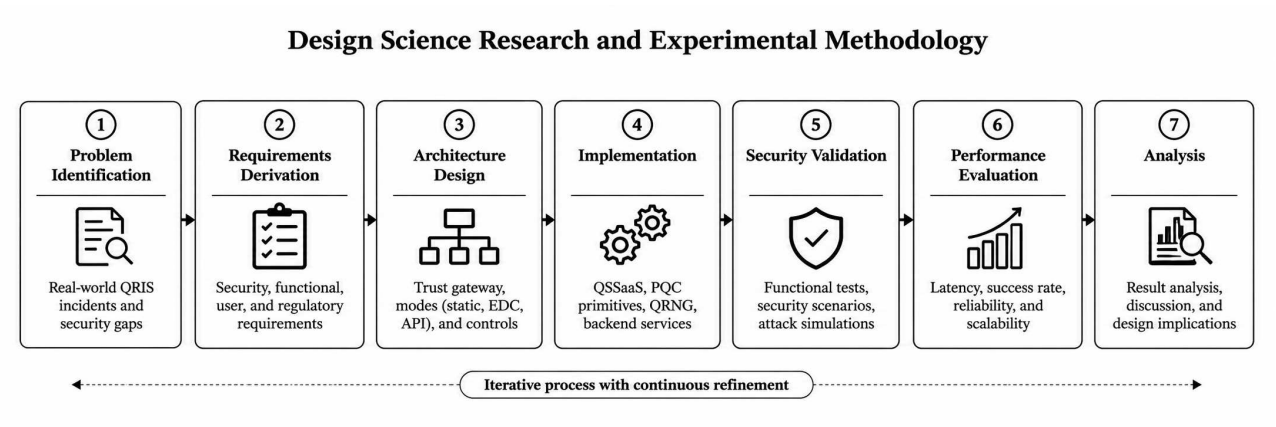

Fig. 4 Design-science and experimental workflow used in the Quantum-QRIS study (AI-assisted explanatory schematic; content verified by the authors)

5.3 Functional and Security Validation

The validation baseline separates positive-flow correctness from expected security rejections. A security control is considered successful when it rejects an invalid or stale transaction with the expected reason. This distinction is essential because HTTP 403/409 responses may represent correct enforcement rather than service failure.

5.4 Performance Benchmark Design

The Pilot is reported separately because its network path differed. It is not used to claim a controlled VPN effect. The Main and Full-Payment runs share the same network condition and therefore provide the stronger cross-run comparison. Timing used high-resolution monotonic measurements. Reported statistics include mean, sample standard deviation, minimum, median/P50, P95, P99, maximum, success/error counts, and serial completed operations per observed time. The latter is not a concurrent saturation-throughput measure.

5.5 Evaluation Metrics

Outcome definitions and uncertainty. For a positive transaction test, completion rate is defined as the number of attempts reaching the application-defined completed payment state divided by the total number of initiated attempts. For an invalid-intent attack test, an expected rejection is counted as correct enforcement only if the recorded reason corresponds to the intended control; such a rejection is not counted as positive payment completion. Service-path latency includes application/network dependencies and is not a cryptographic primitive timing. The end-to-end latency summaries are conditional on successful completion, and should not be interpreted as the response-time distribution of all initiated transactions, including failures and timeouts.

As an additional descriptive uncertainty measure, a 95% Wilson score interval can be computed for an observed completion fraction $\hat{p} = x/n$ with $z = 1.96$:

$$\mathrm{CI}_W = \frac{\hat{p} + z^2/(2n) \pm z\sqrt{V}}{1 + z^2/n}, \tag{1}$$

where $V = \hat{p}(1-\hat{p})/n + z^2/(4n^2)$. This interval is reported descriptively for the existing aggregate transaction counts. Its usual repeated-sampling interpretation assumes sufficiently independent, identically distributed Bernoulli attempts; persistent merchant-state, fixture, or shared-service faults may violate that assumption [30]. Confidence intervals for means or tail percentiles cannot be reconstructed reliably from the aggregate summary tables alone.

Reproducibility boundary. The reported code-hash consistency provides evidence that the measured source versions did not change across the stated runs; it does not by itself enable independent replication. A reproducible research package would also need non-secret configuration, software and algorithm-library versions, machine specifications, operating-system configuration, time-to-live settings, network topology and timing boundaries, scenario scripts, transaction-status definitions, and sanitized per-attempt records. These items are documented here as replication requirements, not as materials asserted to have been released. Claims tied to any unavailable item must remain limited to the summary evidence actually reported.

Table 4 Functional and security validation scenarios

Test	Scenario	Expected security outcome	Assessment
T01	Genuine static scan/payment	Accepted	PASS
T02	Genuine dynamic EDC creation/scan/verify/payment	Accepted	PASS
T03	Reuse of already-paid EDC	Rejected replay	PASS
T04	Fresh EDC after replay rejection	Accepted	PASS
T05	Static token tamper/unknown or rotated token	Rejected	PASS
T06	Static merchant mismatch	Rejected	PASS
T07	Expired/tampered EDC	Rejected	PASS
T08	Revoked merchant EDC access/use	Rejected	PASS
T09	Fake/forged QR; unregistered merchant/reused blocked destination	Payment blocked; incident evidence	PASS

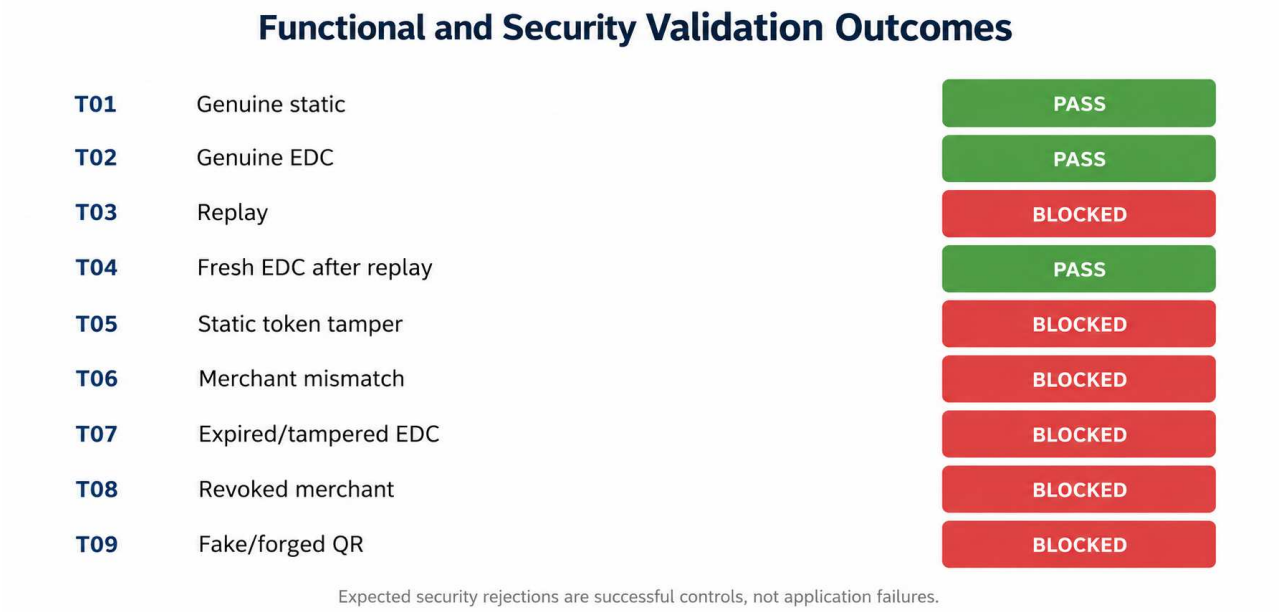

Fig. 5 Functional and security validation baseline; "Blocked" denotes successful security enforcement (AI-assisted explanatory schematic; content verified by the authors)

Table 5 Performance benchmark design

Run	Iterations	Warm-up	Include pay	Network condition	Role
Pilot	20	5	No	VPN / different network path	Initial sanity and sensitivity observation
Main	300	20	No	Same network	Primary component/verification dataset
Full payment	100	10	Yes	Same network	End-to-end payment dataset

Table 6 Evaluation metrics and interpretation

Metric	Interpretation
ML-DSA sign/verify service-path latency	Authenticity overhead including QSSaaS/network path
Nonce generation latency	Freshness/entropy operational cost
Merchant/static verification latency	Identity and QR-token validation cost
EDC create E2E	Dynamic intent creation overhead
HTTP static/EDC verification	Application-level verification
Static/EDC payment E2E	User-visible end-to-end completion latency
P50/P95/P99	Central and tail-latency behavior
Completion / rejection reason	Operational success and security-control behavior
Serial ops/s	Sequential processing rate; not load-test capacity

5.6 Use of generative AI and AI-assisted tools

OpenAI ChatGPT was used to support language refinement, manuscript organization, and preparation of selected explanatory schematics (Figs. 1–5). It was not used to generate, fabricate, alter, or statistically analyze experimental data, benchmark results, or primary research evidence. Data visualizations in Figs. 6–9 were derived from measured benchmark data using reproducible computational plotting. All AI-assisted text and visual materials were critically reviewed, corrected where necessary, and approved by the authors, who retain full responsibility for the scientific content and conclusions.

6 Results

6.1 Main Benchmark: Core Post-Quantum and Verification Stability

All seven core metrics completed successfully in the Main run. ML-DSA-65 signing achieved 300/300 success with mean 515.79 ms and P95 737.98 ms. Verification achieved 300/300 success with mean 497.85 ms and P95 653.08 ms. EDC creation, which composes merchant checks, freshness, signing, verification, and persistence, achieved 300/300 success with mean 2366.76 ms and P95 3259.09 ms.

6.2 Same-Network Cross-Run Consistency

The two same-network runs produced closely aligned means for most core components. EDC creation differed by only about 0.53%; nonce generation by about -3.0%; ML-DSA signing by about -8.0%; and ML-DSA verification by about -8.4%. HTTP static verification increased by approximately 14.2% in the full-payment run, illustrating that application/service state can shift even when the network condition and code base remain stable.

6.3 EDC Lifecycle Enforcement

EDC verification showed a markedly lower apparent success rate under batch sequencing: local verification completed 52/300 and HTTP verification 50/300 in the Main run. Error evidence shows that 250 HTTP verification attempts were rejected with EDC_QR_EXPIRED, while 248 local attempts returned predicate_failed. In the Full-Payment run, HTTP EDC verification produced 50 EDC_QR_EXPIRED rejections. These are expected lifecycle-security outcomes for short-lived intents, not failures of ML-DSA itself.

6.4 Full-Payment Performance

Among successful transactions, EDC payment exhibited lower mean and P95 latency than static payment. Static payment completed 76/100 attempts with mean 4426.77 ms and P95 6295.59 ms. EDC payment completed 69/100 attempts with mean 2967.14 ms and P95 3830.38 ms. These figures must be interpreted jointly with completion rate because latency statistics are computed over successful observations.

6.5 Error and Rejection Composition

The error distribution demonstrates why system-level results must be separated from cryptographic primitive reliability. Hundreds of ML-DSA operations completed successfully, whereas full-payment completion was affected by merchant identity, fixture/state, and service dependencies. This is a systems finding rather than evidence that PQC primitives were unreliable.

6.6 Pilot VPN Observation

Pilot values were generally higher, particularly for nonce generation, ML-DSA signing, and EDC creation. Because the pilot used a different network path and

Table 7 Main benchmark results for core post-quantum and verification services

Metric	Success	N	Success %	Mean ms	P50 ms	P95 ms	P99 ms
ML-DSA-65 sign via QSSaaS	300	300	100.00	515.79	479.07	737.98	1,841.70
ML-DSA-65 verify via QSSaaS	300	300	100.00	497.85	457.32	653.08	1,202.23
Nonce generation	300	300	100.00	669.03	632.87	689.52	1,665.29
Merchant verification	300	300	100.00	500.81	486.00	682.22	759.79
Static QR verification	300	300	100.00	495.47	474.80	666.13	753.31
HTTP static verification	300	300	100.00	478.33	441.07	686.61	931.94
EDC creation E2E	300	300	100.00	2,366.76	2,236.47	3,259.09	4,242.53

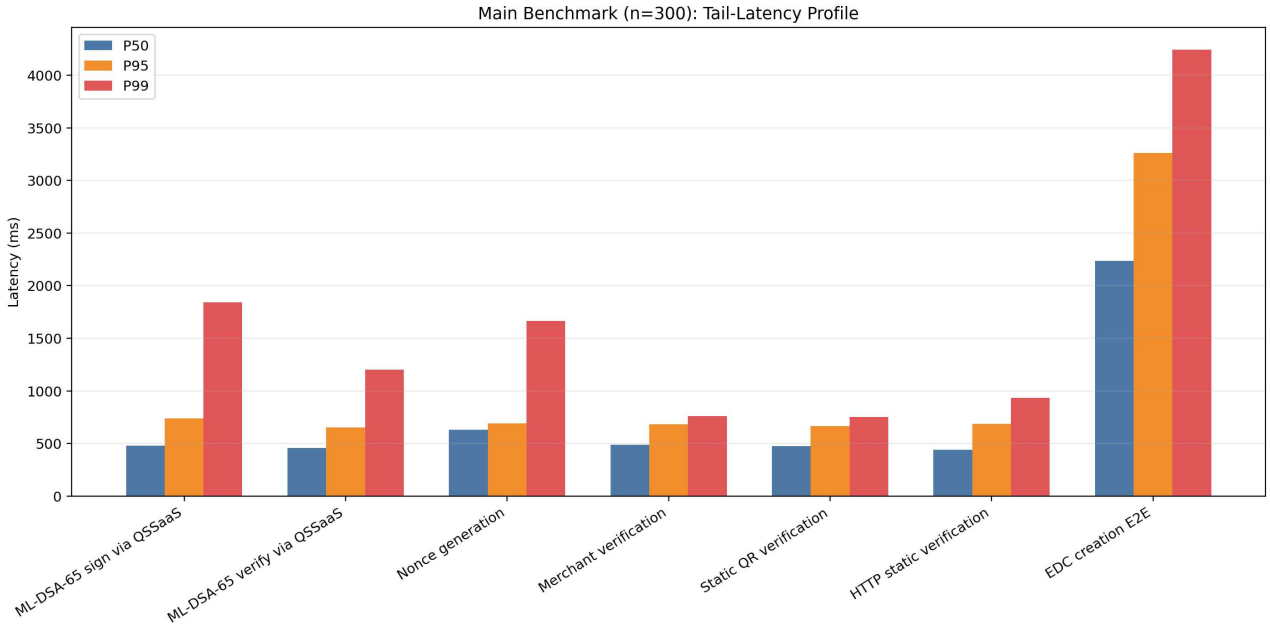

Fig. 6 P50, P95, and P99 service-path latency in the Main benchmark (n=300)

Table 8 Same-network cross-run consistency

Metric	Main mean ms	Full mean ms	Delta %	Main success %	Full success %
Nonce generation	669.03	648.94	-3.00	100.00	100.00
Merchant verification	500.81	465.03	-7.14	100.00	100.00
ML-DSA-65 sign via QSSaaS	515.79	474.44	-8.02	100.00	100.00
ML-DSA-65 verify via QSSaaS	497.85	455.96	-8.41	100.00	100.00
Static QR verification	495.47	459.06	-7.35	100.00	100.00
HTTP static verification	478.33	546.31	14.21	100.00	100.00
EDC creation E2E	2,366.76	2,379.28	0.53	100.00	100.00

Table 9 Full-payment performance

Flow	N	Success	Completion %	Mean ms	P50 ms	P95 ms	P99 ms
Static payment E2E	100	76	76.00	4,426.77	4,214.44	6,295.59	7,681.15
EDC payment E2E	100	69	69.00	2,967.14	2,818.19	3,830.38	6,079.43

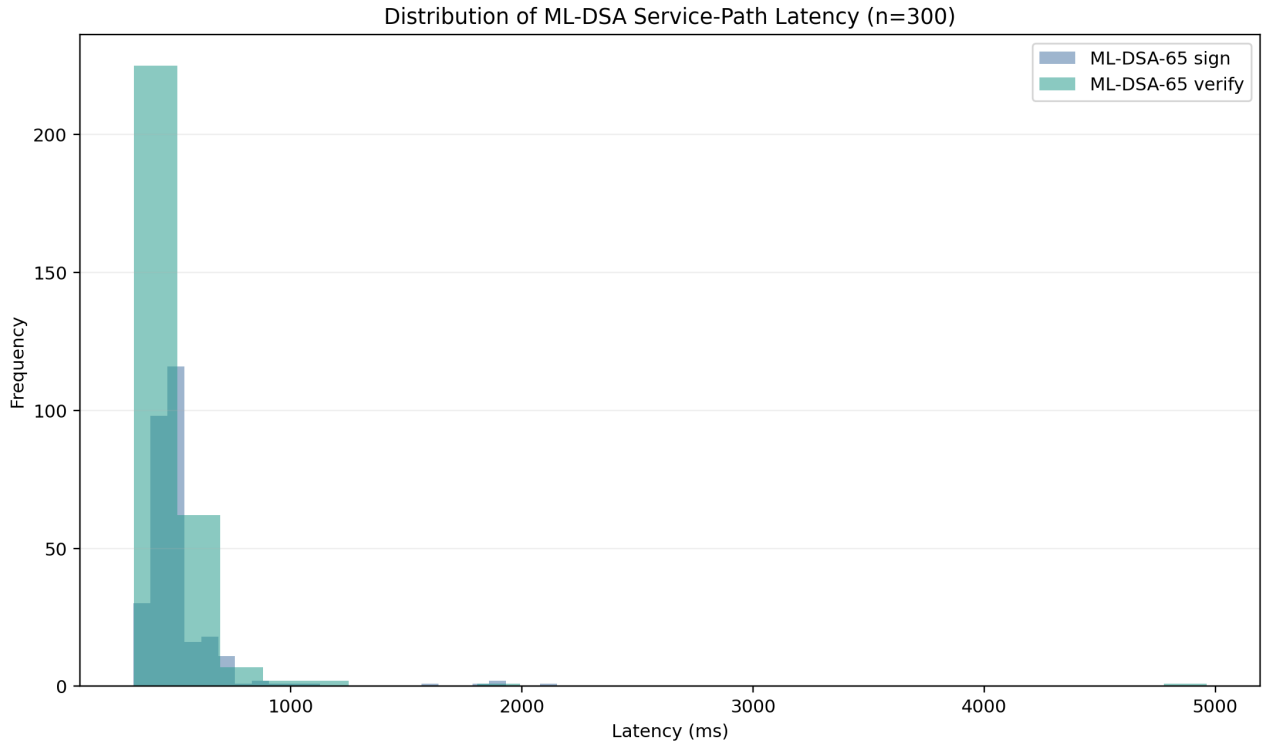

Fig. 7 ML-DSA-65 service-path latency distributions in the Main benchmark

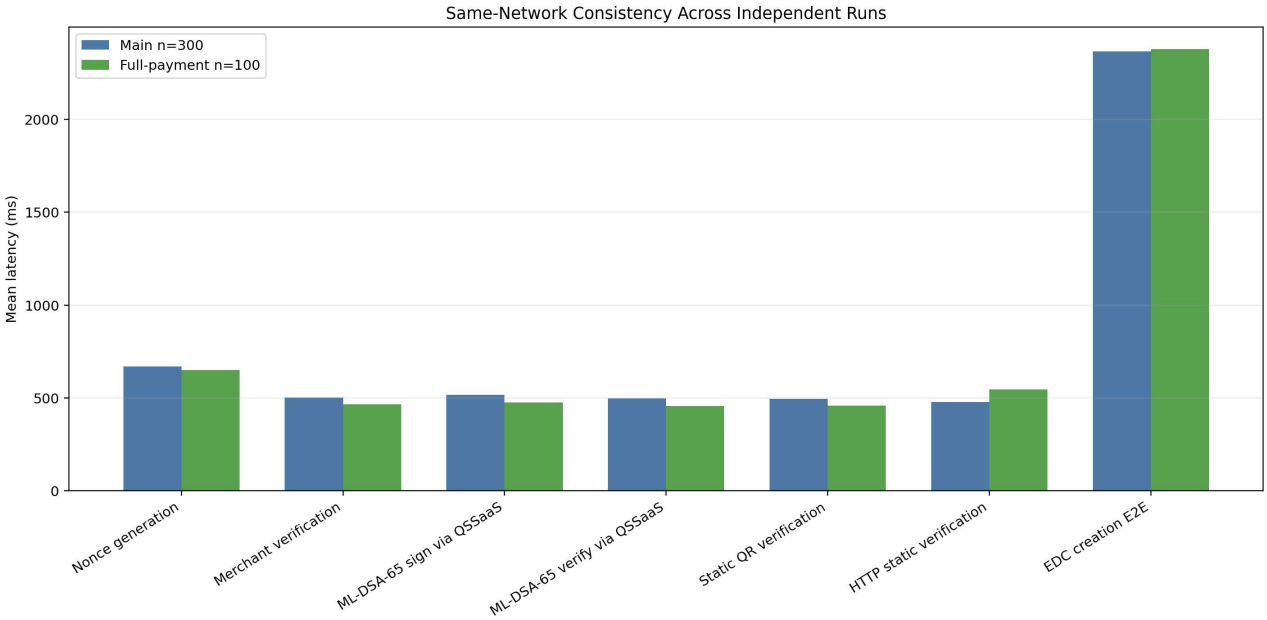

Fig. 8 Mean latency consistency between two independent same-network runs

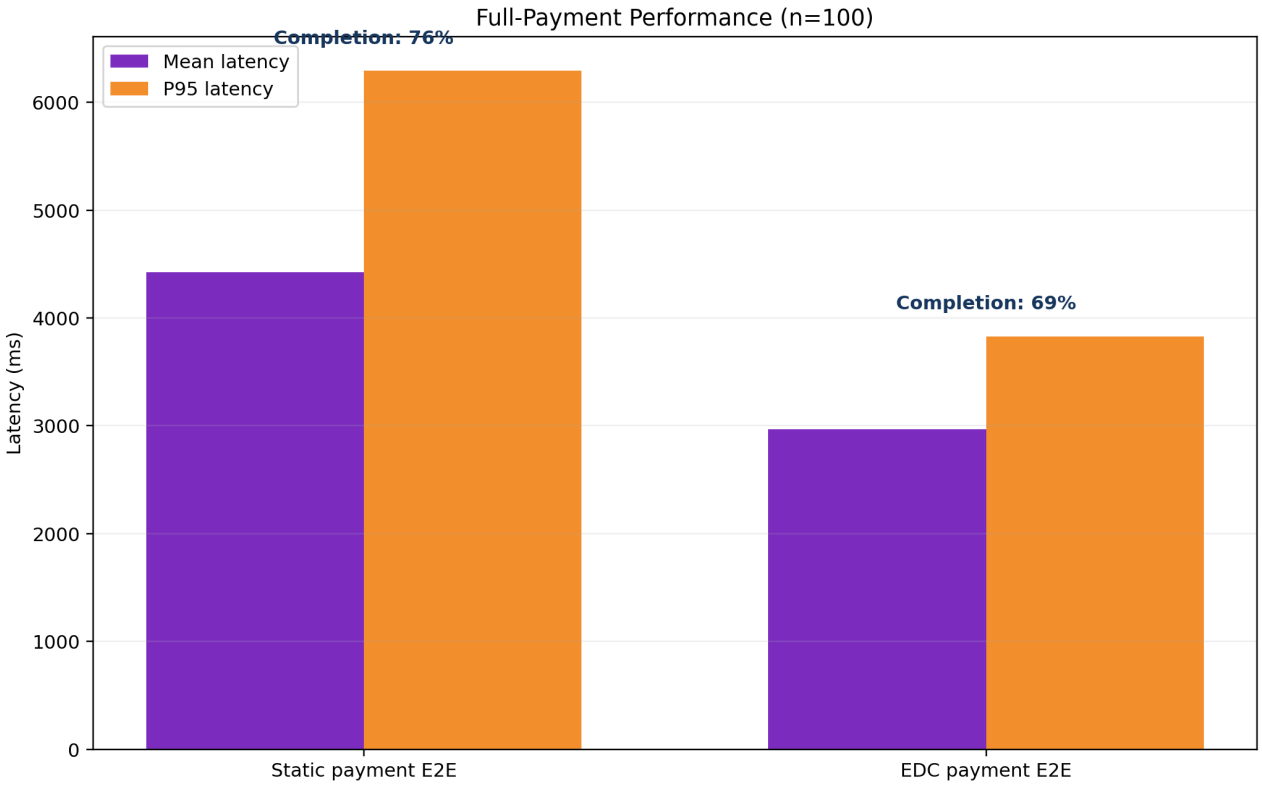

Fig. 9 Full-payment latency and observed completion rate (n=100)

Table 10 Error and security-rejection composition

Run	Metric	Count	Observed reason	Interpretation
Main	HTTP EDC verify	250	403 EDC_QR_EXPIRED	TTL/lifecycle security enforcement
Main	EDC verify local	248	predicate_failed	Lifecycle/state rejection
Full	HTTP EDC verify	50	403 EDC_QR_EXPIRED	TTL/lifecycle security enforcement
Full	EDC verify local	49	predicate_failed	Lifecycle/state rejection
Full	Static payment	21	403 merchant identity verification failed	Merchant identity control
Full	Static payment	3	QSS/key-service HTTP 500 paths	Operational dependency
Full	EDC payment	24	Fixture: merchant could not be verified	Fixture/setup dependency
Full	EDC payment	2	409 MERCHANT_NOT_VERIFIED	Merchant identity control
Full	EDC payment	5	QSS/key-service HTTP 500-related paths	Operational dependency

Table 11 Pilot VPN observation versus main same-network benchmark

Metric	Pilot VPN mean ms	Main same-network mean ms	Descriptive reduction %
Nonce generation	1,705.42	669.03	60.77
Merchant verification	544.34	500.81	8.00
ML-DSA-65 sign via QSSaaS	1,060.47	515.79	51.36
ML-DSA-65 verify via QSSaaS	715.16	497.85	30.39
Static QR verification	642.08	495.47	22.83
HTTP static verification	665.55	478.33	28.13
EDC creation E2E	3,929.68	2,366.76	39.77

smaller n, the difference is reported only as a topology-sensitive observation. It nevertheless supports the methodological point raised by Schöffel et al. [18]: service/network effects can dominate system-level PQC timing and must not be mislabeled as primitive cryptographic cost.

6.7 Uncertainty of Completion Rates and Diagnostic Separation

The 76 completed static-payment attempts and 69 completed EDC-payment attempts in Table 9 can be analyzed without introducing new experimental observations. Using the Wilson formula in Eq. 1, the descriptive 95% interval is approximately 66.77–83.31% for static payment and 59.37–77.22% for EDC payment. These intervals summarize sampling uncertainty under the simplifying independence assumption; they do not correct the systematic fixture, identity, or service faults observed in the experiment. In particular, the intervals must not be used to assert comparative superiority between the two flows, because the successful-attempt latencies and completion rates are conditional on different application states.

The failure composition in Table 10 also motivates distinguishing security-policy rejection from experimental setup and service availability. Rejections attributable to nonmatching or unverified merchant identities are correct *only for a deliberately invalid request*; for an intended legitimate payment they represent a blocking operational outcome requiring root-cause analysis. HTTP 500-related failures are service-path errors rather than intended security rejections. Likewise, expired EDC intents demonstrate that the expiry predicate was enforced, but expiry caused by sequential benchmark scheduling cannot be counted as successful completion of the intended positive payment flow. The existing observations are therefore useful diagnostic evidence, while they also prevent a claim of production-level end-to-end reliability.

7 Discussion

7.1 Answers to the Research Questions

RQ1 - Feasibility. The prototype demonstrates that standardized post-quantum trust services can be integrated into both static and dynamic QR-payment workflows while preserving a compact QR artifact. Core signing, verification, static verification, nonce generation, merchant verification, and EDC creation each achieved 100% success in the 300-iteration main run.

RQ2 - Security controls. The functional/security validation shows successful genuine payments and expected blocking of replay, expired/tampered intents, revoked merchants, merchant mismatch, token tampering, and fake/forged QR scenarios. The result supports a defense-in-depth model in which cryptography, identity state, transaction state, and fraud evidence complement one another.

RQ3 - Operational cost. Same-network ML-DSA sign/verify service paths are approximately half a second on average in this prototype, while EDC creation is approximately 2.37 s and successful end-to-end EDC payment approximately 2.97 s. Tail events are materially higher, reinforcing the need to report P95/P99 rather than only means.

RQ4 - Real-world coverage. Quantum-QRIS directly addresses tampering, stale/replayed dynamic intents, unregistered/revoked merchant states, fake QR patterns, and server-side transaction confirmation. It only partially addresses a harder physical-context attack: an attacker may place a cryptographically valid QR belonging to another valid merchant at the victim location. Preventing that case requires merchant-terminal-location binding, trusted context attestation, or independent physical verification in addition to cryptographic validity.

7.2 Mapping Results to the Indonesian Incidents

This mapping clarifies the central contribution: Quantum-QRIS does not claim that PQC prevents sticker replacement or social engineering by itself. Instead, it moves payment trust away from visual appearance toward backend-verifiable merchant identity, signed transaction state, freshness, and payment completion. The architecture therefore addresses the digital trust chain that real incidents expose, while acknowledging the remaining physical-context problem.

7.3 Novelty Relative to Prior Work

The novelty of Quantum-QRIS lies in a specific systems-security composition and its experimental validation, rather than in proposing a new cryptographic primitive. First, the architecture decouples compact static/dynamic QR artifacts from online trust resolution, allowing printed QR usability to coexist with post-quantum merchant and transaction verification. Second, it binds cryptographic authenticity to merchant lifecycle, transaction freshness, replay/expiry

Table 12 Descriptive uncertainty derived solely from the reported full-payment counts

Flow	Completed/attempted	Rate	Wilson 95% interval
Static	76/100	76%	66.77–83.31%
EDC	69/100	69%	59.37–77.22%

Table 13 Mapping real-world QRIS incidents to Quantum-QRIS coverage

Case	Observed trust failure	Required assurance	Quantum-QRIS coverage	Assessment
Jakarta 2023	Attacker QR placed in trusted donation context	Merchant/context binding	Registry/token validation can reject unknown QR; valid-attacker QR still needs location/organization binding	Partial
Bandung 2025	QR overlay redirects payment to another recipient	Merchant identity binding	Mismatch/unknown token and unregistered merchant can be blocked; valid merchant substitution remains a physical-context gap	Partial-to-strong
Surabaya 2026	Manipulated visual payment proof	Transaction confirmation trust	Authoritative backend transaction state prevents screenshot-only confirmation if merchant workflow is followed	Strong at system layer

state, authoritative payment completion, and fraud evidence within one trust path. Third, it exposes valid-but-misplaced QR as a distinct residual attack: a QR can be cryptographically genuine yet contextually fraudulent when presented at the wrong physical merchant location. Prior QR-security studies focus primarily on anti-counterfeiting, phishing detection, or integrity services; PQC financial studies focus on algorithm selection, migration, wallets, or primitive/system performance; and mobile-fraud studies focus on behavioral classification. Quantum-QRIS connects these previously separate concerns at payment-workflow level and evaluates both expected security rejection and operational service-path behavior. The contribution is therefore an experimentally grounded trust architecture and threat decomposition, not a claim that ML-DSA, ML-KEM, or QRNG individually constitute novel algorithms.

7.4 Security Claims and Boundaries

The evidence supports claims of prototype-level feasibility, enforced lifecycle controls, stable ML-DSA service operations, and end-to-end transaction execution. It does not support claims that the current national QRIS standard already uses PQC, that all fraud classes are solved, or that measured service-path latency equals algorithm-only latency. The system should therefore be described as a security enhancement architecture and experimental platform rather than a production modification of the national payment rail.

7.5 Evidence-to-Claim Traceability

To prevent an architectural proposal from being mistaken for a completed formal or field-security evaluation, Table 14 separates measured prototype observations from analytical conclusions and unresolved assurance obligations. In particular, a successful ML-DSA operation validates a cryptographic service call; it does not by itself validate merchant-onboarding provenance, physical QR placement, or payment-network settlement. This separation strengthens the interpretation of RQ1–RQ4 without adding unmeasured results.

7.6 Operational and Regulatory Implications

A production deployment would require payment-operator and regulatory alignment, high availability, formal merchant onboarding, secure device binding, scalable key lifecycle, privacy controls for audit/fraud evidence, and explicit user-interface semantics that distinguish live verification from cached or unavailable verification. The BIS roadmap similarly emphasizes coordinated financial-sector migration, cryptographic inventory, agility, and staged implementation rather than isolated algorithm replacement [10].

Table 14 Claim–evidence mapping and the remaining security assurance boundary

Claim	Existing supporting evidence	Not established by existing data
PQC signing/verification can operate within the prototype	300/300 ML-DSA service operations in Table 7	Primitive-only cost, standalone ML-KEM assurance, and security under key compromise
Selected identity and lifecycle rules reject invalid requests	T03–T09 in Table 4; expiry/rejection counts in Table 10	Exhaustive attack resistance, concurrent race freedom, and formal authentication proof
A compact QR can reference online security state	Architecture in Fig. 3 and static verification measurements	Interoperability with production QRIS acquirers/readers
End-to-end simulated payment execution is feasible	Completion and conditional latency in Table 9	Production payment settlement, high availability, or reliable completion of all valid attempts
Merchant identity checks support digital trust resolution	Merchant/token mismatch scenarios; Table 13	Physical validity of a QR issued to another legitimate merchant

8 Limitations

1. The prototype is not integrated into the production national QRIS/payment network; results describe the research architecture only.
2. The empirical evaluation does not include a controlled classical RSA/ECC or non-PQC baseline with identical workflow and workload; therefore, the study demonstrates feasibility and system behavior but does not claim comparative performance superiority over classical payment security.
3. The evaluation is a component and end-to-end prototype benchmark rather than an ablation study; the independent performance contribution of each control layer (PQC, merchant verification, entropy service, lifecycle checks, and fraud-memory) is not isolated.
4. ML-DSA and other QSSaaS measurements are service-path timings that include network/application overhead, not pure primitive microbenchmarks.
5. Standalone ML-KEM encapsulation/decapsulation microbenchmark values were not available in the benchmark environment; ML-KEM is evidenced in the integrated transaction path rather than as a separate timing result.
6. The reported throughput is serial completed operations per summed observed time, not concurrent saturation throughput.
7. Physical substitution using a QR that is cryptographically valid for another legitimate merchant is not fully solved without location, terminal, or trusted-context binding.
8. Fake payment-proof resistance depends on merchant-side authoritative confirmation being operationally enforced; human bypass of the merchant system remains a process risk.
9. Formal protocol proof, large-scale key-management evaluation, high-availability testing, and millions-of-merchants scalability are outside the present study.

9 Future Work

The results suggest a concrete research roadmap rather than a generic call for "more testing." The next phase should explicitly separate cryptographic microbenchmarks, security-control experiments, system performance, and deployment-scale resilience.

Priority comparative experiment. To address the remaining comparative-evidence gap, the next benchmark should execute an identical payment workflow under three security profiles—classical baseline, PQC-only substitution, and full Quantum-QRIS trust controls—so that cryptographic migration overhead and the incremental cost of identity/lifecycle controls can be measured separately.

10 Conclusion

Quantum-QRIS was developed from a practical security problem rather than from a cryptographic primitive in isolation. Real Indonesian QRIS incidents show that payment security can fail even when the underlying payment system processes instructions correctly: the wrong merchant can receive a legitimate payment, or a human operator can accept a fabricated payment proof. At the same time, financial systems require a path away from long-lived classical public-key assumptions. The proposed architecture responds by binding merchant identity, transaction context, freshness, lifecycle state, cryptographic evidence, and authoritative completion status through a reusable QSSaaS backend.

The experimental evidence is encouraging but appropriately bounded. In the main 300-iteration same-

Table 15 Future-work roadmap

Future work	Proposed experiment/design	Scientific value
Merchant-terminal-location binding	Bind merchant_id, terminal_id, location/context policy, acquirer, and signing identity	Addresses valid-but-misplaced QR substitution observed in physical sticker attacks
Just-in-time EDC benchmark	Create each intent immediately before verify/pay	Separates performance success from intentional 120-s TTL expiry
Standalone ML-KEM benchmark	Measure keygen/encaps/decaps P50/P95/P99 and bytes	Isolates quantum-safe key-establishment cost
Concurrent load evaluation	1/5/10/25/50 workers; queueing, throughput, CPU/RAM	Determines saturation capacity and QSSaaS scaling behavior
Controlled network A/B	Identical workload over LAN, VPN, WAN, mobile network	Quantifies network contribution to service-path PQC latency
Formal protocol verification	Tamarin/ProVerif plus ROM/UC security argument	Strengthens authentication, replay, and session-security claims
Fraud-memory analytics	Risk scoring over merchant/device/account reuse and anomaly sequences	Moves from rule/blocklist evidence toward measurable fraud detection
Trusted merchant confirmation UI	Live-verification indicator, signed receipt reference, staff SOP	Directly mitigates fake-screenshot operational trust failure
Crypto-agility and hybrid migration	Versioned profiles and classical+PQC hybrid transition	Supports phased financial-sector migration
High-availability QSSaaS	Redundancy, failover, replicated key metadata, recovery drills	Removes single-service dependency revealed by full-payment operational errors
Privacy-preserving audit	Data minimization, retention policy, access separation	Reduces centralized fraud-memory privacy risk
Field trial with MSMEs	Controlled real merchant/customer usability and attack simulations	Measures adoption, user comprehension, and socio-technical effectiveness

network benchmark, ML-DSA signing and verification, nonce generation, merchant/static verification, HTTP static verification, and EDC creation all achieved 100% success. Full-payment experiments demonstrated real end-to-end execution with 76% static-payment and 69% EDC-payment completion; their non-completions exposed application/state/service dependencies rather than instability of the signature primitive. EDC expiry rejections additionally demonstrated that transaction-lifecycle security was enforced, although batch sequencing complicated naive performance interpretation.

The core scientific conclusion is therefore that quantum-safe QR payment security is not achieved by merely replacing RSA/ECC with post-quantum algorithms. It requires a system-level trust architecture in which post-quantum authenticity, quantum-safe key establishment, entropy/freshness, merchant identity, payment state, and fraud evidence reinforce one another. Quantum-QRIS provides a deployable research baseline for that architecture and identifies merchant-location binding, concurrent scalability, formal verification, and production-grade resilience as the next critical steps.

Statements and Declarations

Acknowledgements The authors express their sincere appreciation to the Coordinating Ministry for Economic Affairs of the Republic of Indonesia. This research was conducted within the framework of a Cooperation Agreement between Republic of Indonesia Defense University and the National Research and Innovation Agency of Indonesia.

Author contributions H. A. Danang Rimbawa contributed to the conceptualization and development of the original research idea. Danny Setyowati was responsible for writing the original manuscript and presenting the research findings. Achmad Farid Wadjdi was responsible for data curation, including data organization and preparation.

Funding The authors declare that no financial support, research grants, or external funding were received from any public, commercial, or non-profit organization for conducting this research or preparing the manuscript.

Data availability The datasets generated and analyzed during this study were obtained through research simulations and are available from the corresponding author upon reasonable request.

Code availability The research code used for simulations, experimental evaluations, and data analysis is available from the corresponding author upon reasonable request. The code will be appropriately sanitized to exclude API keys, authentication credentials, and infrastructure-sensitive configuration details before disclosure.

Ethics approval Not applicable. This simulation-based study did not involve human participants, animal subjects, or identifiable personal data.

Consent to participate Not applicable. This study did not involve human participants requiring informed consent.

Consent to publish Not applicable. This manuscript does not contain identifiable personal information or materials requiring publication consent.

Competing interests The authors declare that they have no financial or non-financial competing interests that could have influenced the conduct, analysis, interpretation, or reporting of this research.

References

1. Bank Indonesia. Quick Response Code Indonesian Standard (QRIS) (2026). Available at: https://www.bi.go.id/id/fungsi-utama/sistem-pembayaran/ritel/kanal-layanan/qris/default.aspx. Accessed 30 Oct 2026.
2. EMVCo. EMV QR Code Specification for Payment Systems (EMV QRCPS): Merchant-Presented Mode, Version 1.1 (2020). Available at: https://www.emvco.com/emv-technologies/qr-codes/. Accessed 30 Oct 2026.
3. International Organization for Standardization. ISO/IEC 18004:2024, Information technology—Automatic identification and data capture techniques—QR code bar code symbology specification. ISO (2024). Available at: https://www.iso.org/standard/83389.html.
4. Njuguna, D., Ndia, J.: Quick Response Code Security Attacks and Countermeasures: A Systematic Literature Review. Journal of Cyber Security 7(1), 1–20 (2025). https://doi.org/10.32604/jcs.2025.059398
5. Bank Indonesia. BI Terus Edukasi Masyarakat dan Pedagang/Merchant agar Terhindar dari Upaya Penyalahgunaan QRIS (2023). Available at: https://www.bi.go.id/id/publikasi/ruang-media/news-release/Pages/sp_259323.aspx. Accessed 30 Oct 2026.
6. Tribrata News Polri. Pelaku Penempel QRIS Palsu Kumpulkan Uang Rp13 Juta (2023). Available at: https://tribratanews.polri.go.id/blog/hukum-4/pelaku-penempel-qris-palsu-kumpulkan-uang-rp13-juta-56647. Accessed 30 Oct 2026.
7. Hassani, Y.: Pedagang di Sekitar Kampus Tel-U Jadi Korban Penipuan QRIS Palsu. detikJabar (2025). Available at: https://www.detik.com/jabar/hukum-dan-kriminal/d-8158252/pedagang-di-sekitar-kampus-tel-u-jadi-korban-penipuan-qris-palsu. Accessed 30 Oct 2026.
8. kumparanNEWS. Pedagang di Pujasera Kampus di Bandung Jadi Korban Pemalsuan QRIS (2025). Available at: https://kumparan.com/kumparannews/pedagang-di-pujasera-kampus-di-bandung-jadi-korban-pemalsuan-qris-263ydbxsvbT. Accessed 30 Oct 2026.
9. Devi, A.: Pria Kaltim Tipu Toko di Surabaya Pakai Bukti Transaksi Palsu Hasil AI. detikJatim (2026). Available at: https://www.detik.com/jatim/hukum-dan-kriminal/d-8510532/pria-kaltim-tipu-toko-di-surabaya-pakai-bukti-transaksi-palsu-hasil-ai. Accessed 30 Oct 2026.
10. Auer, R., Dodson, D., Dupont, A., Haghighi, M., Margaine, N., Marsden, D., McCarthy, S., Valko, A.: Quantum-readiness for the financial system: a roadmap. BIS Papers No. 158, Bank for International Settlements (2025). Available at: https://www.bis.org/publications/paper-158-quantum-readiness-financial-system-roadmap.
11. Sadeghi, S., Chouhan, V., Aldarwbi, M., Ghorbani, A., Chow, A., Burko, R.: Securing financial sector applications in the quantum era: a comprehensive evaluation of NIST's recommended algorithms through use-case analysis. Expert Systems with Applications 288, 128243 (2025). https://doi.org/10.1016/j.eswa.2025.128243
12. Bush, C.: Quantum computing and digital payments: opportunities, threats, and the path toward regulation. International Review of Law, Computers & Technology (2026). https://doi.org/10.1080/13600869.2026.2623674
13. Alsuhibany, S.A.: Innovative QR Code System for Tamper-Proof Generation and Fraud-Resistant Verification. Sensors 25(13), 3855 (2025). https://doi.org/10.3390/s25133855
14. Yalda, R., Khan, A., Nepal, N.: B-PhishQR—A blockchain-based framework for secure QR code verification against phishing attacks. Telematics and Informatics Reports 21, 100289 (2026). https://doi.org/10.1016/j.teler.2025.100289
15. Woo, J., Lee, S., Park, I., Lee, S.: WebView-Based Hybrid Analysis of Link and Event for On-Device QR Phishing Detection Framework. Sensors 26(14), 4412 (2026). https://doi.org/10.3390/s26144412
16. Akram, M.W., Sood, K., Hassan, M.U., Subba, B.: Exemplifying Emerging Phishing: QR-Based Browser-in-the-Browser (BiTB) Attack. IEEE Networking Letters 7(4), 274–278 (2025). https://doi.org/10.1109/LNET.2025.3605640
17. Ranganatha, H.R., Syed Mustafa, A.: Enhancing fraud detection efficiency in mobile transactions through the integration of bidirectional 3D Quasi-Recurrent Neural Network and blockchain technologies. Expert Systems with Applications 260, 125179 (2025). https://doi.org/10.1016/j.eswa.2024.125179
18. Schöffel, M., Lauer, F., Rheinländer, C.C., Wehn, N.: Secure IoT in the Era of Quantum Computers—Where Are the Bottlenecks? Sensors 22(7), 2484 (2022). https://doi.org/10.3390/s22072484
19. Hu, M., Liu, Z., Zhou, Y.: Efficient post-quantum secure deterministic wallet scheme. Cybersecurity 7, 24 (2024). https://doi.org/10.1186/s42400-024-00216-w
20. Wang, Q., Liu, J., Li, G., Han, Y., Zhou, Y., Cheng, L.: A measurement-device-independent quantum secure digital payment. Physica A: Statistical Mechanics and its Applications 655, 130178 (2024). https://doi.org/10.1016/j.physa.2024.130178
21. Shaw, S., Dutta, R.: Post-quantum secure identity-based signature achieving forward secrecy. Journal of Information Security and Applications 69, 103275 (2022). https://doi.org/10.1016/j.jisa.2022.103275

22. National Institute of Standards and Technology: Module-Lattice-Based Digital Signature Standard. FIPS 204 (2024). https://doi.org/10.6028/NIST.FIPS.204
23. National Institute of Standards and Technology: Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS 203 (2024). https://doi.org/10.6028/NIST.FIPS.203
24. National Institute of Standards and Technology: Recommendation for the Entropy Sources Used for Random Bit Generation. NIST Special Publication 800-90B (2018). https://doi.org/10.6028/NIST.SP.800-90B
25. Barker, E., Kelsey, J., McKay, K., Roginsky, A., Sönmez Turan, M.: Recommendation for Random Bit Generator (RBG) Constructions. NIST SP 800-90C, National Institute of Standards and Technology (2025). https://doi.org/10.6028/NIST.SP.800-90C
26. Krawczyk, H., Eronen, P.: HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869, Internet Engineering Task Force (2010). https://doi.org/10.17487/RFC5869
27. Alagic, G., Barker, E., Chen, L., Moody, D., Robinson, A., Silberg, H., Waller, N.: Recommendations for Key-Encapsulation Mechanisms. NIST SP 800-227, National Institute of Standards and Technology (2025). https://doi.org/10.6028/NIST.SP.800-227
28. Barker, E.: Recommendation for Key Management: Part 1—General. NIST SP 800-57 Part 1, Revision 5, National Institute of Standards and Technology (2020). https://doi.org/10.6028/NIST.SP.800-57pt1r5
29. Rose, S., Borchert, O., Mitchell, S., Connelly, S.: Zero Trust Architecture. NIST SP 800-207, National Institute of Standards and Technology (2020). https://doi.org/10.6028/NIST.SP.800-207
30. Wilson, E. B.: Probable inference, the law of succession, and statistical inference. J. Am. Stat. Assoc. 22(158), 209–212 (1927). https://doi.org/10.1080/01621459.1927.10502953