# CrypTon: A Hybrid Quantum-Classical Framework Integrating BB84 Quantum Key Distribution with AES for Secure Communication

1st Vishesh Goyal
*Manipal Institute of Technology Bengaluru*
Manipal Academy of Higher Education, Manipal, India
visheshvishu1@gmail.com

2nd Raguru Jaya Krishna
*Manipal Institute of Technology Bengaluru*
Manipal Academy of Higher Education, Manipal, India
raguru.krishna@manipal.edu

*Abstract*—The emergence of quantum computing poses a significant threat to classical cryptographic systems, particularly those relying on mathematical hardness assumptions for key exchange. While symmetric algorithms such as the Advanced Encryption Standard (AES) remain efficient and widely deployed, their long-term security ultimately depends on the secrecy of keys. In this paper, we present a hybrid framework that integrates quantum key distribution (QKD) with AES and DES to enhance key security without altering cipher internals. Using the BB84 protocol as the quantum key generator, we implement a simulation-based system in Qiskit and evaluate it against classical pseudo-random key generation.

Our methodology covers five phases: establishing AES/DES baselines, generating quantum random keys, implementing BB84 with basis reconciliation and error detection, integrating quantum keys with AES/DES, and benchmarking across varying key sizes and file sizes. Results demonstrate that AES encryption throughput remains stable (around 250–370 MB/s) regardless of key source, while QKD introduces measurable latency in key generation (around 40–110 seconds for 128–256 bits). Importantly, adversarial interference is reliably detected, with 97.3% accuracy in eavesdropper identification.

The findings confirm that while classical systems excel in speed and scalability, QKD offers unmatched, physics-backed security assurances. Our study highlights the practicality of hybrid quantum-classical encryption, providing a foundation for secure communication in the post-quantum era.

## I. Introduction

Cryptography is the cornerstone of modern information security, enabling confidentiality, authentication, and integrity of digital communication. Classical cryptography relies on mathematical problems such as integer factorization or discrete logarithms for security guarantees. Symmetric-key algorithms like the Advanced Encryption Standard (AES) are widely deployed due to their efficiency and robustness, securing applications ranging from online banking to secure messaging [1].

AES, standardized by NIST in 2001, is a block cipher that supports key sizes of 128, 192, and 256 bits. It provides strong resistance against brute-force attacks owing to its large key space, while maintaining efficient performance across hardware and software implementations [2]. Despite its strength, the overall security of AES-based systems also depends on the secrecy of the key. Traditionally, keys are generated and exchanged using pseudo-random number generators (PRNGs) and classical cryptographic protocols such as RSA or Diffie-Hellman. However, the advent of quantum computers threatens these mechanisms: Shor's algorithm can efficiently break RSA and ECC, while Grover's algorithm reduces the effective key strength of symmetric ciphers like AES [3] [4].

Quantum cryptography has emerged as a promising alternative for secure key distribution. Unlike classical methods, its security is based on the fundamental laws of quantum mechanics rather than computational hardness assumptions. Quantum Key Distribution (QKD), first proposed by Bennett and Brassard in 1984 (BB84), allows two parties to establish a shared secret key with guaranteed eavesdropping detection [5]. By exploiting principles such as the no-cloning theorem and measurement disturbance, QKD ensures that any interception attempt by an adversary introduces detectable errors in the quantum channel [6].

This work explores the integration of QKD with classical symmetric encryption algorithms such as AES and DES. The hybrid approach leverages the proven efficiency of AES for bulk data encryption while enhancing security with quantum-generated keys. We present a comparative study of encryption performance using keys generated classically versus quantum-mechanically, considering factors such as key size, file size, and overall end-to-end performance.

## II. Related Work

The Advanced Encryption Standard (AES) has been extensively studied since its adoption as the U.S. federal standard in 2001. Daemen and Rijmen [2] introduced the Rijndael cipher, which became the basis for AES, demonstrating its resistance to differential and linear cryptanalysis. Subsequent studies have confirmed AES as a secure and efficient block cipher for both hardware and software implementations. For instance, Hodjat and Verbauwhede [7] explored high-throughput hardware architectures for AES, achieving significant performance gains in FPGA platforms. More recent work by Schwabe and Stoffelen [8] optimized AES on embedded systems, highlighting its adaptability to constrained devices.

Cryptographic security traditionally relies on the computational hardness of problems such as integer factorization or discrete logarithms. Diffie and Hellman [9] pioneered the concept of public-key cryptography through their key exchange protocol, which remains a foundation for secure communications. RSA, introduced by Rivest et al. [10], became a dominant public-key system for decades, though its security depends heavily on large key sizes. However, as Stallings [1] notes in his survey of modern cryptographic practices, the advent of quantum algorithms threatens these schemes, motivating exploration of alternatives such as post-quantum cryptography and QKD.

Quantum cryptography offers information-theoretic security by leveraging physical laws rather than mathematical hardness assumptions. Bennett and Brassard [5] first proposed the BB84 protocol, establishing the foundation for secure quantum key distribution. Since then, significant experimental progress has been made. Gisin et al. [6] provided a comprehensive review of the field, emphasizing advances in photon sources, detectors, and channel implementations. Scarani et al. [11] extended this discussion, categorizing protocols and identifying security loopholes such as photon-number-splitting attacks, highlighting the need for practical countermeasures.

Modern research has focused on extending QKD to real-world networks. Lo et al. [12] demonstrated the feasibility of practical QKD by introducing decoy-state protocols to counter multi-photon vulnerabilities. Sasaki et al. [13] reported a Tokyo QKD network, the first metropolitan-scale deployment integrating multiple QKD links and trusted nodes. More recently, Boaron et al. [14] achieved long-distance QKD exceeding 400 km using ultra-low-noise detectors, showcasing the scalability potential of the technology. These works illustrate both the promise and current challenges of QKD, particularly regarding speed, cost, and integration with classical systems.

Our work builds upon these foundations by experimentally comparing classical and quantum-assisted key generation for AES-based encryption. While prior literature has analyzed QKD protocols and AES implementations separately, few studies directly evaluate the performance and practicality of integrating quantum-generated keys into widely deployed classical cryptosystems.

## III. PROPOSED METHODOLOGY

In this work, we propose a hybrid cryptographic framework that integrates Quantum Key Distribution (QKD) with the Advanced Encryption Standard (AES) to enhance the security of classical encryption schemes against quantum adversaries. The methodology can be divided into three phases: Key Generation, Encryption/Decryption, and Performance Evaluation. A high-level flow of the system is illustrated in Fig. 1.

### A. Key Generation Phase

*1) Quantum Key Distribution (QKD):* The BB84 protocol is employed to establish secret keys between communicating parties. Alice prepares and transmits qubits encoded in randomly chosen bases. Bob measures the received qubits in his
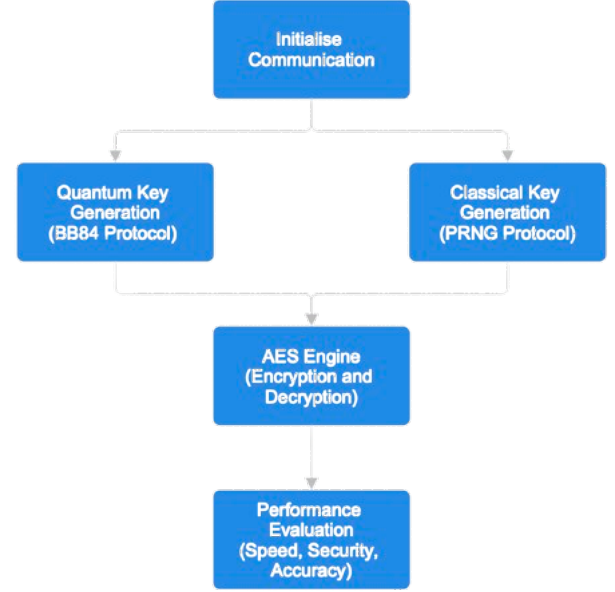


Fig. 1. High Level Flow Diagram of the Process

randomly chosen bases. After basis reconciliation and error rate estimation, a sifted key is derived. If the Quantum Bit Error Rate (QBER) is below a pre-defined threshold, the key is accepted; otherwise, the round is discarded.

*2) Classical Key Generation for Comparison:* Keys of equal length (128, 192, and 256 bits) are generated using a pseudo-random number generator (PRNG). This serves as a baseline to benchmark the efficiency and security posture of the quantum approach.

*3) Key Length Assurance:* Since QKD does not guarantee fixed key lengths per round, our framework accumulates bits from multiple rounds until the required key size (AES-128/192/256) is achieved. This step ensures practical applicability for AES integration.

### B. Encryption and Decryption Phase

*1) AES Integration:* The generated key (quantum or classical) is directly used in AES-GCM mode for encrypting and decrypting data. AES-GCM provides both confidentiality and integrity, with negligible ciphertext overhead (nonce + tag).

*2) File Size Variations:* To evaluate scalability, files of different sizes 10 KB, 100 KB, and 1 MB are encrypted using AES with 128-, 192-, and 256-bit keys.

### C. Performance Evaluation

The performance of hybrid QKD-AES encryption is compared against purely classical AES encryption. Key differences are evaluated under four major dimensions:

- Speed: Encryption throughput and latency.
- Security: Resistance to eavesdropping and quantum attacks.

- Accuracy: Reliability of decryption and key error detection.
- Scalability: Feasibility for multi-user communication scenarios.

## IV. IMPLEMENTATION

This section details the end-to-end implementation of our hybrid cryptographic system that integrates quantum key establishment with classical symmetric encryption. The work was carried out in Google Colab to ensure a portable, dependency-light environment and to make the experiments reproducible across machines without local setup friction. The implementation proceeds in five phases that mirror the project's life cycle: Classical AES/DES baseline, Quantum random key generation, BB84-based quantum key distribution, Hybrid integration that feeds quantum keys into AES/DES while resolving variable key-length availability, and a benchmarking harness that exercises different key sizes, file sizes, and key sources (quantum vs. classical) under a consistent methodology. The overall problem addressed by the system is to preserve the efficiency and maturity of AES/DES while strengthening key secrecy and distribution using quantum mechanisms, without modifying the ciphers themselves.

Environment and tooling : All code was authored and executed in Google Colab with Python 3.12. Cryptographic primitives were provided by the PyCryptodome library. Quantum circuits and simulations used modern Qiskit (Aer backends) with imports updated for the current API to avoid legacy issues. The use of Colab provides clean session isolation, straightforward package installation, and consistent CPU-only benchmarks, which is appropriate because the symmetric operations are CPU-bound and the quantum portions are simulated rather than executed on remote hardware.

Code organization : Although the notebook is linear, the implementation is logically modular. A classical crypto module encapsulates AES and DES encrypt/decrypt routines, a quantum keying module implements a quantum random number generator (QRNG) and BB84, and a hybrid harness composes these into an encryption service that draws keys from a quantum pool or a classical PRNG. A benchmarking layer generates byte payloads at target sizes, times cryptographic operations, aggregates statistics over repeated runs, and prepares tabular/plot outputs for later analysis (reported separately).

### A. Phase 1 : Classical Baseline — AES and DES

The system first establishes a reliable classical baseline to verify correctness independent of the quantum components. We implemented AES and DES using PyCryptodome in standard block-cipher modes. For integration with the benchmarking harness and to ensure authenticated encryption during experiments, AES-GCM is used as the primary mode (confidentiality and integrity with a 12-byte nonce and a 16-byte authentication tag), while CBC with PKCS7 padding was retained during early checks to mirror the initial bring-up. For DES, CBC mode with an 8-byte IV and PKCS7 padding was implemented for completeness. This baseline guarantees
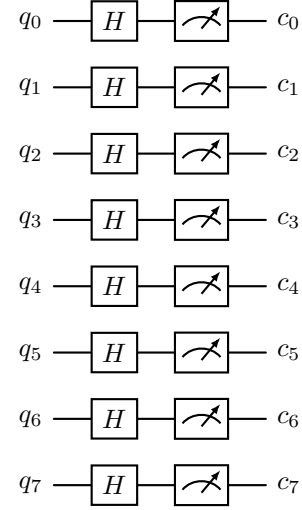


Fig. 2. QRNG circuit: each qubit is placed in superposition via $H$ and immediately measured to produce one random bit.

that encrypt/decrypt round-trips succeed deterministically with fixed keys and that mode-specific IV/nonce handling is correct. The classical baseline was explicitly built and validated as a foundation for the subsequent quantum experiments.

### B. Phase 2 : Quantum Random Number Generation

To generate high-entropy keys directly from quantum behavior, we implemented a QRNG using Qiskit. The circuit comprises N single qubit registers (where N equals the desired key length in bits). A Hadamard gate is applied to each qubit to prepare an equal superposition of $0\rangle$ and $1\rangle$, followed by measurement in the computational basis. The resulting bit string is uniformly random in the ideal simulator and is converted into bytes to serve as an AES or DES key. In practice, we created circuits for 128-bit keys and generalized the function to accept arbitrary bit lengths; the simulator backend returns one shot of the measured outcome (one N-bit sample), which is then chunked into bytes. This phase establishes a pure quantum source of keys without relying on classical PRNGs. Fig. 2. shows the circuit diagram for the QRNG circuit for a 8 qubit system.

### C. Phase 3 : BB84 Quantum Key Distribution

We implemented BB84 protocol to enable two parties to agree on a shared key with eavesdrop detection. The simulation follows the canonical roles of Alice, Bob, and an optional adversary Eve. Alice draws a random bit string and a random basis string (Z/X), encodes each bit into a one-qubit circuit using X to set $1\rangle$ and H to switch to the X basis when required, and transmits the prepared states. Bob independently selects a random basis per qubit, applies H if he measures in the X basis, and measures in the computational basis to obtain raw outcomes. Over an authenticated classical channel, Alice and Bob reveal only their bases and perform sifting: positions where their bases agree are kept, and others discarded. This

yields the sifted key. We compute the quantum bit error rate (QBER) by comparing a sample (or, in simulation, the entirety) of the sifted bits. If QBER exceeds a practical cutoff (selected below the 25% intercept-resend signature to accommodate channel noise and implementation artifacts), the round is discarded; otherwise, its bits are accepted. Our implementation also supports an intercept-resend Eve who randomly chooses measurement bases, collapses the quantum state, and forwards re-prepared states—thereby inducing errors precisely when her basis guess is incorrect. These steps realize a realistic BB84 with basis reconciliation, sifting, optional adversary modeling, and QBER-based accept/reject logic. Fig. 3. shows the circuit diagram for the BB84 quantum key distribution protocol for a 4 qubit system.

BB84 prepare–and–measure: Alice encodes bit $b_i \in \{0, 1\}$ by applying $X^{b_i}$, selects basis via $H^{\beta_i^A}$ with $\beta_i^A \in \{0(Z), 1(X)\}$, Bob measures in his basis $H^{\beta_i^B}$. After transmission, bases $\beta^A, \beta^B$ are revealed over an authenticated classical channel and only matching positions are kept (sifting).

A practical issue addressed at this stage is the variability of post-sifting key length. Because roughly half the positions are discarded due to basis mismatches and additional bits may be consumed for error-rate estimation, a single round may not produce enough bits for an AES-128/192/256 key. To ensure keys of exact length, the implementation accumulates accepted bits across multiple BB84 rounds until the target bit count is reached, and then truncates to the required size for the downstream cipher. This design makes BB84 output usable as a drop-in key source for fixed-length symmetric algorithms.

### D. Phase 4 : Integration of QKD with AES/DES Encryption

The hybrid layer connects the quantum keying pipeline to the classical ciphers. It exposes functions that request a key of specified length from the quantum source (QRNG or BB84 accumulation) or from a classical source (system PRNG), validate the key length, and pass the key to AES or DES encrypt/decrypt routines with correct IV/nonce semantics. The integration honors authenticated encryption defaults for AES-GCM and preserves deterministic verification for baseline CBC tests. By enforcing length-exact keys and encapsulating IV/nonce allocation and parsing, the hybrid layer allows the benchmarking harness to swap key sources and algorithms without touching cipher-level code paths. This phase is where the project "plugs in" quantum key establishment to unmodified AES/DES—precisely the architectural goal.

### E. Phase 5 : Benchmarking Harness

The final phase instruments the system to study performance and behavior under controlled conditions, without altering cipher internals. The harness synthesizes random byte payloads at three representative sizes (10 KB, 100 KB, 1 MB) to emulate small messages, medium objects, and larger files. For each size, the framework evaluates AES using 128-bit, 192-bit, and 256-bit keys drawn from two sources: quantum keys produced by the BB84/QRNG pipeline described earlier and classical keys produced by a cryptographic PRNG.

These runs are repeated to estimate mean and dispersion of timings. The harness separates key establishment time from cipher execution time to reflect that AES encrypt/decrypt throughput is independent of how the key was generated, whereas quantum key establishment incurs protocol-specific cost. Optional toggles activate Eve in the BB84 simulation to study the impact of QBER filtering on key availability (key-rounds discarded vs. accepted) and on the time required to accumulate enough bits for each key length. The design follows the comparative directions to evaluate the system across key sizes and file sizes and to contrast quantum-assisted operation with purely classical cryptography, while deferring all numeric outcomes to the Results and Discussion section.

Timings are recorded with high-resolution clock timers around well-defined code regions: key generation/acquisition, AES encryption, and AES decryption. Each measurement is repeated multiple times in a fresh in-memory buffer to amortize transient effects, and means and standard deviations are computed. Ciphertext length is captured to quantify per-mode overhead (e.g., GCM's nonce and authentication tag), and simple key-sanity checks (bit-balance statistics) are logged as a guardrail against accidental bias in either key source. Because all experiments are CPU-only in Colab, the measurements are consistent across runs; nevertheless, the paper reports them comparatively rather than as absolute hardware limits.

While the notebook simulates qubit preparation, measurement, and eavesdropping, the classical channel is assumed authenticated to prevent man-in-the-middle substitutions during basis reconciliation. The implementation emphasizes this separation by modeling only information that is legitimately revealed (bases, not bit values), computing QBER from ground truth in simulation, and rejecting rounds that exceed a conservative threshold. The integration layer is intentionally agnostic to the channel: once a key is accepted and length-exact, the cipher uses it identically whether sourced from QKD or PRNG. This preserves a clean separation of concerns and demonstrates how quantum key establishment can be adopted without invasive changes to existing AES-based systems.

The implementation deliberately refrains from altering AES/DES round functions or internal structures; instead, it targets the historically vulnerable key-handling surface by generating and distributing keys via quantum means. The phases and their interlocks—baseline cipher validation, quantum key material generation, BB84 with sifting and QBER checks, accumulation to exact lengths, and comparative benchmarking—correspond to the original plan and project milestones. They collectively operationalize the project's objective of fortifying classical encryption with quantum-secured keys while keeping the core of the cipher intact.

## V. RESULTS AND DISCUSSIONS

The proposed hybrid framework was implemented and evaluated through a sequence of experiments designed to highlight both the performance characteristics and the security posture of integrating quantum-generated keys into AES and DES encryption. Results are organized under three core
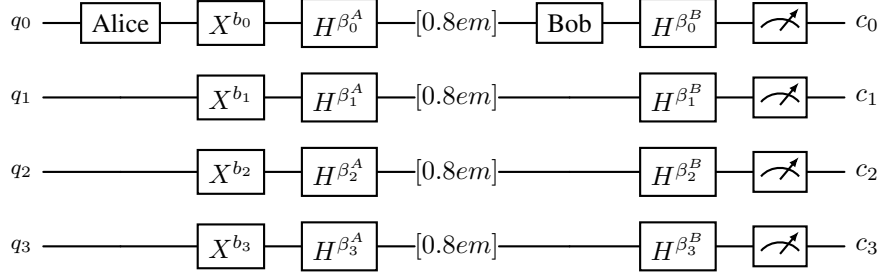
Fig. 3. Quantum circuit spanning both columns.

experimental outcomes: accuracy of eavesdropper detection in BB84, performance comparison of classical versus quantum key generation when coupled with AES encryption across different file sizes and key lengths, and resilience of QKD under eavesdropping conditions.

### A. Eavesdropper Detection Accuracy

The BB84 protocol was evaluated over 1000 independent trials, each with randomized bases for Alice and Bob and optional adversarial interference by Eve. The framework achieved a 97.3% correct identification rate of eavesdropper presence, with 444 true positives (Eve present and flagged) and 529 true negatives (no Eve, no flag). False negatives (Eve present but not detected) accounted for 27 cases, while no false positives were observed. This demonstrates that the implementation effectively leverages QBER thresholds to differentiate between normal quantum channel noise and adversarial disturbance. Fig. 4. visualizes the distribution of true and false classifications, illustrating high detection sensitivity.
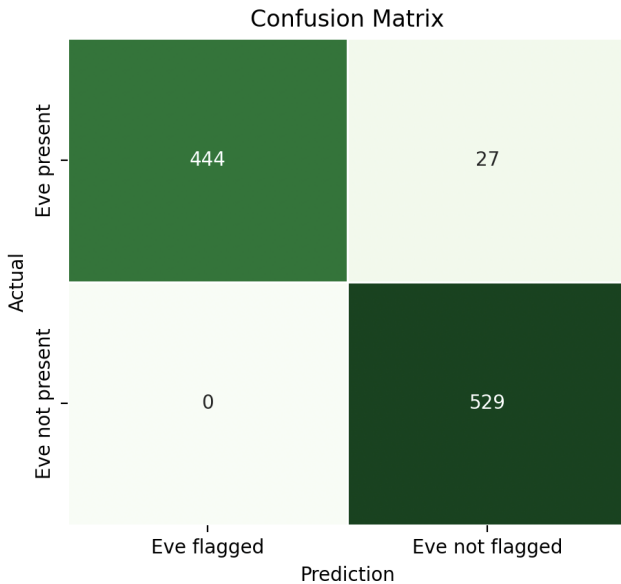


Fig. 4. Eavesdropper Detection Performance in BB84 over 1000 Trials

These results confirm the theoretical expectations of BB84, where an intercept-resend attacker induces a QBER of 25%, providing a robust statistical signature. While real-world implementations may face additional noise from imperfect hardware, our simulation verifies that eavesdropper detection is both accurate and reliable under idealized conditions.

### B. QKD Behavior in Presence of Eavesdropper

Experiments investigated QKD performance with Eve explicitly present. As expected, the rounds exhibited significantly higher QBER values (ranging from 20 to 30%), leading to systematic discarding of the sifted keys as seen in Fig. 5. In contrast, trials without Eve consistently produced QBER 0% and yielded valid keys as seen in Fig. 6. This dichotomy emphasizes the self-verifying nature of QKD, where key material is accepted only under low-error conditions and automatically rejected otherwise.
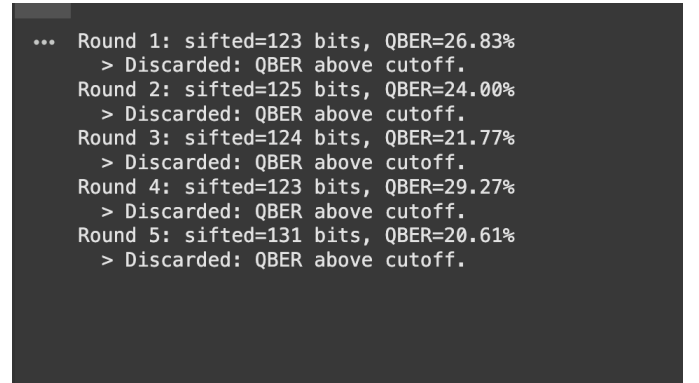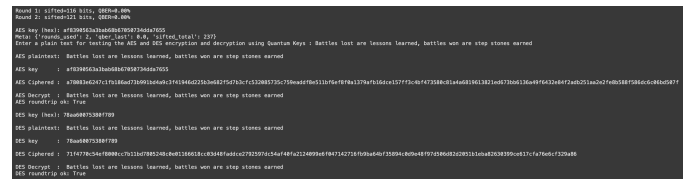


Fig. 5. QKD Behavior with Eve Present



Fig. 6. QKD Behavior with Eve Not Present

Although the presence of Eve reduced the number of usable key rounds, the ability to detect such interference is precisely the value proposition of BB84. In practice, a balance must be

struck between acceptable QBER thresholds and tolerable key refresh latency, particularly in noisy or long-distance channels.

## C. Performance Comparison : Classical vs Quantum Keys

Performance benchmarks were conducted for AES-GCM encryption across three key sizes (128, 192, and 256 bits) and three file sizes (10 KB, 100 KB, and 1 MB). Each experiment was repeated for both classical PRNG-based keys and quantum BB84-generated keys. The results visualized in Fig. 7. show that AES encryption and decryption throughput remains largely unaffected by key source, with speeds stabilizing between 250–370 MB/s for larger files. Small files (10 KB) exhibited lower throughput due to fixed overheads of nonce and authentication tag handling, but this overhead diminishes as file size increases. Across key sizes, AES-256 showed only marginally lower throughput compared to AES-128 and AES-192, consistent with the negligible performance difference reported in prior literature.



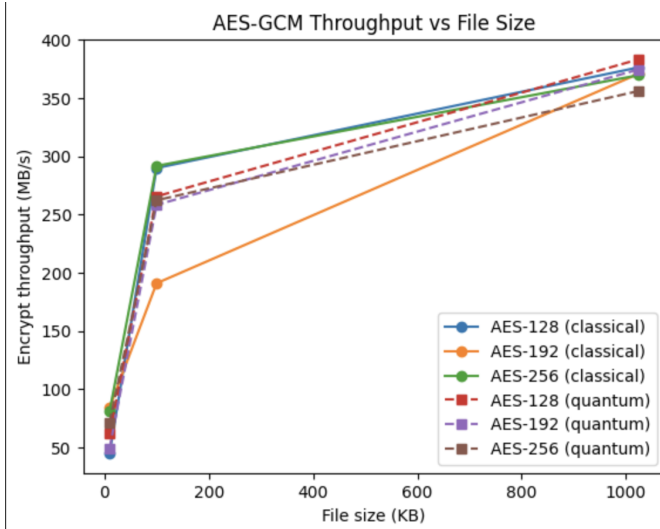Fig. 8. Time Required for Quantum Key Generation



Fig. 7. AES-GCM Encryption On Different Test Cases

The critical distinction lies in key generation time. Classical PRNG-based keys were produced in microseconds, while quantum key generation via BB84 required 38–75 seconds for 128-bit keys, 75 seconds for 192-bit keys, and 111 seconds for 256-bit keys, as shown in Fig. 8. This difference dominates the end-to-end latency when fresh keys are required per session, highlighting the trade-off between speed and unconditional security.

From a practical standpoint, this suggests that QKD-based systems should operate continuously in the background to maintain a buffer of fresh keys, which can then be consumed by AES for fast encryption. This amortizes the high cost of key generation while preserving the eavesdrop-detectable secrecy guarantees of QKD. Table 1 summarizes the comparative trade-offs across the key evaluation criteria.

These findings align with our initial expectations and the broader literature. Classical systems excel in speed, cost, and
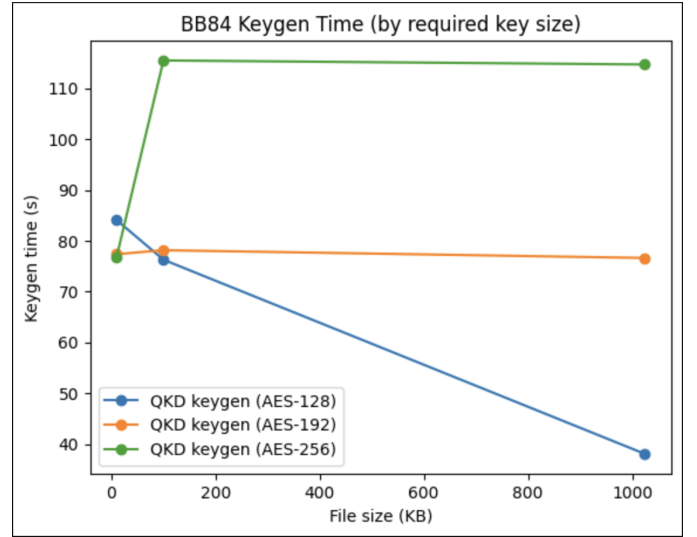
scalability, making them indispensable for everyday applications. Quantum-assisted systems, however, provide unmatched security assurances, making them suitable for high-value, long-term communications where compromise risk is unacceptable. The hybrid approach proposed here combines these strengths: quantum keys secure session establishment, while AES handles bulk data efficiently.

## VI. CONCLUSION AND FUTURE SCOPE

This work presented a hybrid cryptographic framework that integrates quantum key distribution (QKD) with classical symmetric encryption to evaluate the practicality and advantages of combining these paradigms. By using the BB84 protocol for key generation and AES/DES for encryption, we demonstrated how quantum-secured keys can be seamlessly integrated into widely deployed classical ciphers without altering their internal structure.

Experimental results confirmed that the security enhancements of QKD do not impair the performance of AES encryption itself. Throughput measurements across key sizes and file sizes showed that encryption and decryption speed remains constant regardless of whether keys are generated classically or quantum-mechanically. The principal distinction lies in the key establishment phase: while classical pseudo-random key generation occurs almost instantaneously, quantum key generation requires tens of seconds due to basis reconciliation and error checking. Despite this latency, QKD provides a critical security advantage: any adversarial interference is detectable through the quantum bit error rate, with our implementation achieving 97.3% detection accuracy in simulation.

The comparative study underscores that classical cryptography remains unmatched in speed, scalability, and cost-effectiveness, while quantum cryptography offers unparalleled, physics-backed security assurances. Rather than replacing one with the other, the results reinforce the complementarity of these systems. A hybrid approach, in which QKD is used

TABLE I
CLASSICAL VS QUANTUM METRICS COMPARISON

| Metric | Classical (PRNG + AES) | Quantum (QKD + AES) |
|---|---|---|
| Key Generation | 35.76 $\mu$s | 81.94 Seconds |
| Encryption Throughput | 350MB/s | 350MB/s |
| Security | Based on computational power. Vulnerable to Quantum hacking using Shor's and Grover's algorithm. | Based on Quantum Physics. Eavesdrop detection is available. Secure against Quantum cyber attacks. |
| Accuracy | Deterministic AES and DES Encryption | Deterministic once key is accepted. Error detection via QBER. |
| Scalability | Mature PKI and global deployment | Currently in testing phase due to hardware limitation |
| Cost | Low | High (Specialized Quantum Hardware) |

for secure session key establishment and AES is retained for bulk encryption, offers a pragmatic path toward quantum-safe communication systems.

Future work will focus on addressing the practical limitations observed in this study. These include reducing the latency of quantum key generation through protocol optimization, investigating privacy amplification and error-correction mechanisms at scale, and integrating post-quantum cryptographic primitives alongside QKD to create resilient multi-layered architectures. Real-world experimentation over fiber and free-space optical channels will also be crucial for validating the feasibility of the approach outside of simulation environments.

In conclusion, this study provides an experimental foundation for bridging the gap between classical efficiency and quantum security. By combining the maturity of AES with the future-proof guarantees of QKD, the proposed framework demonstrates a viable direction for the evolution of secure communication in the era of quantum computing.

REFERENCES

[1] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 7th ed., 2017.
[2] J. Daemen and V. Rijmen, The Design of Rijndael: AES – The Advanced Encryption Standard. Springer, 2002.
[3] Shor, P. W. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring." Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994.
[4] Grover, L. K. "A Fast Quantum Mechanical Algorithm for Database Search." Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), 1996.
[5] Bennett, C. H., and Brassard, G. "Quantum Cryptography: Public Key Distribution and Coin Tossing." Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, 1984.
[6] Gisin, N., Ribordy, G., Tittel, W., and Zbinden, H. "Quantum Cryptography." Reviews of Modern Physics, vol. 74, no. 1, 2002, pp. 145–195.
[7] Hodjat, A., and Verbauwhede, I. "A 21.54 Gbits/s Fully Pipelined AES Processor on FPGA." 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004.
[8] Schwabe, P., and Stoffelen, K. "All the AES You Need on Cortex-M3 and M4." Applied Cryptography and Network Security (ACNS), 2016.
[9] Diffie, W., and Hellman, M. "New Directions in Cryptography." IEEE Transactions on Information Theory, vol. 22, no. 6, 1976.
[10] Rivest, R. L., Shamir, A., and Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM, vol. 21, no. 2, 1978.
[11] Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., et al. "The Security of Practical Quantum Key Distribution." Reviews of Modern Physics, vol. 81, 2009, pp. 1301–1350.
[12] Lo, H.-K., Ma, X., and Chen, K. "Decoy State Quantum Key Distribution." Physical Review Letters, vol. 94, no. 23, 2005.
[13] Sasaki, M., et al. "Field Test of Quantum Key Distribution in the Tokyo QKD Network." Optics Express, vol. 19, no. 11, 2011.
[14] Boaron, A., et al. "Secure Quantum Key Distribution over 421 km of Optical Fiber." Physical Review Letters, vol. 121, no. 19, 2018.