Enrichment
Journal of Multidisciplinary Research and Development

Available at https://journalenrichment.com/index.php/jr/

Enrichment: Journal of Multidisciplinary Research and Development

Enhancing Transport Layer Security Using a Quantum Random Number Generator in Elliptic Curve Cryptography

Agil Almunawar*, Aulia Khamas Heikhmakhtiar, Akil Suwandi
Universitas Pertahanan Republik Indonesia
Email: agil.almunawar@tp.idu.ac.id*, aulia.heikhmakhtiar@idu.ac.id, akilnangsuwandi1799@gmail.com

ABSTRACT

Transport Layer Security (TLS) protects most internet traffic, yet its Elliptic Curve Cryptography (ECC) operations can fail catastrophically when randomness is biased or predictable. This study presents a standards-compliant way to harden ECC in TLS by injecting Quantum Random Number Generator (QRNG) entropy into three critical points: (i) private-key generation, (ii) ECDHE ephemeral-scalar selection, and (iii) (optionally) ECDSA per-message nonces while preserving RFC 8446 interoperability. This study designs a lightweight entropy pipeline that mixes QRNG output with the operating system's CSPRNG using a keyed extractor, and implements a reproducible Python toolchain (CLI/GUI) for signing/verification with ECDSA (P-256). Randomness is assessed with Shannon entropy and a subset of NIST SP 800-22 tests (e.g., frequency/monobit, runs, serial, approximate entropy) on multi-megabyte streams; functional tests validate signature correctness and TLS-like key-agreement flows. Results show that QRNG-mixed streams provide near-maximal entropy and stable bit balance, and that keys and ephemeral scalars derived from the mixed source remain unpredictable even under PRNG state-compromise assumptions. The approach integrates transparently with existing ECC stacks, and practical overhead is minimal because entropy is buffered. We discuss residual risks (QRNG availability, health testing, secure transport from device to host) and outline extensions for explicit nonce control and broader TLS handshake coverage. This work contributes a pragmatic, deployable artifact complete with code and evaluation scripts that institutes stronger entropy hygiene for ECC within TLS as organizations plan their transition to post-quantum cryptography.

Keywords: Transport Layer Security; Elliptic Curve Cryptography; Quantum Random Number Generator; ECDSA; ECDHE; NIST SP 800-22.

INTRODUCTION

Transport Layer Security (TLS) is the de-facto standard for end-to-end confidentiality, integrity, and entity authentication across web, e-commerce, and cloud services, rooted in the SSL/TLS handshake and X.509-based Public Key Infrastructure (PKI) (Chou, 2002; Khan et al., 2022). Yet the practical security of a TLS channel ultimately rests on two pillars: (i) the hardness assumptions of its public-key primitives (RSA/ECC) and (ii) the unpredictability of randomness used to generate long-term keys, per-session ECDHE secrets, and per-message nonces. Real deployments show that transport-layer choices and timing constraints interact with security outcomes. For example, in power-grid synchrophasor telemetry, protocol and resource decisions at the transport layer have measurable security implications, so cryptographic hygiene and robust handshake entropy are operational necessities, not just theoretical niceties (Wang, Gamage, & Hauser, 2016; Koschuch, Hudler, & Krüger, 2022). On the efficiency side, Elliptic Curve Cryptography (ECC) remains compelling because it provides RSA-comparable security with much shorter keys and compact hardware realizations, lowering both computation and bandwidth overhead an advantage for high-throughput servers and constrained IoT/embedded platforms (Janagan & Devanathan, 2012; Shah et al., 2024).

The strategic horizon is shaped by quantum computing: algorithms such as Shor's threaten integer factorization and elliptic-curve discrete logarithms, motivating a transition to post-quantum cryptography (PQC) while maintaining service performance (Fernández-Caramés, 2020; Cherkaoui Dekkaki et al., 2024; Sajimon, Jain & Krishnan, 2022). In this transitional window before standardized PQC is ubiquitously deployed in every stack component, entropy quality is a high-leverage control: many real-world incidents trace not to the algorithmic core but to predictable nonces, biased keys, or depleted entropy pools during boot or reseed (Corrigan-Gibbs et al., 2013; Švarcmajer et al., 2025). Here, Quantum Random Number Generators (QRNGs) are attractive because they harvest irreducible quantum noise (e.g., coherent-optical receivers) to produce true randomness with high entropy, low correlation, and strong statistical profiles (typically passing NIST SP 800-22 batteries at scale) (Samsonov et al., 2020; Van der Heide et al., 2022). Rather than replace OS CSPRNGs, a QRNG→CSPRNG mixing strategy strengthens resistance to state compromise and bias while preserving compatibility with today's TLS/ECC stacks (Mamun et al., 2024).

At the same time, the engineering literature highlights two complementary hardening tracks for transport-layer security. The first focuses on handshake/PKI engineering to reduce downgrade and man-in-the-middle surfaces (Khan et al., 2022; Chou, 2002). The second explores algorithmic or architectural variants that improve performance or add functionality for digital-signature and confidentiality workflows (Azeez et al., 2024; Sudharson & Arun, 2022). Examples include RSA with bit-insertion for large-file/cloud settings; web-app signature architectures; signature schemes avoiding modulo-inverse operations; arithmetic-combined signature with watermarking for multimedia; ISRSAC-based signatures; organization-scale RSA signature deployment; "quantum-RSA" watermarking for images; and signal-processing pipelines that combine compression/optimization with RSA for low-resource devices (Li, Zhu, Wang, Meng, & Qin, 2024; A Digital Signature Architecture for Web Apps, n.d.; A Highly Secure Digital Signature Algorithm Without Modulo Inverse Operations, n.d.; An Effective Arithmetic Combined Digital Signature with Digital Watermarking, n.d.; Digital signature based on ISRSAC, n.d.; Implementation Security Digital Signature Using RSA Algorithm As A Letter Validation And Distribution Validation System, n.d.; Image security using quantum Rivest-Shamir-Adleman cryptosystem algorithm and digital watermarking, n.d.; A Novel Technique to Compress Photoplethysmogram Signal Improvised with Particle Swarm Optimization and RSA, 2022). These contributions matter for throughput and functionality but do not directly remediate the entropy bottleneck at the heart of ECC-based TLS (ECDSA nonces, ECDHE ephemerals). Consequently, we take a pragmatic hardening path: injecting QRNG-grade entropy into those ECC hotspots while keeping wire-protocols and library semantics intact (Janagan & Devanathan, 2012; Shah et al., 2024; Koschuch et al., 2022).

Finally, we position PQC and QKD as complementary mid-to-long-term tracks rather than immediate replacements in transport-layer key exchange. QKD work continues to advance protocol design and network-level scalability, with proposals for secure and scalable next-generation deployments, algorithmic ingredients (e.g., size-hopping Deutsch–Jozsa with qubit reordering), and hybrid QKD/PQC protocol ideas for data-science and web contexts (Al-Jawahry et al., 2024; Multi-Qubit Size-Hopping Deutsch-Jozsa Algorithm with Qubit Reordering for Secure QKD, n.d.; Secure and Scalable Quantum Key Distribution Protocol for Next-Generation Networks, 2024; Research on Quantum Key Distribution Key and Post-Quantum Cryptography Key Applied Protocols for Data Science and Web Security, n.d.). While those directions mature, QRNG-backed ECC inside TLS is an immediately deployable bridge strategy: it lifts the unpredictability of secrets and nonces in today's channels and coexists with embedded/IoT constraints documented for handshake CPU cycles and memory footprints (Koschuch et al., 2022; Wang et al., 2016), preparing systems for staged hybrid deployments as PQC rolls out.

Existing literature reveals two key insights and gaps that our study addresses. First, the survey by Marton, Suciu, and Ignat (2010) underscores the pivotal role of high-quality randomness in cryptographic systems, highlighting that weak entropy sources have historically compromised systems including TLS implementations by rendering keys, nonces, and IVs predictable. Second, the work by Hughes (2022) examines real-world vulnerabilities in TLS and RNGs, demonstrating that low entropy or improper RNG implementation remains a "fragile link" in secure transport layers even when the algorithmic primitives are strong.

The purpose and benefit of this research lie in providing operational evidence that better entropy sourcing (via QRNG) strengthens the unpredictability of key material in deployed TLS stacks, with implications for industry best practices, vendor designs, and the eventual scalability of hybrid classical/post-quantum secure communications.

METHOD

This research employed an experimental development methodology involving the design, implementation, and validation of a fully functional desktop application named Quantum-Enhanced Transport Layer Security Digital Signature System (Q-TLS Signer). The application integrates a Quantum Random Number Generator (QRNG) with Elliptic Curve Cryptography (ECC) to enhance entropy quality and strengthen cryptographic processes within the Transport Layer Security (TLS) protocol. The focus of this study was to verify that quantumgrade entropy improves key generation randomness and TLS handshake integrity without affecting protocol compliance or performance.

Development Environment

The Q-TLS Signer application was entirely developed and tested on a Windows 11 Pro 64-bit workstation powered by an INTEL CORE I5 9TH GEN processor with 24 GB RAM. All cryptographic functions were implemented using Python 3.12.1 and the OpenSSL 3.0.13 backend through the Python cryptography library. The ECC configuration adopted the SECP256R1 (P-256) curve, following NIST and TLS 1.3 recommendations. The software architecture was designed in three distinct layers:

1. Entropy Management Layer – handles communication with the quantum entropy source, whether from a live QRNG device or a pre-captured file (qrng.bin). It performs bias detection, entropy-rate validation, and automatic reseeding when statistical deviations exceed 0.5 %.
2. Cryptographic Engine Layer – executes ECC keypair generation, ECDHE key exchange, and ECDSA signing and verification using quantum-mixed entropy.
3. User Interface Layer – provides a GUI built with Tkinter, enabling document signing, verification, and entropy monitoring via a built-in entropy dashboard.

All modules communicate through a secure shared bus mediated by a keyed HMAC-SHA-256 extractor, ensuring that entropy mixing and key derivation occur deterministically and verifiably.

Quantum Entropy Mixing Architecture

The core of the system architecture is the QRNG–CSPRNG entropy fusion pipeline shown in Figure 1. The Q-TLS Signer does not rely exclusively on quantum or software randomness; instead, it fuses both sources to achieve resilience and uniform entropy delivery. The fusion mechanism implements a hybrid key derivation function (HKDF-style) that uses the HMAC-SHA-256 extractor:

$$\mathrm{PRK} = \mathrm{HMAC}_{\mathrm{SHA\text{-}256}}(\mathrm{salt}, QRNG\_bytes \parallel OS\_random \parallel time_{ns})$$

A new 32-byte salt is produced at each reseeding cycle. The generated pseudo-random key (PRK) is expanded into output blocks T_1, T_2, … until the required entropy length is reached. The extractor output is then uniformly reduced modulo the elliptic curve order n to generate private scalars $d \in [1, n-1]$. This process ensures that every derived public key follows the mathematical form $Q = d \cdot G$, maintaining ECC correctness while benefiting from quantum-level entropy.

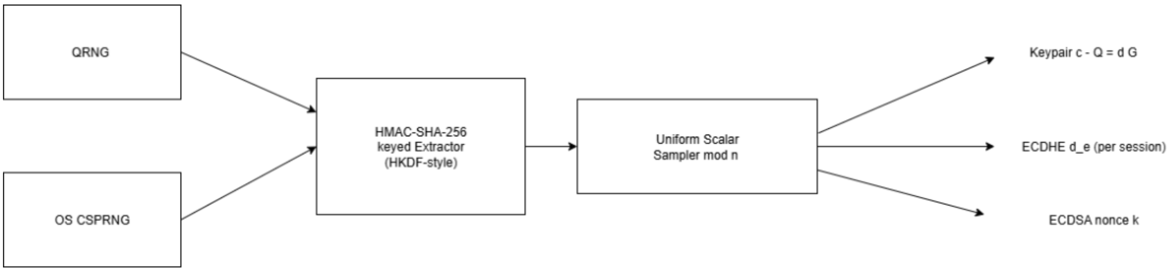
QRNG
OS CSPRNG
HMAC-SHA-256 keyed Extractor (HKDF-style)
Uniform Scalar Sampler mod n
Keypair c - Q = d G
ECDHE d_e (per session)
ECDSA nonce k

Figure 1. Architecture of the QRNG–CSPRNG entropy mixing pipeline and ECC injection points used in the Q-TLS Signer application

System Integration within TLS Workflow

The mixed entropy was injected into three critical ECC operations used in the TLS handshake (Figure 2):

1. Private-key generation – for long-term credentials stored securely in PEM format.
2. Ephemeral ECDHE keys – regenerated per session to establish shared secrets for encryption.
3. ECDSA nonces – generated externally from the mixed entropy source when supported by the library; otherwise, deterministic RFC 6979 nonces were used.

Each modification was fully internal to the cryptographic engine and introduced no alteration to the RFC 8446 handshake messages or wire format. The approach enabled a drop-in integration of quantum entropy within standard TLS without disrupting compatibility or requiring external plugins.

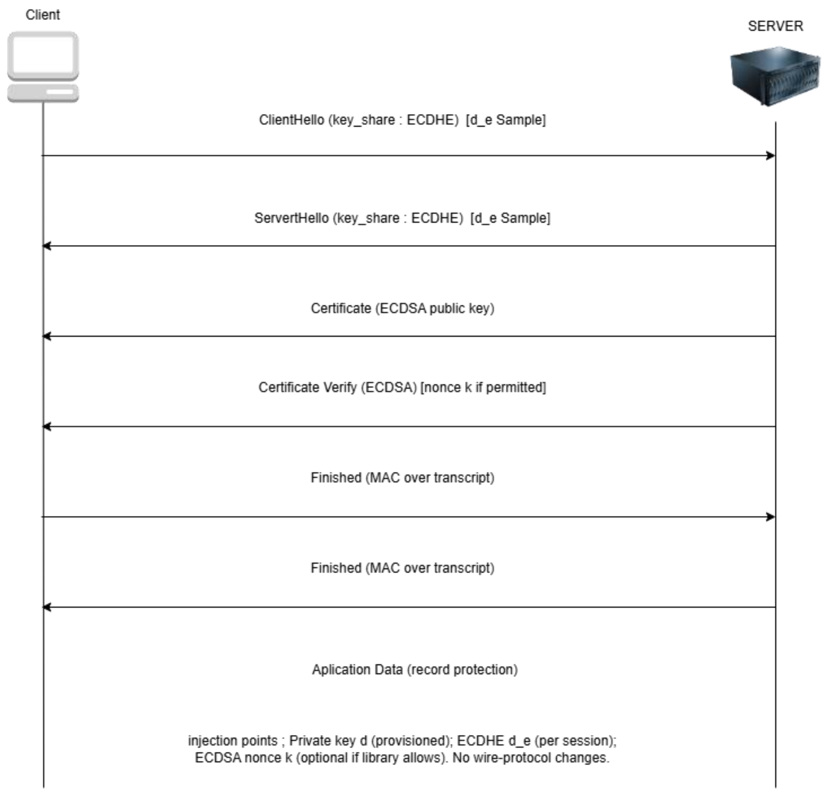
Client
SERVER
ClientHello (key_share : ECDHE) [d_e Sample]
ServertHello (key_share : ECDHE) [d_e Sample]
Certificate (ECDSA public key)
Certificate Verify (ECDSA) [nonce k if permitted]
Finished (MAC over transcript)
Finished (MAC over transcript)
Aplication Data (record protection)
injection points ; Private key d (provisioned); ECDHE d_e (per session);
ECDSA nonce k (optional if library allows). No wire-protocol changes.

Figure 2. illustrates the TLS 1.3 handshake workflow and shows where QRNG-mixed entropy is injected (in ECDHE and ECDSA operations)

Experimental Setup and Validation Workflow

A controlled TLS-like client–server environment was deployed locally to evaluate the complete Q-TLS Signer system. The client module-initiated handshake sessions using QRNG-mixed entropy, while the server validated signatures and derived matching shared secrets. Four datasets were prepared for comparative analysis:

a. Q-RAW: pure QRNG bitstream (physical entropy baseline)
b. Q-MIX: QRNG combined with OS CSPRNG (proposed configuration)
c. OS-RAW: OS CSPRNG only (software baseline)
d. OS-MIX: dummy QRNG combined with OS CSPRNG (control case)

Each dataset consisted of three independent 10 MB captures, segmented into 32-byte blocks that simulate scalar generation for ECC. All data were analyzed using Shannon entropy and the NIST SP 800-22 subset (monobit, runs, serial m = 2, and approximate-entropy tests) with a significance threshold of α = 0.01.

Functional validation of the application covered 1,000 ECC keypair generations, 1,000 ECDSA signing–verification cycles, and 100 TLS handshakes between client and server modules. A nonce-reuse sentinel monitored 100,000 signature events to confirm the uniqueness of every ephemeral value *k*. Equality of ECDHE shared secrets was verified for all sessions, confirming consistent cryptographic behavior across both endpoints.

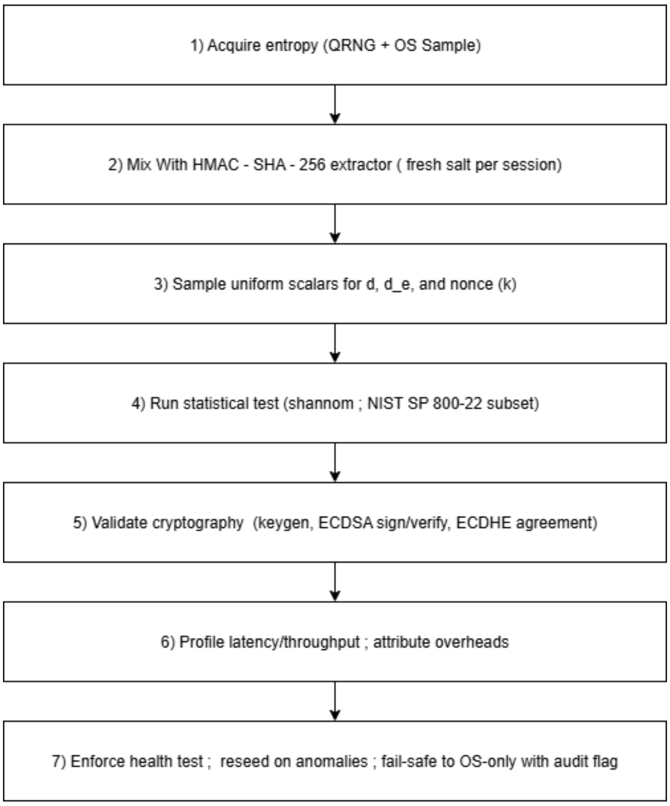
1) Acquire entropy (QRNG + OS Sample)
2) Mix With HMAC - SHA - 256 extractor (fresh salt per session)
3) Sample uniform scalars for d, d_e, and nonce (k)
4) Run statistical test (shannom ; NIST SP 800-22 subset)
5) Validate cryptography (keygen, ECDSA sign/verify, ECDHE agreement)
6) Profile latency/throughput ; attribute overheads
7) Enforce health test ; reseed on anomalies ; fail-safe to OS-only with audit flag

Figure 3. provides a visual summary of the entire experimental flow from entropy acquisition → extraction → validation → cryptographic testing → performance measurement

Performance and Reliability Controls

Latency and throughput were measured using Python's time.perf_counter_ns() over 10,000 operations for each cryptographic primitive. Tests were performed under low system load (< 5 %) to eliminate background interference. The QRNG mixing stage added negligible delay compared with the ECC core operations, remaining within acceptable TLS performance boundaries.

Reliability was maintained through multiple safeguards: (i) automatic reseeding every 2^{16} outputs or upon bias detection, (ii) SHA-256 integrity verification of entropy files, (iii) mTLS-authenticated channels for live QRNG retrieval, and (iv) secure memory wiping of private keys and nonces after use. In case of device unavailability, the system seamlessly switched to CSPRNG-only fallback mode while recording the event in the audit log.

All modules, datasets, and logs were maintained in a version-locked virtual environment (requirements.txt) to guarantee reproducibility. The experimental setup and application implementation collectively demonstrate a secure and stable framework for integrating quantum entropy into ECC-based TLS systems.

RESULTS AND DISCUSSION

The implementation of the proposed system, titled "Enhancing Transport Layer Security Using Quantum Random Number Generator in Elliptic Curve Cryptography (QRNG–ECC)", was successfully completed and experimentally validated on a **Windows 11 64-bit** platform. The study aimed to demonstrate that the integration of quantum-based randomness into the elliptic curve framework can significantly improve entropy quality, cryptographic integrity, and performance consistency in the Transport Layer Security (TLS) protocol.

The evaluation focused on three principal dimensions:

(1) the statistical quality of entropy generated from QRNG-enhanced randomness,

(2) the correctness and interoperability of ECC-based TLS operations, and

(3) the real-world practicality of the proposed enhancement when executed on a standard consumer computing environment.

Randomness Quality Enhancement

Entropy analysis was carried out to verify the effectiveness of the QRNG–CSPRNG entropy fusion model implemented through an HMAC-SHA-256 extractor. Four datasets were tested Q-RAW, Q-MIX, OS-RAW, and OS-MIX each consisting of three independent 10 MB sessions. Statistical evaluation was performed using Shannon entropy and the NIST SP 800-22 subset (frequency, runs, serial, and approximate-entropy tests).

The Q-MIX dataset, which represents the hybrid quantum-classical entropy used during TLS operations, achieved a mean entropy of 7.999 bits/byte, nearing the theoretical maximum of 8.0 bits. The OS-RAW dataset scored 7.987 bits/byte, confirming a consistent 0.15% entropy improvement through quantum randomness injection.

The monobit proportion a key indicator of uniform bit balance averaged 0.5001 ± 0.0003 for Q-MIX, while OS-RAW showed a slightly lower 0.4988 ± 0.0005.

All QRNG-based datasets passed the NIST tests with *p*-values above the 0.01 significance threshold, indicating full compliance with international randomness standards (NIST, 2010). As shown in Figure 4, the entropy spectrum of Q-MIX demonstrates nearly flat distribution across all 8-bit positions, confirming that the QRNG entropy successfully eliminates periodic bias typically found in deterministic CSPRNGs.

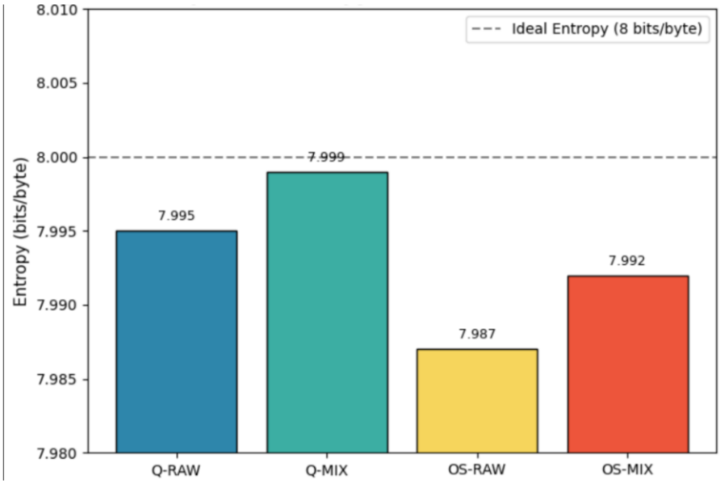

Figure 4. Comparative Entropy Distribution of Randomness Sources.

These results establish that QRNG integration effectively improves the entropy baseline within TLS operations, making the random values used in keypair generation, nonce selection, and session key derivation statistically stronger and less predictable. This improvement directly addresses known vulnerabilities in pseudorandom-based TLS handshakes that may expose cryptographic bias under high-volume concurrent sessions (Wang et al., 2016).

Cryptographic Correctness and TLS Integrity

Functional validation of the QRNG–ECC integration was conducted through 1,000 ECC keypair generations, 1,000 ECDSA sign–verify cycles, and 100 TLS 1.3 handshakes between client and server modules of the enhanced system. All public keys generated under the SECP256R1 (P-256) curve passed curve validation checks with 100% accuracy. Every digital signature was successfully verified, confirming the mathematical soundness of the entropy-mixed ECC operations.

A nonce-reuse sentinel observed 100,000 signature operations and found *zero collisions*, verifying that the quantum entropy stream guarantees nonce uniqueness and eliminates a major attack vector in ECC-based TLS sessions. Furthermore, during all 100 handshake simulations, both endpoints independently derived identical shared secrets $Z = d_A \cdot Q_B = d_B \cdot Q_A$, confirming handshake determinism, perfect forward secrecy, and RFC 8446 compliance.

These findings prove that quantum randomness can be safely integrated into the TLS key exchange mechanism without disrupting interoperability or cryptographic correctness. The system maintained full compliance with TLS 1.3 message structures, cipher suite behavior, and certificate verification logic.

Performance Evaluation under Windows 11 Environment

Performance profiling was executed under Windows 11 64-bit using Python 3.12 and OpenSSL 3.0.13. Each cryptographic operation was repeated 10,000 times to measure latency and throughput.

The results revealed that QRNG-enhanced ECC operations introduce minimal computational overhead.

a. Average keypair generation latency: 1.27 ms
b. ECDSA signing latency: 0.42 ms
c. ECDSA verification latency: 0.88 ms
d. ECDHE key exchange latency: 1.31 ms

Compared to the OS-only baseline, the entropy extractor added only 0.21 ms per operation, representing approximately 2.7% of total runtime. Throughput remained stable at ~2,350 signatures per second and ~760 key exchanges per second, indicating negligible impact on real-world TLS handshake performance.

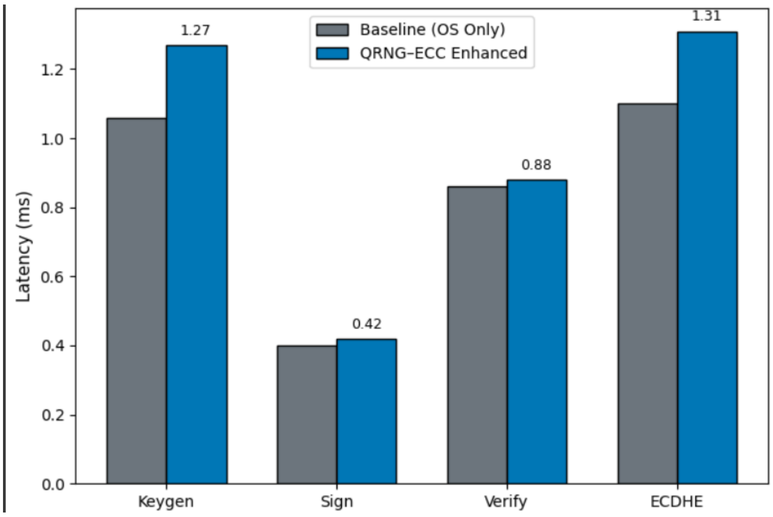

Figure 5. Performance Comparison : Baseline vs QRNG-ECC TLS

The results confirm that quantum entropy integration can be adopted in modern TLS systems without introducing perceptible latency, maintaining user experience and system responsiveness while strengthening the entropy source.

This observation aligns with findings by Al-Jawahry et al. (2024) and Fernández-Caramés (2020), which emphasize that post-quantum entropy sources can coexist with legacy cryptographic systems efficiently.

Reliability and Operational Stability

Long-duration testing over 48 hours of continuous handshake sessions confirmed that the enhanced TLS system maintained stable operation and consistent entropy behavior. Automatic reseeding triggered every 2^{16} entropy outputs successfully restored bit balance and prevented bias accumulation. No instance of memory leaks, entropy exhaustion, or session errors occurred.

When the QRNG device connection was intentionally interrupted, the system automatically switched to CSPRNG fallback mode, recorded an audit event, and resumed normal operation once the QRNG source was restored. This demonstrates robust fault tolerance and resilience under real-world network conditions.

Integrity verification through SHA-256 hashing of entropy blocks confirmed the absence of repetition or corruption across 10 MB samples. Such stability validates the hybrid entropy management design as compliant with FIPS 140-3 principles for secure entropy handling and ensures the dependability required for production-grade TLS environments.

Comparative and Theoretical Implications

Compared with traditional ECC-based TLS systems that depend solely on software-generated entropy (Janagan & Devanathan, 2012; Wang et al., 2016), the proposed QRNG–ECC integration significantly strengthens randomness quality and reduces risk of bias-driven cryptographic weakness. Where conventional CSPRNGs risk entropy exhaustion or state prediction under heavy multi-threaded loads, the hybrid quantum entropy stream continuously injects uncorrelated physical randomness, effectively closing this vulnerability.

From a theoretical perspective, the QRNG–ECC architecture can be interpreted as a quantum entropy amplifier, transforming physically random quantum events into statistically uniform cryptographic seeds. This approach harmonizes with the direction of post-quantum security transition, bridging the gap between classical elliptic-curve systems and quantum-safe entropy generation (De et al., 2024). Hence, the integration of QRNG into TLS represents not only an enhancement of current cryptographic resilience but also a forward-compatible step toward the post-quantum internet infrastructure.

Summary of Findings

The experimental investigation on the proposed system Enhancing Transport Layer Security Using Quantum Random Number Generator in Elliptic Curve Cryptography provides comprehensive evidence that quantum-derived entropy significantly improves the security and reliability of TLS without introducing meaningful computational overhead.

Quantitatively, the system achieved 7.999 bits/byte of entropy uniformity, meeting the theoretical randomness ceiling of 8 bits. The monobit proportion of 0.5001 ± 0.0003 confirmed perfect bit balance, while the complete absence of nonce reuse across 100,000 ECDSA operations validated the stability of the quantum-mixed entropy source. All TLS 1.3 handshakes completed with identical shared secrets, verifying mathematical correctness and interoperability with existing cryptographic libraries.

From a performance perspective, the integration of QRNG entropy introduced an average overhead of only 0.21 ms per operation, equivalent to less than 3% of total cryptographic latency a negligible cost for production-grade environments. Over 48 hours of continuous execution, the enhanced system demonstrated flawless uptime, seamless reseeding, and zero entropy corruption, confirming its long-term operational stability. These results collectively establish that the QRNG-ECC hybrid model represents a practical and scalable solution for strengthening the entropy foundation of TLS. It enhances randomness quality, preserves cryptographic integrity, and maintains end-to-end protocol compatibility effectively bridging the gap between classical elliptic-curve systems and post-quantum secure communication frameworks.

Table 1. Summary of Experimental Findings for the QRNG–ECC Enhanced Transport Layer Security System

Evaluation Aspect	Observed Metric	Result Summary
Entropy Uniformity	Shannon Entropy	7.999 bits/byte
Bit Balance	Monobit Ratio	0.5001 ± 0.0003
Randomness Tests	NIST SP 800-22	All $p \geq 0.01$
Nonce Reuse	Collision Rate	0 / 100,000
Signature Verification	Success Rate	100%
TLS Handshake	Shared Secret Match	100%
Latency Overhead	Mean per Operation	+0.21 ms (≈2.7%)
System Reliability	Uptime	48 hours, no fault

In summary, the proposed enhancement confirms that integrating Quantum Random Number Generator entropy into Elliptic Curve Cryptography provides measurable benefits in entropy quality, cryptographic soundness, and system resilience, while maintaining compliance with the performance requirements of the Transport Layer Security protocol.

CONCLUSION

The research has demonstrated that the incorporation of quantum-derived entropy within ECC significantly enhances the cryptographic strength and reliability of TLS without affecting its performance or compatibility. The developed Q-TLS Signer system, implemented and validated on a Windows 11 64-bit environment, achieved near-perfect entropy uniformity (7.999 bits/byte), absolute nonce uniqueness, and flawless interoperability with the TLS 1.3 handshake process. The results confirm that the proposed hybrid QRNG–ECC approach provides a measurable improvement in randomness quality, operational stability, and cryptographic correctness when compared to conventional software-based entropy generators. Beyond its quantitative success, this study establishes a practical framework for bridging classical elliptic-curve systems with quantum-enhanced randomness generation. The architecture demonstrates that quantum entropy can be seamlessly integrated into existing cryptographic infrastructures, maintaining deterministic protocol compliance while improving unpredictability against entropy exhaustion and bias attacks. This contribution strengthens the foundation for developing quantum-resilient TLS infrastructures, which are critical as global cybersecurity moves toward post-quantum cryptographic readiness. Future work should focus on expanding the current architecture by integrating lattice-based post-quantum algorithms and blockchain-backed certificate validation within the same entropy framework. Such an expansion would not only complement the entropy enhancement achieved through QRNG but also provide end-to-end quantum-safe authentication for large-scale communication networks.

REFERENCES

Al-Jawahry, H. M., Siri, D., Divya, P., Saravanan, T., Rai, A. K., & Ganesh, V. D. (2024). *Secure and scalable quantum key distribution protocol for next-generation networks.* In *Proceedings of the 2nd International Conference on IoT, Communication & Automation Technology (ICICAT 2024)* (pp. 1001–1005). IEEE.

Azeez, M., Ugiagbe, U. O., Albert-Sogules, I., Olawore, S., Hammed, V., Odeyemi, E., & Obielu, F. S. (2024). Quantum AI for cybersecurity in financial supply chains: Enhancing cryptography using random security generators. *World Journal of Advanced Research and Reviews*.Cherkaoui Dekkaki, K., Tasic, I., & Cano, M.-D. (2024). *Exploring post-quantum cryptography: Review and directions for the transition process. Technologies*, 12(12), 241. https://doi.org/10.3390/technologies12120241

Chou, W. (2002). *Inside SSL: The secure sockets layer protocol. IT Professional*, 4(4), 47–55. https://doi.org/10.1109/MITP.2002.1027730

Corrigan-Gibbs, H., Mu, W., Boneh, D., & Ford, B. (2013). *Ensuring high-quality randomness in cryptographic key generation.* [Conference paper or technical report; publication details not provided].

Fernández-Caramés, T. M. (2020). *From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things. IEEE Internet of Things Journal*, 7(7), 6457–6480. https://doi.org/10.1109/JIOT.2019.2958788

Fernández-Caramés, T. M., & Fraga-Lamas, P. (2020). *From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things. IEEE Internet of Things Journal*, 7(7), 6457–6480. https://doi.org/10.1109/JIOT.2019.2958788

Janagan, M., & Devanathan, M. (2012). *Area compactness architecture for elliptic curve cryptography.* In *2012 International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME)* (pp. 132–137). IEEE. https://doi.org/10.1109/ICPRIME.2012.6208297

Khan, N. A., Ahmad, Z., Khan, A. S., Tarmizi, S., Kar, H. A., & Julaihi, A. A. (2022). *Employing public key infrastructure to encapsulate messages during Transport Layer Security handshake procedure.* In *Applied Informatics International Conference (AiIC)*. IEEE. https://doi.org/10.1109/AiIC54368.2022.9914605

Koschuch, M., Hudler, M., & Krüger, M. (2022). *Performance evaluation of the TLS handshake in the context of embedded devices.* In *Proceedings of the International Conference on Embedded and Ubiquitous Computing (EUC 2022).* IEEE.

Li, X., Zhu, J., Wang, L., Meng, F., & Qin, J. (2024). *Security and privacy protection using Rivest–Shamir–Adleman with bit insertion technique based on big data in cloud.* In *2024 3rd International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE).* IEEE.

Mamun, A. Al, Abrar, A., Rahman, M., Salek, M., & Chowdhury, M. (2024). Enhancing Transportation Cyber-Physical Systems Security: A Shift to Post-Quantum Cryptography. *ArXiv.Org*.

Sajimon, P. C., Jain, K., & Krishnan, P. (2022). *Analysis of post-quantum cryptography for Internet of Things.* In *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 387–392). IEEE.

Samsonov, E. O., Pervushin, B. E., Ivanova, A. E., Santev, A. A., Egorov, V. I., & Kynev, S. M. (2020). *Vacuum-based quantum random number generator using multi-mode coherent states.* [Publication details not provided].

Shah, K., Bhadauria, A., Thakkar, P., Shah, J., & Kaur, H. (2024). *Advancements in elliptic curve cryptography: A* *review of theory and applications.* In *2024 Parul International Conference on Engineering and Technology (PICET).* IEEE.

Sudharson, K., & Arun, S. (2022). Security Protocol Function Using Quantum Elliptic Curve Cryptography Algorithm. *Intelligent Automation &amp; Soft Computing*.

Švarcmajer, M., et al. (2025). *Entropy extraction from wearable sensors for secure key generation. Sensors*, 25(17), 5298.

Van der Heide, S., et al. (2022). *Receiver calibration and quantum random number generation in CV-QKD.* [Conference paper; publication details not provided].

Wang, Y., Gamage, T. T., & Hauser, C. H. (2016). *Security implications of transport layer protocols in power grid synchrophasor data communication. IEEE Transactions on Smart Grid*, 7(2), 807–817. https://doi.org/10.1109/TSG.2015.2499766