Adaptive Multi-Algorithm Key Exchange for Quantum-Resilient Secure Communication: Dynamic Switching among QKD, Post-Quantum, and Classical Key Establishment with Entropy Fusion*

Ogbodo Tochukwu Hillary [1], Bilkisu Larai Muhammad-Bello [1], and Saleh El-Yakub Abdullahi [1]

[1]Department of Software Engineering, Faculty of Computing, Nile University of Nigeria, Abuja, Nigeria

Abstract

With the arrival of scalable quantum computers, classical key exchange protocols like RSA, elliptic-curve and finite-field Diffie–Hellman are vulnerable to harvest-now-decrypt-later attacks. Quantum key distribution offers information-theoretic security but is sensitive to channel noise, loss, and distance, while post-quantum cryptography provides quantum resistance on conventional hardware at the cost of larger keys and a dependence on hardware computational strength. Existing hybrid defenses generally rely on static configurations that require manual intervention when channel conditions degrade, and no prior software-defined system performs real-time three-way switching among these approaches while preserving uninterrupted key availability. This paper presents an adaptive multi-algorithm key generation and exchange framework that dynamically selects among quantum key distribution (BB84), post-quantum cryptography (Kyber512, standardized as ML-KEM-512), and classical Diffie–Hellman according to real-time monitoring of the quantum bit error rate and network latency, fusing key material from all active sources through an HMAC-based key derivation stage. The framework was implemented and evaluated in a controlled simulation environment built on Qiskit, liboqs, and the Python cryptography library. Across all five operating modes it attained a 100% key-generation success rate, with the quantum-resistant modes sustaining a secret-key throughput of approximately 3 kbps at a 256-bit key size and mode transitions completing without loss of key availability. A Kruskal–Wallis test confirmed that the timing differences among modes were statistically significant ($H = 133.32$, $p < 0.001$), and the security model was placed on a formal footing using the robust key-combiner framework. The results indicate that adaptive multi-algorithm key exchange can substantially improve the quantum resilience of secure communication systems in terms of security, availability, and performance.

Keywords Quantum key distribution · Post-quantum cryptography · Adaptive cryptography · Entropy fusion · Quantum-resilient communication · Key exchange

*Preprint. This manuscript is being prepared for submission to *Quantum Information Processing*. Simulation code and datasets are available from the corresponding author on reasonable request.

1 Introduction

Secure communication protocols are used in modern digital systems across finance, healthcare, telecommunications, and government. The security of classical schemes like Diffie–Hellman (DH), RSA and the Advanced Encryption Standard (AES) is based on assumptions of computational hardness [1]. This is under threat from the development of scalable quantum computers: Shor's algorithm efficiently factors large integers and solves discrete logarithms, which breaks protocols based on RSA and elliptic curves [2]. This can be used for harvest-now-decrypt-later attacks, where an attacker stores the ciphertext today and decrypts it when quantum hardware is available [3].

Quantum key distribution (QKD) is a method of information security based on the BB84 protocol, which is based on the no-cloning theorem and the uncertainty principle, making eavesdropping detectable because it disturbs the quantum states [4]. However, QKD requires secure quantum channels, and is sensitive to noise, attenuation and loss. Post-quantum cryptography (PQC) provides an alternative: lattice-based key-encapsulation mechanisms such as Kyber, standardized by NIST, resist both classical and quantum attacks on conventional hardware, though with larger keys and additional computation [5].

Adaptive classical–quantum frameworks aim to combine the information-theoretic security of QKD with the practicality of PQC and classical systems. Existing practice, however, relies largely on static configurations that employ QKD when the quantum channel is stable and revert manually to PQC or classical methods when disruptions occur. To date, no system integrates real-time three-way switching among QKD, PQC, and DH together with adaptive entropy fusion. This work addresses that gap with a software-defined framework that (i) monitors the quantum bit error rate (QBER) and latency in real time, (ii) automatically switches among QKD (BB84), PQC (Kyber512, standardized as ML-KEM-512), and classical DH, and (iii) applies HMAC-based entropy fusion to maintain continuous, high-entropy key production. The framework is implemented and evaluated using Qiskit, liboqs, and the Python cryptography library within a controlled network simulation, and is assessed on key-generation rate, switching latency, and security overhead.

2 Related Work

Hybrid classical–quantum approaches have been widely investigated. Erkilic et al. applied Gaussian post-selection in a simultaneous quantum–classical communication scheme for continuous-variable QKD, extending key rates and transmission distances under noise; the method is channel-specific and does not address switching among diverse cryptographic systems [6]. Raj and Rajawat proposed an encryption framework combining EdDSA/ECDH, ML-DSA/ML-KEM, and QKD for enterprise networks with improved processing time, but reported large key sizes and did not evaluate scalability across network conditions [7].

Pirandola et al. presented a scalable hybrid classical–quantum key-distribution blueprint that remained largely theoretical, with limited performance evaluation [8]. Blanco-Romero et al. conducted Docker-based experiments on sequential and parallel QKD–PQC hybrids interoperating with IPsec/IKEv2, reporting reduced latency and bandwidth usage but restricted device coverage and complex application interfaces [9]. Ricci et al. combined classical, quantum, and post-quantum keys on FPGA hardware with provable security, yet did not test dynamic network conditions or entropy fusion [10]. Collectively, these works establish the value of hybrid architectures while leaving open the need for a modular, software-only framework that performs real-time multi-source switching with fusion and uninterrupted key availability. Table 1 summarizes representative studies.

Table 1: Summary of representative related works

Study	Contribution	Limitation
Erkilic et al. [6]	Gaussian post-selection for CV-QKD	Channel-specific; no dynamic switching
Raj and Rajawat [7]	ECDH/ML-KEM/QKD hybrid framework	Large keys; scalability untested
Pirandola et al. [8]	Scalable hybrid key-distribution blueprint	Largely theoretical
Blanco-Romero et al. [9]	QKD–PQC hybrids in IPsec/IKEv2	Device-specific; complex interfaces
Ricci et al. [10]	Hybrid keys on FPGA hardware	No dynamic conditions; no fusion

3 Proposed Framework

The proposed adaptive key-management framework operates as a closed-loop feedback system comprising five stages: real-time metric monitoring, a decision engine, mode-switching logic, entropy fusion, and key generation. Figure 1 shows the overall architecture. The system continuously evaluates channel conditions and selects the highest-security cryptographic mode the environment can support, preserving both security and availability as conditions change.

3.1 Real-time metric monitoring

Two metrics drive adaptation. The QBER is a measure of noise in the quantum channel which ranges from 0 to 1, with lower values representing a cleaner channel. Network latency is measured as a round triple delay in milliseconds. Both values are updated so it reflects realistic channel behavior which is gradually changed over time.

3.2 Threshold-based classification

Instead of continuous normalization every metric is distributed into four tiers simplifying and clarifying mode selection. QBER tiers are Excellent (≤ 0.03), Acceptable (0.03–0.08), Viable (0.08–0.11) and Non-viable (≥ 0.11), a QBER at or above 0.11 exceeds the BB84 theoretical security threshold and renders QKD insecure [11]. Latency tiers are Excellent (< 100 ms), Acceptable (100–200 ms), Viable (200–500 ms) & non-viable (≥ 500 ms). Table 2 enlists the latency thresholds & their interpretation.

Table 2: Latency thresholds and operational interpretation

Tier	Threshold	Interpretation
Excellent	< 100 ms	Supports all modes, including QKD
Acceptable	100–200 ms	Supports QKD, PQC, and DH
Viable	200–500 ms	Single mode only; high delay
Non-viable	≥ 500 ms	Fallback to classical only

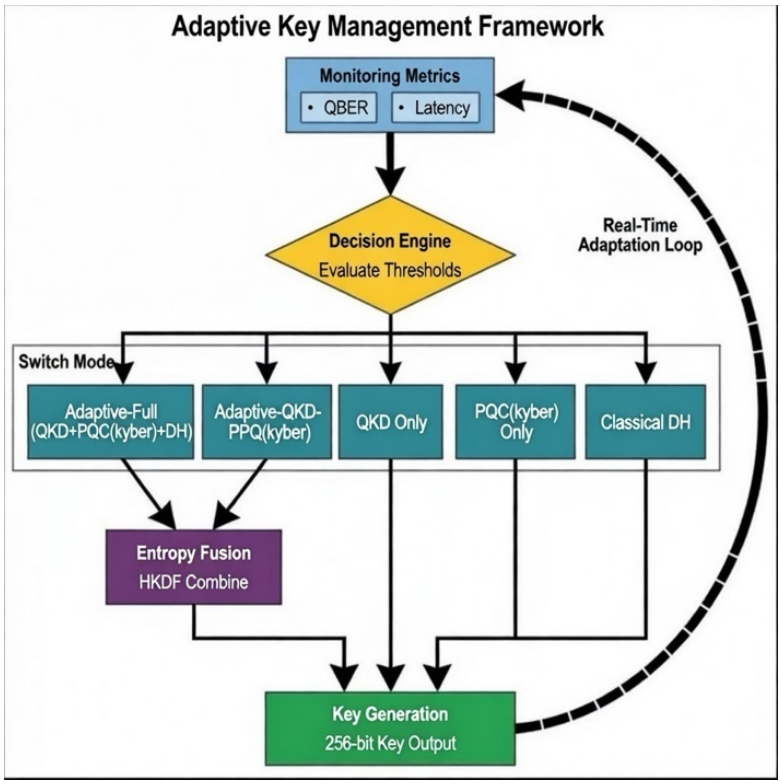


Figure 1: Adaptive multi-algorithm framework operating as a real-time feedback loop over QBER and latency

3.3 Decision engine

The decision engine is the central intelligence of the system. A hierarchical decision tree is present in its mode selection, which help to prioritize security while checking into account performance constraints. When both metrics are excellent, the engine selects Adaptive-Full mode (QKD + PQC + DH); as conditions degrade it steps down through Adaptive-QKD-PQC, QKD-only, PQC-only, and finally classical DH as a last-resort fallback. Figure 2 depicts the sixteen-branch decision logic. A key property is that when QKD becomes non-viable due to excessive error, the engine prefers Kyber512 (128-bit quantum resistance) over classical DH (112-bit, quantum-vulnerable) whenever the channel remains usable.

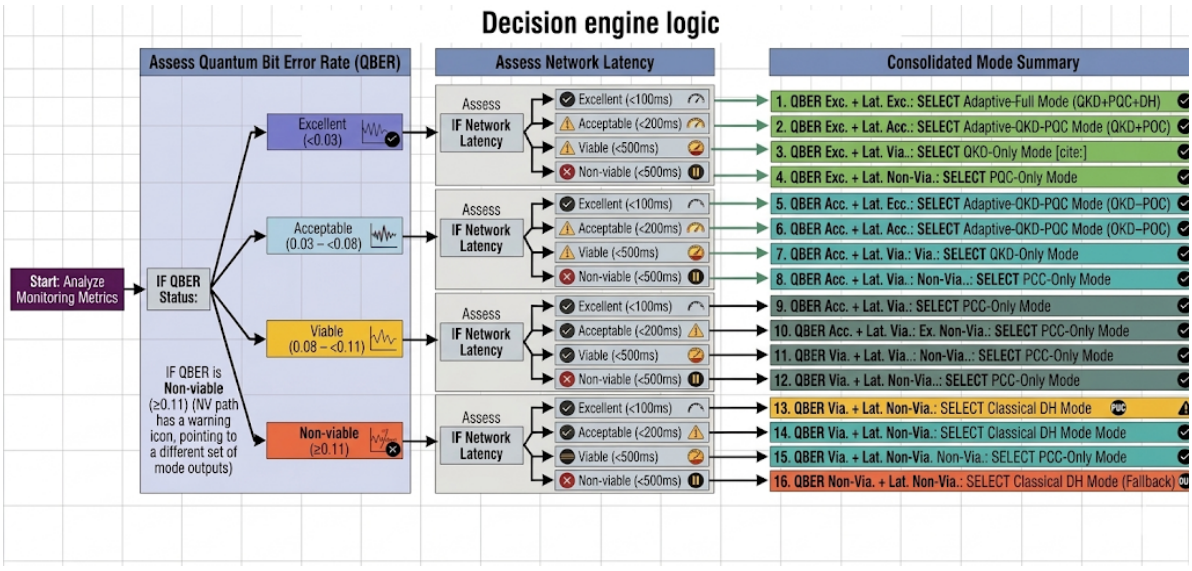


Figure 2: Decision-engine logic mapping each QBER × latency combination to the selected cryptographic mode

3.4 Key-generation modules

Four cryptographic primitives underpin the modes. The QKD module implements BB84 on Qiskit's qasm_simulator: Alice encodes random bits in random bases, Bob measures in independently chosen bases, and after basis sifting the retained bits form the raw key, with a random sample reserved for QBER estimation. The PQC module uses the Kyber512 key-encapsulation mechanism (standardized as ML-KEM-512 in NIST FIPS 203) from liboqs, at NIST Post-Quantum Security Category 1. The classical module performs 2048-bit Diffie–Hellman using cached parameters, providing an estimated 112-bit classical security level. Two composite modes Adaptive-QKD-PQC and Adaptive-Full combine these sources for defense in depth. Figure 3 illustrates the BB84 process and Fig. 4 the end-to-end key-generation pipeline.

3.5 Entropy fusion

Key material from all active sources is combined by a hybrid key-derivation stage implementing the HMAC-based key derivation function (HKDF) with SHA-256. Raw key bytes are concatenated, extracted with a 32-byte salt, and expanded into a single 256-bit output key, with the HKDF info

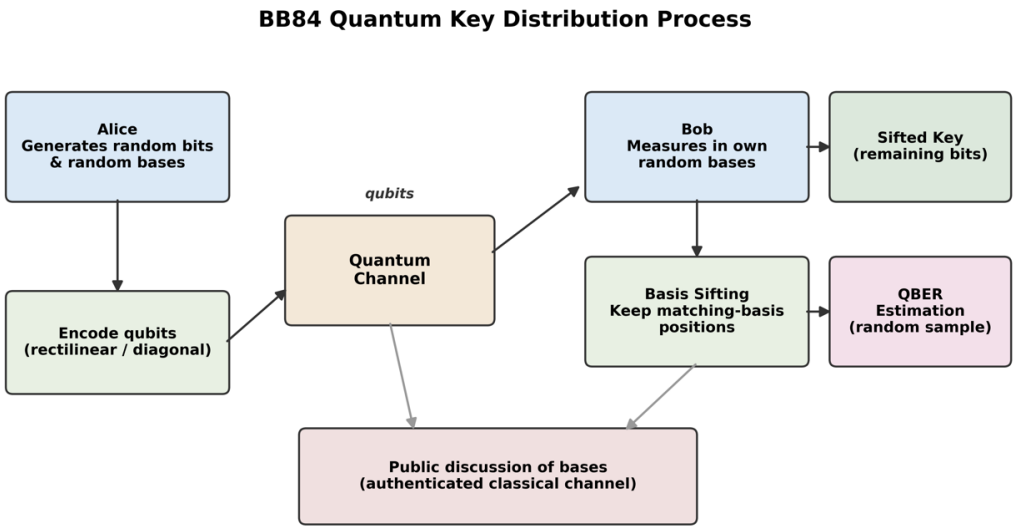


Figure 3: BB84 quantum-key-distribution process: encoding, transmission, sifting, and QBER estimation

parameter providing domain separation between modes. Randomness quality is assessed through the Shannon entropy of the key material,

$$H(X) = -\sum_i p(x_i)\ \log_2 p(x_i) \tag{1}$$

where p(xi) is the probability of byte value i. Byte data has a maximum of 8 bits/byte; the framework targets values above 4.8 bits/byte for the short 32-byte keys.

3.6 Security model of the hybrid modes

The security level credited to each mode is grounded in the established theory of robust key-encapsulation-mechanism (KEM) combiners rather than an ad hoc scoring rule. For the standalone modes, the levels are those of the underlying primitives: QKD (BB84) provides information-theoretic security in principle, proven under ideal conditions by Shor and Preskill [12]; the post-quantum mode uses Kyber512 (standardized as ML-KEM-512), which meets NIST Post-Quantum Security Category 1, equivalent to approximately 128-bit strength [13]; and the classical mode uses 2048-bit Diffie–Hellman, which NIST estimates at a 112-bit security strength [14].

For the adaptive hybrid modes, the framework derives the shared key by combining the key material of all active sources through an HKDF (HMAC-SHA-256) extract-and-expand step. This construction follows the dual-PRF combiner design of Bindel et al. [15], for which the combined key inherits the security of the strongest component input under the combiner's dual-PRF assumptions that HKDF satisfies, since HMAC has been validated as a dual pseudorandom function. The practical consequence is the "hybrid property" adopted in IETF TLS 1.3 hybrid key exchange: the combined secret is secure as long as at least one component remains unbroken [16].

Accordingly, each hybrid mode is credited with a conservative security level of at least 128 bits of post-quantum security, guaranteed by the Kyber512 component alone and preserved even if the classical Diffie–Hellman component is broken by a future quantum adversary and irrespective of the state of the quantum channel. This lower bound follows directly from the robust-combiner property: because Kyber512 contributes a post-quantum-secure input to the HKDF combiner, the

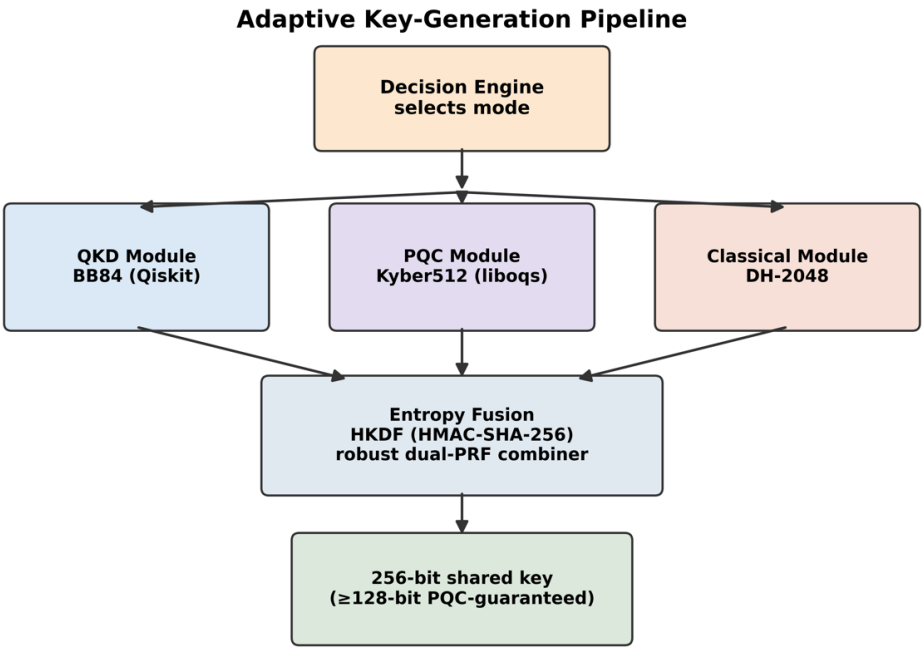


Figure 4: Key-generation pipeline from mode selection through per-module generation to HKDF fusion and 256-bit output

hybrid key is no weaker than Kyber512 regardless of the other components. The information-theoretic contribution of QKD can raise the effective security further where a physical quantum channel is available; in the present simulation QKD is modeled rather than realized on hardware, so the framework's guaranteed claim rests on the post-quantum floor. The asterisk in Table 4 denotes this PQC-guaranteed floor.

4 Methodology and Experimental Setup

The framework was implemented entirely in Python within a Jupyter Notebook and evaluated over a simulated point-to-point link between two parties operating across a quantum channel and a classical authenticated channel. No physical quantum hardware was used; QKD was simulated for repeatable evaluation.

4.1 Network-condition simulation

To model realistic channel behavior, zero-mean Gaussian perturbations of the QBER and latency were applied per cycle, which resulted in temporally correlated drift and not erratic jumps. The Gaussian increments for each cycle had a standard deviation of 0.02 for QBER and 30 ms for latency. The QBER update was constrained to [0, 0.25] and latency was constrained to [50, 650]ms, a well-connected local link and a severely degraded long-distance link, respectively. More formally, each metric is incremented by a gaussian at time t + 1, and then capped at these limits. Since this framework is aimed at a dynamic network environment, the simulation intentionally models continually changing conditions; if the results were the same for each run, it would not be consistent with a dynamic network environment. To make the stochastic components reproducible within a consistent range, a fixed random seed was therefore set, and absolute timing values reflect real

wall-clock performance and have minor run-to-run variations. The evaluation therefore does not measure absolute numeric results of the decision engine, but whether or not it is always selecting the best available security mode when necessary.

4.2 Benchmarking configuration

30 iterations were run on each mode to get good timing statistics. To compensate for the approximately 50% loss from basis sifting, BB84 generated four times the desired key length, with up to 50 bits reserved for QBER estimation, when a 256-bit key was desired, 1024 raw qubits were generated. All modes produced a uniform 32-byte (256-bit) output to keep timing comparisons fair, and DH parameters were precomputed once at startup. Five entropy samples per mode were collected, and three failover scenarios one per component failure verified graceful degradation. For each iteration the framework recorded key-generation time, Shannon entropy, security level, and success or failure, from which the mean, standard deviation, extrema, and 95th percentile were computed. Table 3 lists the tools used.

Table 3: Tools and technologies

Component	**Tool / library**	**Purpose**
Quantum simulation	Qiskit, Qiskit-Aer	BB84 protocol
Post-quantum	liboqs-python	Kyber512 KEM
Classical cryptography	Python cryptography	DH exchange, HKDF
Data analysis	NumPy, Pandas	Metrics processing
Visualization	Matplotlib, Seaborn	Performance charts
Statistics	SciPy	Significance testing

4.3 Evaluation metrics

Four metrics were used. Key-generation time is the interval between mode selection and key availability. Switching latency is the overhead incurred when changing modes after a threshold breach. Security level is the equivalent bit strength of the active mode. For the standalone modes it is the strength of the underlying primitive; for the hybrid modes it is derived from the robust-combiner property (Section 3.6), which credits the combined key with a conservative post-quantum-guaranteed floor rather than an ad hoc numerical bonus. Entropy quality is the measured Shannon entropy relative to the 8 bits/byte maximum.

5 Results

5.1 Framework operation and module verification

The framework initialized successfully and completed key generation across all five modes. During a representation session, the engine choses Adaptive-Full in good conditions & Adaptive-QKD-PQC whenever conditions are degraded, keeping the track of clean transitions along with no lost keys. Each individual module BB84, Kyber512, DH-2048, HKDF fusion, and the decision engine passed independent verification, producing valid 256-bit outputs.

5.2 Key-generation time and security

Table 4 reports per-mode performance over 30 iterations of a seeded run. PQC (Kyber) was fastest at 0.034 ms mean over three orders of magnitude faster than QKD while QKD required 77.85 ms owing to quantum-circuit simulation but delivered information-theoretic security. Adaptive-Full incurred the highest generation time (85.08 ms) in exchange for three-layer protection. Every mode achieved a 100% success rate. Figure 5 plots the security-versus-performance trade-off.

Table 4: Performance analysis of key-exchange modes (seeded run, 30 iterations per mode)

Mode	Mean (ms)	Std (ms)	95th %ile (ms)	Security (bits)	Success (%)
QKD	77.85	5.79	79.51	256	100
PQC (Kyber)	0.034	0.001	0.035	128	100
Classical DH	8.00	0.16	8.30	112	100
Adaptive-QKD-PQC	77.97	3.28	81.76	≥128*	100
Adaptive-Full	85.08	1.68	88.17	≥128*	100

*Hybrid modes are credited with the conservative post-quantum-guaranteed floor (≥128-bit) established by the robust-combiner analysis in Section 3.6; the QKD component can raise this further with a physical quantum channel.

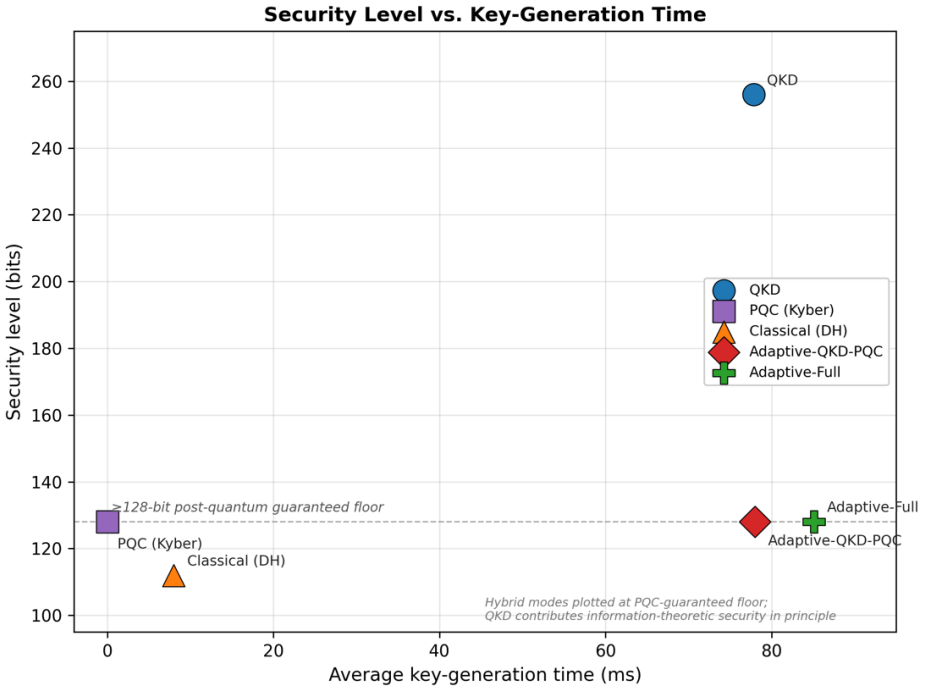


Figure 5: Security level versus average key-generation time across the five modes; hybrid modes plotted at the 128-bit post-quantum-guaranteed floor

5.3 Entropy quality

All modes produced mean entropy above 4.8 bits/byte (approximately 61% of the theoretical maximum for 32-byte keys), confirming strong randomness. Differences among modes—ranging from

4.844 bits/byte (PQC) to 4.898 bits/byte (Classical and Adaptive-QKD-PQC)—fell within the statistical variation expected at this sample size and reflect the use of cryptographically secure key material.

5.4 Dynamic switching behavior

Over 100 continuous key generations the framework performed 27 mode switches (a 27% switch rate) with no generation failures. In this representative seeded run the mode-usage distribution was QKD 34%, Adaptive-QKD-PQC 24%, Adaptive-Full 23%, and PQC (Kyber) 19%, with the classical-only fallback not selected because channel conditions remained viable throughout. The exact distribution varies with the simulated conditions, but across runs the engine consistently prioritizes the strongest viable mode and resorts to classical Diffie–Hellman only when no quantum-resistant option is supportable. Figure 6 shows mode transitions tracking simulated QBER and latency over time.

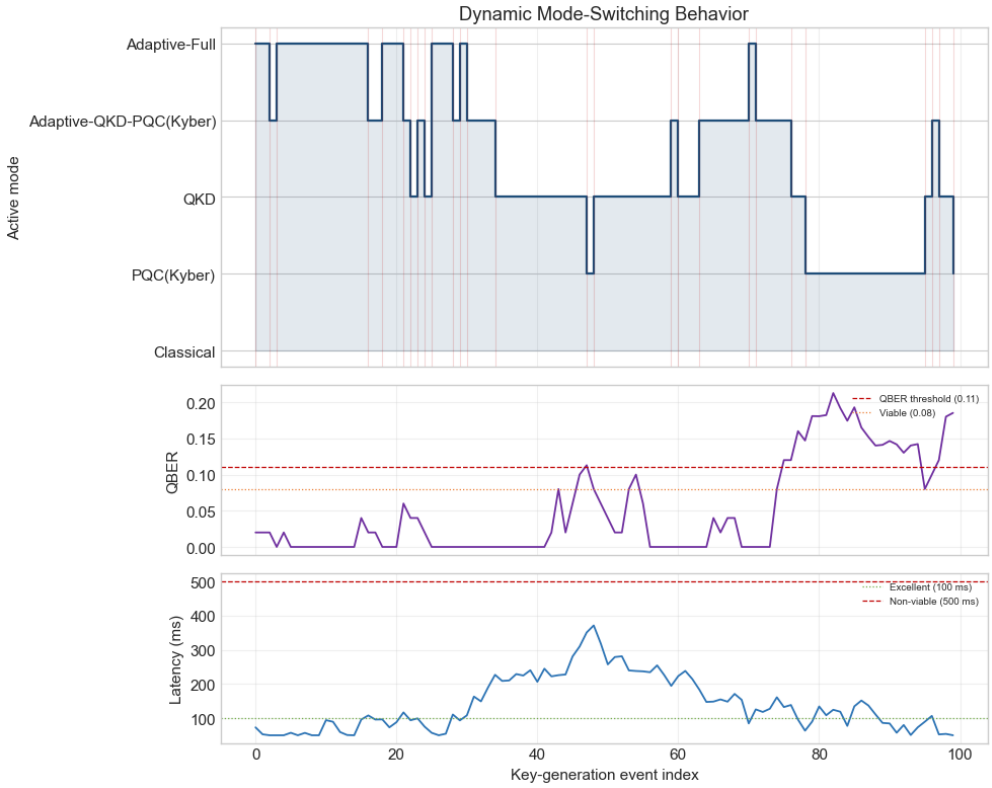


Figure 6: Dynamic mode-switching behavior over 100 key-generation events as QBER and latency drift

5.5 Optimization and statistical significance

A separate optimization benchmark measured the effect of precomputing the 2048-bit Diffie–Hellman parameters rather than regenerating them on every exchange. Because parameter generation involves a randomized search for safe primes, its cost is highly variable (tens of seconds, with a standard deviation comparable to the mean); precomputation reduced generation time to a few milliseconds an improvement of over two orders of magnitude, exceeding 2000× for the classical mode and 140× for Adaptive-Full. A Kruskal–Wallis H test chosen over one-way ANOVA

because the timing distributions were non-normal yielded $H = 133.32$ with $p < 0.001$ across all modes, indicating highly significant differences. Nine of the ten pairwise comparisons were significant at $p < 0.001$ with large effect sizes (Cohen's d up to 71.5); the single exception was QKD versus Adaptive-QKD-PQC ($p = 0.66$, $d = 0.03$), which is expected because the near-instant Kyber component adds negligible latency to the QKD-dominated cost. This confirms that the observed performance differences are genuine rather than sampling artifacts, while correctly identifying the two QKD-based modes as statistically indistinguishable in timing.

6 Discussion

The results support prioritizing security over raw speed in key generation. The additional milliseconds required by quantum-resistant modes are negligible against the consequences of a compromised key, and the modest overhead of the hybrid modes relative to standalone QKD, together with mode transitions that complete without loss of key availability shows that the security benefit does not carry operationally excessive cost. Notably, the QKD and Adaptive-QKD-PQC modes are statistically indistinguishable in timing, so the post-quantum security layer is obtained essentially for free on top of QKD. Although classical DH is computationally cheaper, its vulnerability to Shor's algorithm disqualifies it as a standalone long-term solution, and the framework correctly relegates it to a last-resort fallback.

The framework operates at the key-agreement layer of the protocol stack the same phase where ephemeral elliptic-curve Diffie–Hellman operates in a TLS 1.3 handshake and produces a 256-bit symmetric key suitable for AES-256-GCM. It does not itself encrypt application data; that remains the role of standard symmetric ciphers. This scoping makes the design a drop-in key-exchange component for protocols such as TLS and IPsec. The principal limitations are the reliance on simulated rather than physical QKD channels and evaluation confined to a point-to-point topology; both are natural targets for future validation on metropolitan-distance fiber links and multi-hop networks. A further refinement concerns the combiner itself: the present HKDF stage binds component key material but not the component ciphertexts, and incorporating ciphertext binding to achieve full IND-CCA robustness against active adversaries is a natural next step for deployment beyond the simulated setting.

7 Conclusion

This paper presented a software-defined adaptive framework that performs real-time three-way switching among QKD, PQC (Kyber512, standardized as ML-KEM-512), and classical Diffie–Hellman with HKDF-based entropy fusion. In a seeded simulation the framework achieved a 100% key-generation success rate, sustained approximately 3 kbps of secret-key throughput in its quantum-resistant modes at a 256-bit key size, and completed mode transitions without loss of key availability, with a decision engine that consistently prioritized the strongest viable quantum-resistant mode. Its security model was placed on a formal footing through the robust key-combiner framework, under which the hybrid modes inherit a conservative post-quantum-guaranteed floor. Future work will validate the threshold logic on a physical metropolitan-distance QKD link, align the key-derivation stage with emerging hybrid-KEM standards (e.g., NIST SP 800-227, ETSI GS QKD 014) to ease integration into TLS 1.3 and IPsec, pursue FIPS 140-3 evaluation of the derivation modules, expose decision telemetry for operational monitoring, and replace the static thresholds with lightweight predictive models to make switching proactive across multi-hop, wide-area deployments.

Statements and Declarations

Funding No funding was received to assist with the preparation of this manuscript.

Competing interests The author has no relevant financial or non-financial interests to disclose.

Data availability The simulation code and datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] National Institute of Standards and Technology: Recommendation for using cryptographic standards in the federal government: cryptographic mechanisms. NIST SP 800-175B Rev. 1 (2020)

[2] Barrett-Danes, F., Ahmad, F.: Quantum computing and cybersecurity: a systematic review of emerging threats, post-quantum solutions, and research directions (2019–2024). Discov. Appl. Sci. 7(11), 1083 (2025)

[3] Beder, C.: Post-quantum cryptography: a call to action. ISACA (2025). https://www.isaca.org/resources/news-and-trends/industry-news/2025/post-quantum-cryptography-a-call-to-action

[4] Barney, N., Gillis, A.S.: What is quantum key distribution? TechTarget (2025). https://www.techtarget.com/searchsecurity/definition/quantum-key-distribution-QKD

[5] Chundury, S., et al.: Scaling hybrid quantum–HPC applications with the quantum framework. In: Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (2025)

[6] Erkilic, O., Shajilal, B., Zaunders, N., Ralph, T.C.: Software-enhanced simultaneous quantum–classical communication protocol with Gaussian post-selection. arXiv:2510.13138 (2025)

[7] Raj, A., Rajawat, A.S.: A hybrid encryption framework combining classical, post-quantum, and QKD methods. arXiv:2509.10551 (2025)

[8] Pirandola, S., Lukens, J.M., Peters, N.A.: Hybrid classical–quantum communication networks. Prog. Quantum Electron. 99, 100551 (2025)

[9] Blanco-Romero, J., et al.: Hybrid quantum security for IPsec. arXiv:2507.09288 (2025)

[10] Ricci, S., Dobias, P., Malina, L., Hajny, J., Jedlicka, P.: Hybrid keys in practice: combining classical, quantum and post-quantum cryptography. IEEE Access 12, 23206–23219 (2024). https://doi.org/10.1109/ACCESS.2024.3364520

[11] Hayek, R.J., Chung, J., Kettimuthu, R.: A review of software for designing and operating quantum networks. arXiv:2510.00203 (2025)

[12] Shor, P.W., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Phys. Rev. Lett. 85(2), 441–444 (2000). https://doi.org/10.1103/PhysRevLett.85.441

[13] National Institute of Standards and Technology: Module-lattice-based key-encapsulation mechanism standard. FIPS 203 (2024). https://doi.org/10.6028/NIST.FIPS.203

[14] Barker, E.: Recommendation for key management: Part 1 – General. NIST SP 800-57 Part 1 Rev. 5 (2020). https://doi.org/10.6028/NIST.SP.800-57pt1r5

[15] Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., Stebila, D.: Hybrid key encapsulation mechanisms and authenticated key exchange. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography (PQCrypto 2019). LNCS, vol. 11505, pp. 206–226. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_12

[16] Giacon, F., Heuer, F., Poettering, B.: KEM combiners. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography (PKC 2018). LNCS, vol. 10769, pp. 190–218. Springer (2018). https://doi.org/10.1007/978-3-319-76578-5_7

[17] Bellare, M., Lysyanskaya, A.: Symmetric and dual PRFs from standard assumptions: a generic validation of an HMAC assumption. Cryptology ePrint Archive, Report 2015/1198 (2015). https://eprint.iacr.org/2015/1198

[18] Stebila, D., Fluhrer, S., Gueron, S.: Hybrid key exchange in TLS 1.3. IETF Internet-Draft draft-ietf-tls-hybrid-design (2023)

[19] Krawczyk, H., Eronen, P.: HMAC-based extract-and-expand key derivation function (HKDF). RFC 5869 (2010). https://doi.org/10.17487/RFC5869

[20] National Institute of Standards and Technology: Transitioning the use of cryptographic algorithms and key lengths. NIST SP 800-131A Rev. 2 (2019). https://doi.org/10.6028/NIST.SP.800-131Ar2

[21] Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: Proc. IEEE Int. Conf. Computers, Systems and Signal Processing, pp. 175–179 (1984)

[22] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997). https://doi.org/10.1137/S0097539795293172