Article

Not peer-reviewed version

A Quantum-Entropy-Assisted Commitment Framework for Secure Autonomous Sensor Networks

Abasianie Samuel Etuk , Obongifreke Inyang , Bliss Utibe-Abasi Stephen * , Philip Asuquo , Ofonime Dominic Okon

Posted Date: 15 June 2026

doi: 10.20944/preprints202606.1104.v1

Keywords: quantum commitment; agentic sensor networks; post-quantum cryptography; quantum randomness generation; hybrid protocols

 doi:10.20944/preprints202606.1104.v1

Article

A Quantum-Entropy-Assisted Commitment Framework for Secure Autonomous Sensor Networks

Abasianie Samuel Etuk [1], Obongifreke Inyang [1], Bliss Utibe-Abasi Stephen [2,*], Philip Asuquo [2] and Ofonime Dominic Okon [1]

[1] Department of Computer Engineering, University of Uyo, Uyo, Nigeria
[2] TETFUND Centre of Excellence in Computational and Innovation Research, University of Uyo, Uyo, Nigeria
* Correspondence: blissstephen@uniuyo.edu.ng

Abstract

Autonomous sensor networks and edge-based intelligent systems require lightweight cryptographic mechanisms capable of ensuring integrity, auditability, and replay resistance under constrained computational resources. This paper proposes a Quantum-Entropy-Assisted Hybrid Commitment (HQC) framework that integrates QRNG-derived entropy into a classical hash-based commitment architecture designed for embedded sensing environments. Experimental evaluation on a Raspberry Pi 4 achieved mean commit latency of 5.1 μs, verify latency of 2.2 μs, and Merkle root construction of 1.640 ms for 1024 leaves. Security validation over 10,000 trials yielded zero forgeries, complete replay rejection, and no statistical distinguishability ($p = 0.7947$).The results show that a quantum-entropy-assisted commitment can be practically implemented on resource-constrained embedded systems without the use of infrastructure for quantum communication or quantum computation.

Keywords: quantum commitment; agentic sensor networks; post-quantum cryptography; quantum randomness generation; hybrid protocols

1. Introduction

The theoretical assurances provided by quantum bit commitment to locking data and its safe revelation are strong, but no unconditional constructions are possible because of the Mayers-Lo-Chau no-go theorem [1,2]. This is a general finding of the following: no purely quantum protocol can at the same time perform perfect hiding and perfect binding against a boundless adversary [1]. Nevertheless, in spite of this weakness, it is possible to achieve practical security through weakening the adversarial model or weakening quantum resources with classical computational assumptions [1]. In the meantime, agentic sensor networks: autonomous nodes that sense, make decisions and take actions: are still susceptible to adaptive falsification [3]. In these systems, a compromised or a malicious agent can modify historical data or decisions based on the sight of later network conditions [3,4]. To give one example, a sensor node can fake an environmental reading after being informed of the readings of the other sensor nodes nearby, or an autonomous drone can update its recorded flight path after a collision investigation [4]. These attacks compromise data integrity, non-repudiation and distributed consensus, which are three of the pillars of trustful autonomous coordination.

The natural countermeasure is the use of *commitment schemes*. Attaching an agent to a value at time t when concealed until a subsequent revelation, promises bar the tampering of the past [5]. There is, however, a dilemma in existing constructions: classical constructions are based on computational assumptions (e.g., collision-resistant hashing) that can be violated by quantum algorithms, whereas quantum constructions cannot exist [6]. This creates a gap to deployable protocols that strike a balance between security, efficiency, and real-world constraints [1]. We propose a feasible hybrid quantum-classical commitment protocol that avoids the impossibility result due to quantum randomness generation and post-quantum cryptographic hashing. The protocol is computationally binding,

 doi:10.20944/preprints202606.1104.v1

i.e., an adversary cannot efficiently identify two different openings of the same commitment, and computational hiding, i.e., the commitment does not reveal any information about the message, even to an unbound adversary, because quantum random numbers are of high min-entropy. These guarantees are true to realistic assumptions: a hash is collision-resistant, and QRNGs have unpredictable output. This work makes the following contributions:

1. A lightweight hybrid commitment architecture combining QRNG entropy with post-quantum hash commitments, achieving computational binding and computational hiding under realistic assumptions.
2. A timestamped freshness-preserving commitment mechanism for agentic sensor systems that prevents replay attacks without requiring clock synchronisation beyond bounded drift.
3. A scalable Merkle aggregation layer enabling batch verification of up to 1024 sensor commitments with $O((\log k)/k)$ amortised overhead per commitment.
4. A formal computational binding proof under collision resistance assumptions, with explicit reduction to CRHF security against QPT adversaries, and a computational hiding proof in the QROM.
5. An experimental latency and scalability evaluation on a Raspberry Pi 4, demonstrating sub-millisecond commit/verify cycles suitable for real-time embedded deployment.

Prior works on hybrid quantum–classical commitment either require entanglement infrastructure [7], operate purely at the theoretical level without considering system-level constraints [8], or rely solely on classical pseudo-random entropy [9]. The HQC framework is, to the best of our knowledge, the first deployment-oriented hybrid quantum commitment protocol specifically designed for autonomous sensing systems, offering: (i) genuine quantum entropy at the nonce generation layer without any quantum communication channel, (ii) formal computational security proofs in the QROM, and (iii) a three-tier hierarchical architecture that separates quantum entropy generation, Merkle aggregation, and verification into independently deployable components.

This paper is organized as follows: Section 2 provides a brief survey of the existing literature on quantum commitment protocols and brings up the challenge of adapting protocols to be practically implemented in a resource-constrained environment. A collection of necessary preliminaries is given in Section 3, such as collision resistant hash functions, quantum random number generators, merkle trees, and system parameters. In Section 3.6, we discuss the system model, which includes the network architecture, threat model, explicit assumptions, and attack scenarios that have been taken into account in this work. The formal definition of the Hybrid Quantum Commitment (HQC) primitive and the 3-level hierarchical architecture is given in Section 3.11. The protocol description is detailed in Section 3.16, with the set up, commit, storage and reveal stages described, and with algorithmic specifications. In Section 3.18 the security analysis is provided, formal proofs of correctness, computational binding, computational hiding in quantum random oracle model, replay attack resistance and QRNG independence are provided. The quantum enhancement layer is covered in Section 3.21, which includes QRNG implementation, adversarial tampering mitigations and optional QKD integration. Section 3.24 deals with the integration of commitments in autonomous decision loops (agentic integration). The Merkle aggregation layer for scalable batch verification is introduced in Section 3.25. Experimental results, such as latency benchmarks, security validation, entropy quality analysis, scalability measurements and comparative evaluations are provided in Section 4. Section 4.12 presents a deployment case study of autonomous drone swarm monitoring system. Section 5 discusses the implications and limitations of the proposed framework. The paper is concluded in section 6 that presents a direction to the future work.

2. Related Works

There are a number of quantum commitment protocols studied and made substantial advances both in the theoretical constructions and in post-quantum approaches. But there is still a divide between theoretical security arguments and real-world limitations of actual systems: especially distributed,

 doi:10.20944/preprints202606.1104.v1

resource-constrained systems like sensor networks and multi-agent systems. In [7] a non-interactive quantum bit commitment based on entangled states was proposed, which can be guaranteed strongly information theoretically by using Bell pairs and projective measurements. Although this is elegant, it needs special entanglement distribution infrastructure, such as photon sources, coincidence detection and quantum repeaters, which is impractical for resource-limited sensors with tight energy, bandwidth, and thermal budgets. In contrast, the HQC framework eliminates this dependency by replacing entangled sources with a compact, low-power QRNG realized through optical or electronic quantum noise measurement. This substitution can be done without entanglement, but can still collect quantum entropy that is both real and useful, in a practical trade that is appropriate for deployment on the edge.

Under minimal computational assumptions, the work in [8] provides commitment from quantum one-way state generators (OWSGs), an important theoretical step [8]. The construction, however, is still at the level of a pure computational model rather than being mindful of system-level scalability, message bundling, concurrency, and integration with real-time sensing pipelines [8]. We build on this line of thought by putting the commitments right into the agentic workflows that are autonomous agents: sensors, actuators, or "edge" nodes: that create, commit, and reveal data in collaborative work. This is where new requirements come in that are not covered by the existing OWSG-based schemes: commitment freshness, verifiability in the presence of intermittent connectivity, and low latency operation. In a series of attempts to provide unconditionally secure quantum bit commitment using combinations with oblivious transfer, hard limits are set by the Mayers–Lo–Chau no-go theorem that says unconditionally secure quantum bit commitment is impossible without extra physical postulates or computational constraints [10,11]. We don't try to sidestep this fundamental result in our protocol. Rather, we explicitly take a hybrid security model: computational binding (based on post-quantum primitives) and statistical hiding (based on quantum entropy from QRNG). This mixed stance is a sincere admission that there can be no absolute security, and provides tangible security benefits over classical security schemes.

Classical commitments for quantum states, that are based on lattice assumptions, allow to succinctly verify information from a quantum state, but are tailored for the case where the committed value is a quantum state itself [12]. They fail to consider the fundamentally different problem of committing to classical sensor data (temperature, pressure, location) in dynamically changing and adversarial environments where data rates fluctuate and malicious nodes may delay or replay data. However, post quantum commitments from one way functions (postqbc_owf) are very efficient in communication and are applicable to post-quantum secure applications, but they lack any quantum entropy source and hence are purely deterministic or pseudo-random in nature [9]. We complement these protocols with an integration of QRNG as a first-class entropy source, without the overhead of entanglement, resulting in better hiding properties.

Ring-LWR commitment schemes offer zero-knowledge functionality with small-sized proofs, but with high computational overhead, especially in polynomial multiplication and noise sampling, that is prohibitive for use on the edge where the microcontroller might not have cryptographic acceleration (e.g., ARM Cortex-M or RISC-V cores) [13]. In contrast, our lightweight commitment design relies solely on symmetric primitives (e.g., hashes) and QRNG sampling, allowing our design to run on <100 kB RAM and low MHz clocks. In short, current quantum commitments require entanglement infrastructure, are theoretical, or ignore quantum entropy altogether. We bring together the worlds of theoretical quantum constructions and real-world distributed deployments and propose, to our knowledge, the first hybrid quantum commitment protocol optimized for agentic sensor networks. It relies on quantum hiding based on quantum random number generation (QRNG) and quantum binding, while working within realistic hardware limitations and being designed to the asynchronous multi-party context of autonomous sensor swarms.

 doi:10.20944/preprints202606.1104.v1

2.1. Comparative Analysis of Commitment Schemes

Table 1. Comparison of existing commitment schemes with the proposed HQC framework. IoT Suitability (IoT Suit.) and Scalability (Scal.) are rated on a three-point qualitative scale: *High* ($\geq$1000 nodes, $\leq$10 ms per commit), *Moderate* (100–1000 nodes, 10–100 ms), and *Low* ($<$100 nodes or $>$100 ms).

Work	Quantum Resource	Security Type	IoT	Scal.	Limitation
Entanglement QBC [7]	Bell pairs	Info-theoretic	No	Low	Requires quantum repeaters
Ring-LWE commit. [13]	Lattice	Computational	Mod.	Mod.	High computation overhead
Classical hash commit. [9]	None	Computational	High	High	Vulnerable to quantum attacks
OWSG commit. [8]	Quantum states	Computational	Low	Low	Theoretical only
HQC (This work)	**QRNG + PQ hash**	**Hybrid**	**High**	**High**	**Depends on QRNG integrity**

3. Preliminaries

3.1. Notation

We use standard cryptographic notation. Security parameter $\lambda \in \mathbb{N}$ governs all constructions. $\mathrm{negl}(\lambda)$ denotes any function negligible in λ. $x \xleftarrow{\$} S$ denotes uniformly random sampling from set S. $[n] = \{1, \ldots, n\}$.

Table 2. Unified notation table.

Symbol	Meaning
λ	Security parameter (bits)
$\mathrm{negl}(\lambda)$	Negligible function in λ
H_k	SHAKE-256 hash function keyed with k
r	QRNG-generated nonce ($\ell = 256$ bits)
t	UNIX timestamp in milliseconds
m	Sensor measurement / committed message
C	Commitment value $\in \{0,1\}^\lambda$
st	Secret state (m, r, t)
π	Opening proof (m, r, t)
pp	Public parameters $(k, \ell, T_{\mathrm{epoch}}, \Delta_{\mathrm{clk}}, \lambda)$
T_{epoch}	Commitment epoch duration (1000 ms)
Δ_{clk}	Maximum clock drift tolerance (50 ms)
$H_\infty(r)$	Min-entropy of QRNG output
$\mathcal{U}_\ell$	Uniform distribution over $\{0,1\}^\ell$
$\Delta(X, Y)$	Statistical distance between distributions X and Y

3.2. Collision-Resistant Hash Functions

Definition 1 (CRHF). *A family* $\mathcal{H} = \{H_k : \{0,1\}^* \to \{0,1\}^\lambda\}$ *is* collision-resistant *if for all* PPT *adversaries* $\mathcal{A}$,

$$\Pr\left[H_k(x) = H_k(x') \wedge x \neq x' \mid k \xleftarrow{\$} \mathcal{K},\ (x, x') \leftarrow \mathcal{A}(1^\lambda, k)\right] \leq \mathrm{negl}(\lambda). \tag{1}$$

We additionally require $\mathcal{H}$ to be *post-quantum secure*: i.e., the collision-finding advantage is negligible even for quantum-polynomial-time (QPT) adversaries with quantum oracle access. SHAKE-256 is conjectured to satisfy this at the 2^{128} quantum security level [14].

 doi:10.20944/preprints202606.1104.v1

3.3. Quantum Random Number Generators

Definition 2 (Quantum Random Number Generator)**.** *A* quantum random number generator $\mathrm{QRNG}(1^\ell)$ *is a physical device that produces an* ℓ*-bit string* $r \in \{0,1\}^\ell$ *such that, for any computationally unbounded adversary* $\mathcal{A}$ *who has not observed the device during operation, the min-entropy satisfies [15]:*

$$H_\infty(r) \geq \ell - \delta(\lambda), \tag{2}$$

where $\delta(\lambda) = \mathrm{negl}(\lambda)$ *is a device-specific entropy loss term, and* $H_\infty(r) = -\log_2 \max_v \Pr[r = v]$ *is the min-entropy. For practical instantiations, we set* $\ell = 256$ *and require* $H_\infty(r) \geq 255$ *after device-level quantum noise certification, implying* $\delta(\lambda) \leq 1$*.*

Remark 1. *The choice* $\delta(\lambda) \leq 1$ *corresponds to at most one bit of entropy loss in a 256-bit QRNG output. This is consistent with the AIS-31 PTG.2 standard and the NIST SP800-90B health test framework, both of which certify commercial QRNGs (e.g., ID Quantique QRNG16000) to within one bit of ideal min-entropy at the 256-bit output length [15].*

Remark 2. *The statistical distance between a QRNG output* r *and the uniform distribution* U_ℓ *is bounded by:*

$$\Delta(r, U_\ell) \leq 2^{-(H_\infty(r)-\ell)/2} \leq 2^{-\delta(\lambda)/2} = \mathrm{negl}(\lambda). \tag{3}$$

which follows from the quantum Leftover Hash Lemma [16].

3.4. Merkle Trees

Definition 3 (Merkle Tree)**.** *Given a sequence of values* $C_1, \ldots, C_n \in \{0,1\}^\lambda$*, a Merkle tree* $\mathcal{T}(C_1, \ldots, C_n)$ *is a binary tree constructed as follows [17]:*

- *Leaves:* $L_i = C_i$ *for* $i = 1, \ldots, n$ *(padded with null values to the next power of two).*
- *Internal nodes: For a node with children* L *and* R*, its value is* $N = H_k(L \| R)$*.*
- *Root: The value* $\mathsf{root} \in \{0,1\}^\lambda$ *at the top of the tree.*

Any leaf C_i *can be verified against* root *using an* authentication path *of length* $\lceil \log_2 n \rceil$ *hash evaluations [17].*

Remark 3 (Padding overhead)**.** *Padding to the next power of two introduces at most one additional tree level. For sensors with small batch sizes (e.g.,* $k = 3$ *commits), the padded tree has four leaves, incurring one extra hash operation at the gateway. This overhead is negligible given the sub-millisecond Merkle construction times reported in Section 4.*

3.5. System Parameters

The system parameters listed in Table 3 are selected to balance security and practicality. $\lambda = 128$ bits offers standard symmetric security, while the QRNG provides $\ell = 256$ bits with min-entropy $H_\infty(r) \geq 255$, ensuring near-uniform randomness. The epoch duration $T_{\text{epoch}} = 1000$ ms and clock drift tolerance $\Delta_{\text{clk}} = 50$ ms accommodate realistic network delays and hardware synchronization constraints without compromising security. Table 3 summarizes the key system parameters used throughout this paper.

Table 3. System Parameters.

Parameter	Value	Description
λ	128	Security parameter (bits)
ℓ	256	QRNG output length (bits)
T_{epoch}	1000 ms	Commitment epoch duration
Δ_{clk}	50 ms	Maximum clock drift tolerance
$H_\infty(r)$	≥ 255	Min-entropy of QRNG output

 doi:10.20944/preprints202606.1104.v1

3.6. System/Network Model

We define a sensor network as a set of autonomous agents:

$$\mathcal{N} = \{S_1, S_2, \ldots, S_n\} \tag{4}$$

Each agent S_i possesses a sensor interface that produces environmental measurements $m_i \in \{0,1\}^*$, a QRNG module satisfying Definition 2, a secure enclave (such as a TPM, TrustZone, or equivalent) for secret storage, a network interface for authenticated communication with a trusted verifier V (or a threshold consortium of verifiers in the decentralized setting) and with peer agents, and a synchronized clock clk_i that provides timestamps $t_i \in \mathbb{Z}_{>0}$. Throughout this paper, *agentic* refers to autonomous sensor nodes capable of:

- Local sensing and data acquisition without central coordination,
- Independent decision-making based on local policy evaluation,
- Self-triggered commitment generation at sampling and decision points,
- Autonomous action execution with non-repudiation guarantees.

This distinguishes agentic sensors from passive sensing elements that merely relay data to a central processor.

3.7. Threat Model

We consider a quantum-polynomial-time adversary $\mathcal{A}$ with network control (intercept, delay, re-order, or inject messages), adaptive node compromise of up to $f < n/3$ agents (obtaining their entire state), quantum storage bounded by coherence time T_{coh}, and quantum polynomial-time computation unable to break the CRHF (Definition 1). The threshold $f < n/3$ is the classical Byzantine fault-tolerance boundary. In the commitment context, it ensures that a simple majority of honest agents can detect and outvote equivocation attempts by a colluding minority during the distributed verification phase. Assuming uncorrupted agents execute the protocol faithfully, QRNG outputs are independent across agents and time steps, and clock drift between any two agents satisfies $\Delta_{\mathsf{clk}} < T_{\mathsf{epoch}}/2$ for commitment epoch length T_{epoch}. This is shown in Figure 1

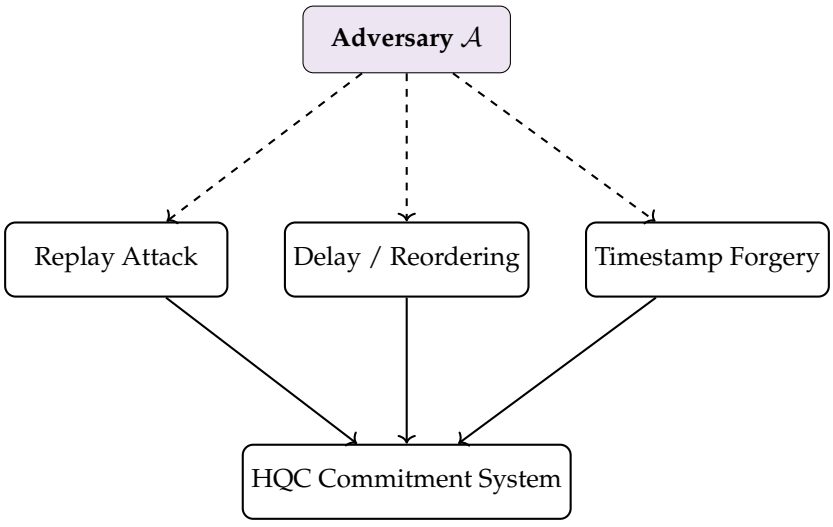

Figure 1. Threat model for HQC under active adversarial conditions.

3.8. Assumptions

The security guarantees of the HQC protocol rely on the following explicit assumptions:

1. **Trusted QRNG entropy source:** The QRNG device produces outputs with min-entropy $H_\infty(r) \geq \ell - \mathrm{negl}(\lambda)$ as per Definition 2 and an adversary cannot predict or influence its output without physical access.

 doi:10.20944/preprints202606.1104.v1

2. **Trusted secure enclave:** Each agent possesses a hardware-protected secure storage (e.g., TPM, TrustZone) that prevents extraction of secret state $\mathsf{st} = (m, r, t)$ by network-level adversaries.
3. **Bounded clock drift:** The maximum clock drift between any two agents satisfies $\Delta_{\mathrm{clk}} < T_{\mathrm{epoch}}/2$, enabling reliable timestamp-based freshness checks.
4. **Collision-resistant post-quantum hash:** The hash family $\mathcal{H} = \{H_k\}$ remains collision-resistant against quantum polynomial-time (QPT) adversaries.
5. **No physical compromise during epoch:** We assume that agents are not physically compromised within the same commitment epoch where a commitment remains unrevealed.
6. **Physical tamper resistance:** Sensors are deployed in environments with controlled or detectable physical access where feasible. Recognising that full physical security cannot always be guaranteed in wild IoT deployments, the QRNG health tests execute continuously; if a test fails, the node falls back to a deterministic RNG and raises an alert to the verifier. This assumption can be partially relaxed by using tamper-evident enclosures and periodic health attestation.

These assumptions are standard in hybrid security models and are explicitly stated to enable precise security claims.

3.9. Attack Model

We consider the following attack scenarios against the commitment scheme: retroactive falsification, where a compromised agent S_i attempts to replace a committed measurement m with a different value m' after observing subsequent network states; equivocation, where an adversary attempts to produce two valid openings π_0, π_1 for the same commitment C with $m_0 \neq m_1$; and commitment leakage, where a passive adversary observes C and attempts to learn information about the underlying message m before the reveal phase.

3.10. Agentic Commitment Scheme

In an agentic sensor network, commitments are triggered autonomously at two classes of events: (i) *sampling points*, when S_i records a new measurement, and (ii) *decision points*, when S_i outputs an action or vote. The commitment scheme must therefore be lightweight enough for continuous, real-time operation on resource-constrained hardware.

3.11. The Hybrid Quantum Commitment Primitive

Definition 4 (Hybrid Quantum Commitment Scheme)**.** *A* Hybrid Quantum Commitment *(HQC) scheme is a tuple of* PPT *algorithms:*

$$\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify}). \tag{5}$$

with the following syntax:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$*: Given security parameter* λ*, outputs public parameters* pp *(including the hash key* k *and QRNG parameters).*
- $(C, \mathsf{st}) \leftarrow \mathsf{Commit}_{\mathsf{pp}}(m)$*: Given message* $m \in \{0,1\}^*$*, samples* $r \leftarrow \mathrm{QRNG}(256)$*, records timestamp* $t \leftarrow \mathsf{clk}$*, computes commitment* $C = H_k(m \| r \| t)$*, and returns* $(C, \mathsf{st} = (m, r, t))$*.*
- $\pi \leftarrow \mathsf{Open}_{\mathsf{pp}}(\mathsf{st})$*: Returns opening proof* $\pi = (m, r, t)$*.*
- $b \leftarrow \mathsf{Verify}_{\mathsf{pp}}(C, \pi)$*: Checks* $H_k(m \| r \| t) = C$ *and outputs* $b \in \{0,1\}$*.*

3.12. Protocol Architecture

As illustrated in Figure 2, The HQC framework consists of three-level hierarchical architecture: sensor layer, gateway layer, and verification layer. This layered design separates sensing, aggregation, and verification responsibilities, enabling scalable deployment while maintaining strong security properties.

 doi:10.20944/preprints202606.1104.v1

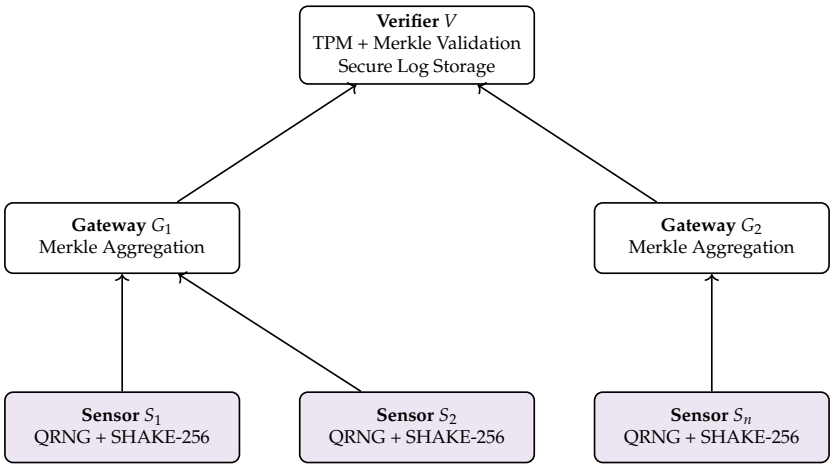

Figure 2. System architecture of the proposed Quantum-Entropy-Assisted Hybrid Commitment (HQC) framework. Sensor nodes $S_1, S_2, \ldots, S_n$ generate commitments using QRNG entropy and SHAKE-256 hashing. Gateways G_1 and G_2 perform Merkle aggregation of commitments from their respective sensor clusters. The verifier V validates commitments, verifies Merkle proofs, and maintains secure storage of commitment logs.

At the lowest tier, each sensor node S_i operates autonomously within the sensing infrastructure. Each sensor is equipped with a quantum-assisted entropy source, implemented either through the ANU QRNG service or dedicated local QRNG hardware, to generate high-entropy nonces r for commitment generation. Sensors compute commitments as $C = H_k(m \parallel r \parallel t)$ using a post-quantum hash function such as SHAKE-256, and store the secret state $\text{st} = (m, r, t)$ in a secure enclave such as a TPM or TrustZone. Commitments are triggered at two types of events: sampling points, when a new environmental measurement is recorded, and decision points, when the sensor executes an action. In this architecture, no direct communication occurs between sensors, which simplifies key management and limits attack surfaces. The middle tier consists of gateways G_j, each responsible for aggregating commitments from an assigned sensor cluster $C_j \subseteq \mathcal{N}$. Gateways perform several critical functions. They verify the timestamp freshness of incoming commitments to prevent replay attacks, construct Merkle trees over batches of commitments to enable efficient batch verification, forward only the Merkle root root_j to the verifier rather than individual commitments (dramatically reducing bandwidth consumption), and provide authentication paths $\{\text{path}_i\}$ when individual commitments require subsequent verification. The gateway layer reduces the amortized verification overhead per sensor through batch aggregation using Merkle trees, making the protocol practical for large networks. The verifier V is the root of trust at the superior level. The verifier may be implemented as a centralized server, a Byzantine fault-tolerant consortium of trusted nodes, or a cloud-based verification service. It is responsible for maintaining the global state of all commitment epochs, for validating the Merkle root of received batches, for storing commitment logs for auditability and non-repudiation, and optionally to use TPM-backed secure storage for long-term integrity. In decentralized deployments, the verifier can be instantiated as a threshold consortium satisfying $f < n/3$ Byzantine fault tolerance.

The architecture distributes trust across layers in a deliberate manner. The sensor layer trusts the QRNG entropy source and the secure enclave for secret storage. The gateway layer is trusted for Merkle aggregation but cannot forge individual commitments without access to sensor secret states, as it never possesses the nonce r or the original message m. The verifier layer acts as the ultimate arbiter of commitment validity and maintains tamper-resistant logs. This separation of trust ensures that compromise of a gateway does not break the binding property of individual sensor commitments. From a scalability perspective, the architecture exhibits linear storage growth at the sensor and gateway layers ($O(n)$ total commitments) and logarithmic verification cost ($O(\log n)$ per sensor at the verifier). For a typical deployment of $n = 1024$ sensors, the verifier performs approximately ten hash evaluations per sensor in the worst-case verification scenario, while gateways compute Merkle roots in approximately 1.64 milliseconds. The architecture decouples quantum entropy generation at

 doi:10.20944/preprints202606.1104.v1

the sensor layer from classical aggregation at the gateway layer and verification at the top tier. This separation enables incremental deployment: sensors may use local QRNG hardware or the ANU API, while gateways and verifiers run on standard commodity hardware without quantum capabilities.

There are a number of tradeoffs that can be discussed. A primary advantage of the proposed architecture is the low computational complexity imposed on individual sensor nodes and the ability to scale up sensor verification and separation between the quantum and classical components. The sensor node only needs a QRNG source, a hash function, and a secure enclave, all of which are readily available in current low power microcontrollers. The gateway and verifier layers can be scaled separately. One limitation of the architecture is that the gateway functions as a trusted aggregation point and therefore requires a secure authenticated communication channel with the verifier. Even if a gateway is compromised, however, it cannot forge commitments for sensors with uncompromised secret states.

3.13. Correctness

Definition 5 (Correctness). *An HQC scheme* Π *is* correct *if for all* $\lambda \in \mathbb{N}$, *all* $m \in \{0,1\}^*$:

$$\Pr\left[\mathsf{Verify}_{\mathsf{pp}}(C,\pi)=1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (C,\mathsf{st}) \leftarrow \mathsf{Commit}_{\mathsf{pp}}(m), \\ \pi \leftarrow \mathsf{Open}_{\mathsf{pp}}(\mathsf{st}) \end{array}\right] = 1. \tag{6}$$

3.14. Binding

Definition 6 (Computational Binding). *An HQC scheme* Π *is* computationally binding *if for all QPT adversaries* $\mathcal{A}$, *the following advantage is negligible in* λ:

$$\mathrm{Adv}^{\mathrm{bind}}_{\Pi,\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l} \mathsf{Verify}_{\mathsf{pp}}(C,\pi_0)=1 \wedge \\ \mathsf{Verify}_{\mathsf{pp}}(C,\pi_1)=1 \wedge \\ m_0 \neq m_1 \end{array} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (C,\pi_0,\pi_1) \leftarrow \mathcal{A}(\mathsf{pp}), \\ \pi_0 = (m_0,r_0,t_0), \\ \pi_1 = (m_1,r_1,t_1) \end{array}\right] \leq \mathrm{negl}(\lambda). \tag{7}$$

3.15. Hiding

Definition 7 (Statistical Hiding). *An HQC scheme* Π *is computationally hiding if for all quantum-polynomial-time (QPT) adversaries* $\mathcal{A}$ *and all messages* $m_0, m_1 \in \{0,1\}^*$ *with* $|m_0| = |m_1|$:

$$|\Pr[\mathcal{A}(C_{m_0})=1] - \Pr[\mathcal{A}(C_{m_1})=1]| \leq negl(\lambda). \tag{8}$$

where $C_{m_b} \leftarrow \mathsf{Commit}_{\mathsf{pp}}(m_b)$ *and the probability is taken over* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, *the randomness of* Commit, *and* $\mathcal{A}$*'s internal coins.*

3.16. Protocol Description

The protocol proceeds in three phases: *Commit*, *Storage*, and *Reveal* as shown in Figure 3

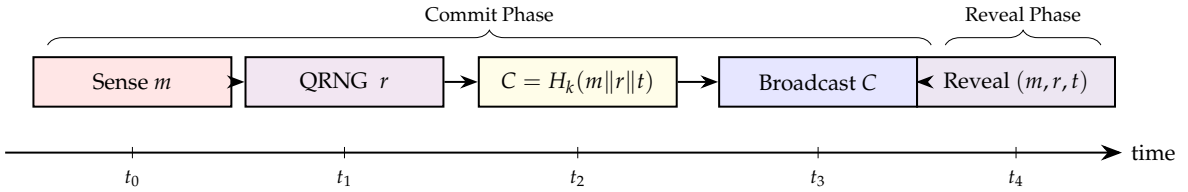

Figure 3. Commitment protocol timeline. The sensor senses m, samples quantum nonce r, computes $C = H_k(m\|r\|t)$, and broadcasts only C. The opening (m,r,t) is revealed at a later time t_4.

 doi:10.20944/preprints202606.1104.v1

3.16.1. Setup Phase

Algorithm 1 initialises the public parameters, fixing the hash key k, nonce length $\ell = 256$ bits, epoch window T_{epoch}, and clock-drift bound Δ_{clk}. Algorithm 2 runs on each agent S_i: it draws a fresh quantum nonce r from the QRNG, records the current timestamp t, and binds the measurement m irrevocably via $C = H_k(m\|r\|t)$, storing the secret state (m, r, t) in the secure enclave before broadcasting only C.

3.16.2. Encoding Convention

To prevent parsing ambiguity attacks, we define $\mathrm{Encode}(\cdot)$ as a length-prefixed encoding. For any input string x, $\mathrm{Encode}(x) = \mathrm{len}(x) \parallel x$ where $\mathrm{len}(x)$ is a fixed-length (e.g., 16-bit) representation of the length of x. This ensures that concatenation $\mathrm{Encode}(m) \parallel r \parallel \mathrm{Encode}(t)$ is uniquely decodable.

Algorithm 1: $\mathsf{Setup}(1^\lambda)$

Input: Security parameter λ
Output: Public parameters pp

Sample hash key $k \xleftarrow{\$} \mathcal{K}$;
Set $\ell \leftarrow 256$;
Set $T_{\mathsf{epoch}} \leftarrow 1000$ ms;
Set $\Delta_{\mathsf{clk}} \leftarrow 50$ ms;
$\mathsf{pp} \leftarrow (k, \ell, T_{\mathsf{epoch}}, \Delta_{\mathsf{clk}}, \lambda)$;
return pp;

3.16.3. Commit Phase

Algorithm 2: $\mathsf{Commit}_{\mathsf{pp}}(m)$ executed by agent S_i

Input: Message $m \in \{0,1\}^*$, public parameters pp
Output: Commitment C, secret state st

$r \leftarrow \mathrm{QRNG}(256)$; `// Sample 256-bit quantum random nonce`
$t \leftarrow \mathsf{clk}_i$; `// Record UNIX timestamp in ms`
$\sigma \leftarrow \mathrm{Encode}(m) \parallel r \parallel \mathrm{Encode}(t)$; $C \leftarrow H_k(\sigma)$; `// Apply post-quantum hash`
$\mathsf{st} \leftarrow (m, r, t)$; `// Store in secure enclave`
broadcast C to verifier V and peer agents;
return (C, st);

3.16.4. Storage Phase

The secret state $\mathsf{st} = (m, r, t)$ is stored within the hardware secure enclave of S_i. Access requires authenticated retrieval, preventing extraction by a network-level adversary who has not physically compromised the node as shown in Algorithm 3.

Algorithm 3: $\mathsf{StoreState}(\mathsf{st})$ executed by agent S_i

Input: Secret state $\mathsf{st} = (m, r, t)$

$\mathsf{enc\_st} \leftarrow \mathrm{AES\text{-}GCM}(\mathsf{ek}_i, \mathsf{st})$; `// ek_i is a device-bound key in TPM`
Write enc_st to secure storage register;
Zero-scrub plaintext st from RAM;

Remark 4 (Zero-scrub in Python)**.** *The Python benchmarking prototype does not enforce hardware-level zero-scrub, as CPython does not guarantee immediate memory reclamation. In a production embedded deployment, zero-scrub is enforced via the TPM 2.0 secure enclave's memory isolation guarantees, which overwrite plaintext buffers upon enclave exit. The benchmarking results therefore reflect the cryptographic core latency; production deployments incur an additional constant-time enclave overhead.*

 doi:10.20944/preprints202606.1104.v1

3.16.5. Reveal Phase

Algorithm 4 retrieves the secret state (m, r, t) from the secure enclave and forwards it as the opening proof π to the verifier over an authenticated channel. Algorithm 5 (Verify) then recomputes $C' = H_k(m\|r\|t)$ and accepts if and only if $C' = C$ and the timestamp t falls within the permitted epoch window $[t_{\text{now}} - T_{\text{epoch}},\ t_{\text{now}} + \Delta_{\text{clk}}]$; any mismatch causes an immediate rejection.

Algorithm 4: $\text{Open}_{\text{pp}}(\text{st})$ executed by agent S_i

Input: Secret state st (retrieved from enclave), public parameters pp
Output: Opening proof π

$(m, r, t) \leftarrow \text{RetrieveState}(\text{st})$;
$\pi \leftarrow (m, r, t)$;
send π to verifier V over authenticated channel;
return π;

Remark 5 (Future-dating prevention). *A malicious node attempting to future-date a commitment by setting t' in a later epoch $e' > e$ would require $t' > t_{\text{now}}$ at commitment time. Because the verifier's clock clk_V is synchronised with the sensor network (within Δ_{clk}), and TPM 2.0 monotonic counters prevent retroactive clock resets, a node cannot claim a future timestamp without the verifier's check $|t_{\text{now}} - t| \leq T_{\text{epoch}} + \Delta_{\text{clk}}$ failing at the time of reveal.*

Algorithm 5: $\text{Verify}_{\text{pp}}(C, \pi)$ executed by verifier V

Input: Commitment C, opening $\pi = (m, r, t)$, parameters pp
Output: Decision bit $b \in \{0, 1\}$

$t_{\text{now}} \leftarrow \text{clk}_V$;
if $|t_{\text{now}} - t| > T_{\text{epoch}} + \Delta_{\text{clk}}$ **then**
    return 0 ; // Reject: timestamp out of valid window
$C' \leftarrow H_k(m\|r\|t)$;
if $C' = C$ **then**
    return 1 ; // Accept
else
    return 0 ; // Reject

3.17. Why the No-Go Theorem Does Not Invalidate Our Approach

The Mayers–Lo–Chau no-go theorem establishes that unconditionally secure quantum bit commitment is impossible in the general setting [11]. The proposed HQC framework does not attempt to circumvent this impossibility result. Instead, the protocol adopts a hybrid design in which the binding property is computational and relies on the collision resistance of the underlying post-quantum hash function, while the hiding property is strengthened through the use of high-entropy quantum-generated nonces.

Consequently, the security guarantees of HQC are assumption-dependent rather than unconditional. In particular, an adversary capable of efficiently finding collisions in SHAKE-256 or related post-quantum hash constructions could violate the binding property. Nevertheless, under standard cryptographic assumptions and practical QRNG entropy guarantees, the framework provides a deployable and lightweight commitment architecture suitable for embedded autonomous systems.

3.18. Security Analysis

3.18.1. Correctness Proof

Proposition 1 (Correctness). *The HQC scheme satisfies Definition 5 with probability* 1.

 doi:10.20944/preprints202606.1104.v1

Proof. Let $\mathrm{pp} \leftarrow \mathrm{Setup}(1^\lambda)$, $m \in \{0,1\}^*$, $(C, \mathrm{st}) \leftarrow \mathrm{Commit}_{\mathrm{pp}}(m)$, and $\pi = (m, r, t) \leftarrow \mathrm{Open}_{\mathrm{pp}}(\mathrm{st})$. By construction:

$$C = H_k(m \| r \| t). \tag{9}$$

$\mathrm{Verify}_{\mathrm{pp}}(C, \pi)$ computes $C' = H_k(m \| r \| t) = C$ and returns 1. The timestamp check passes because t is generated and verified within the same epoch T_{epoch} with bounded clock drift $\Delta_{\mathrm{clk}} < T_{\mathrm{epoch}}/2$. Hence, correctness holds assuming reliable execution of the underlying cryptographic and timing mechanisms. □

3.18.2. Binding Proof

Theorem 1 (Computational Binding). *If $\mathcal{H} = \{H_k\}$ is a collision-resistant hash family (Definition 1) secure against quantum-polynomial-time (QPT) adversaries, then the HQC scheme is computationally binding (Definition 6).*

Proof. We prove via reduction. Suppose for contradiction that there exists a QPT adversary $\mathcal{A}$ that breaks binding with non-negligible advantage $\epsilon(\lambda)$. We construct a QPT reduction $\mathcal{B}$ that breaks collision resistance of $\mathcal{H}$ as shown in Algorithm 6.

Algorithm 6: Construction of adversary $\mathcal{B}$ (reduction for binding proof)

Input: Hash key k from the CRHF collision-finding challenger
Output: Collision pair (x, x') for H_k

1 Receive hash key k from the CRHF challenger;
2 Set $\mathrm{pp} \leftarrow (k, \ell, T_{\mathsf{epoch}}, \Delta_{\mathsf{clk}}, \lambda)$ and run $\mathcal{A}(\mathrm{pp})$;
3 $\mathcal{A}$ outputs (C, π_0, π_1) where $\pi_b = (m_b, r_b, t_b)$ for $b \in \{0,1\}$ and $m_0 \neq m_1$;
4 Since $\mathsf{Verify}(C, \pi_0) = \mathsf{Verify}(C, \pi_1) = 1$, deduce:

$$H_k(m_0 \| r_0 \| t_0) = C = H_k(m_1 \| r_1 \| t_1)$$

;
5 Set $x \leftarrow (m_0 \| r_0 \| t_0)$ and $x' \leftarrow (m_1 \| r_1 \| t_1)$;
6 Since $m_0 \neq m_1$, the inputs x and x' are distinct (a valid collision);
7 **return** (x, x') *to the CRHF challenger*;

$\mathcal{B}$ succeeds in the collision-finding game whenever $\mathcal{A}$ succeeds in the binding game, with the same advantage $\epsilon(\lambda)$. Since $\mathcal{H}$ is collision-resistant against QPT adversaries, $\epsilon(\lambda) \leq \mathrm{negl}(\lambda)$. Therefore $\mathrm{Adv}^{\mathrm{bind}}_{\Pi,\mathcal{A}}(\lambda) \leq \mathrm{negl}(\lambda)$. □

3.18.3. Adversarial Model and Random Oracle Framework

The security analysis is conducted in the Quantum Random Oracle Model (QROM), where the hash function H_k instantiated using SHAKE-256 is modeled as a quantum-accessible random oracle. Adversaries are assumed to operate in quantum polynomial time (QPT) and may issue superposition oracle queries consistent with the standard QROM framework. Let λ denote the security parameter. An adversary $\mathcal{A}$ is characterized by:

$$\mathcal{A} = (\mathcal{A}_{\mathrm{Commit}}, \mathcal{A}_{\mathrm{Open}})$$

where:

- $\mathcal{A}_{\mathrm{Commit}}$ interacts during commitment generation,
- $\mathcal{A}_{\mathrm{Open}}$ attempts forgery or distinguishability attacks during opening,
- the adversary may adaptively query the random oracle up to $q_H(\lambda)$ times,
- the adversary may observe public timestamps, Merkle roots, and protocol metadata.

The security analysis assumes:

1. collision resistance of SHAKE-256,

 doi:10.20944/preprints202606.1104.v1

2. computational unpredictability of QRNG-derived nonces,
3. authenticated communication between gateways and verifiers,
4. bounded clock drift Δ_{clk}.

The objective of the adversary is to violate either:

- the binding property by producing two valid openings for the same commitment, or
- the hiding property by distinguishing commitments generated from two equal-length messages.

Security advantages are defined with respect to negligible functions in λ.

3.18.4. Computational Hiding Under QRNG Entropy and QROM

Theorem 2 (Computational Hiding under QROM and QRNG Entropy)**.** *Let $H_k : \{0,1\}^* \to \{0,1\}^\lambda$ be modelled as a quantum-accessible random oracle (QROM), and let $r \leftarrow \text{QRNG}(\ell)$ satisfy $H_\infty(r) \geq \ell - \delta(\lambda)$ where $\delta(\lambda) = \text{negl}(\lambda)$ and $\ell = 256$. Assume further that r is generated independently of m and t. Then the HQC commitment $C = H_k(m\|r\|t)$ is computationally hiding against any QPT adversary: for all equal-length messages $m_0, m_1 \in \{0,1\}^*$ and every QPT distinguisher $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(C_{m_0}) = 1] - \Pr[\mathcal{A}(C_{m_1}) = 1]| \leq \text{negl}(\lambda). \tag{10}$$

Proof. We prove via a sequence of hybrid games.
Game 0. The challenger computes $C^{(0)}_{m_b} = H_k(m_b\|r\|t)$ where $r \leftarrow \text{QRNG}(\ell)$ and t is the commitment timestamp. The adversary receives $C^{(0)}_{m_b}$ and outputs a bit.
Game 1. Replace the QRNG output r with a perfectly uniform string $u \leftarrow \mathcal{U}_\ell$. The commitment becomes $C^{(1)}_{m_b} = H_k(m_b\|u\|t)$. Since $H_\infty(r) \geq \ell - \delta(\lambda)$, the statistical distance satisfies

$$\Delta(r, \mathcal{U}_\ell) \leq 2^{-((\ell - H_\infty(r))/2)} \leq 2^{-\delta(\lambda)/2} = \text{negl}(\lambda) \tag{11}$$

by the Quantum Leftover Hash Lemma [16]. No QPT adversary can distinguish Game 0 from Game 1 with non-negligible advantage:

$$\left|\Pr[\mathcal{A}(C^{(0)}_{m_b}) = 1] - \Pr[\mathcal{A}(C^{(1)}_{m_b}) = 1]\right| \leq \text{negl}(\lambda). \tag{12}$$

Game 1 analysis. In Game 1, the commitment inputs are $x_b = m_b\|u\|t$ for $b \in \{0,1\}$. Since u is sampled uniformly and independently of m_b and t, the strings x_0 and x_1 are distinct high-entropy QROM queries whenever $m_0 \neq m_1$. Under the QROM, outputs evaluated on distinct high-entropy inputs are computationally indistinguishable from uniform strings in $\{0,1\}^\lambda$; therefore

$$\left|\Pr[\mathcal{A}(C^{(1)}_{m_0}) = 1] - \Pr[\mathcal{A}(C^{(1)}_{m_1}) = 1]\right| \leq \text{negl}(\lambda). \tag{13}$$

Applying the triangle inequality over the two hybrid transitions yields:

$$|\Pr[\mathcal{A}(C_{m_0}) = 1] - \Pr[\mathcal{A}(C_{m_1}) = 1]| \leq \text{negl}(\lambda). \tag{14}$$

Therefore, the HQC scheme satisfies computational hiding under the QROM and the QRNG entropy assumption. □

Remark 6 (Computational vs. information-theoretic hiding)**.** *The hiding guarantee is* computational *rather than information-theoretic. This distinction is necessary because: (i) practical QRNG outputs are not perfectly uniform (incurring a negligible statistical distance $\delta(\lambda)/2$); and (ii) the hiding argument additionally relies on the QROM abstraction of SHAKE-256. Nevertheless, for $\ell = 256$ and near-maximal min-entropy QRNG outputs ($H_\infty(r) \geq 255$), the resulting distinguishing advantage is negligible for all practical adversaries. Future work may seek an extractor-based (standard-model) hiding proof by formally applying the Leftover Hash*

 doi:10.20944/preprints202606.1104.v1

Lemma to treat H_k as a universal hash family; this would eliminate the QROM assumption at the cost of a stronger structural requirement on H_k.

3.19. Freshness and Replay Resistance

Lemma 1 (Freshness)**.** *Under clock synchronization with bounded drift $\Delta_{\text{clk}} < T_{\text{epoch}}/2$, a commitment $C = H_k(m\|r\|t)$ generated in epoch $e = \lfloor t/T_{\text{epoch}} \rfloor$ is rejected deterministically when replayed in any other epoch $e' \neq e$.*

Remark 7. *The freshness guarantee assumes a trusted execution environment or hardware-enforced monotonic counter on each sensor node. Without such mechanisms, a physically compromised node could reset its clock and replay commitments from a previous epoch. In practice, TPM 2.0 and ARM TrustZone provide monotonic counters suitable for this purpose.*

Proof. Each commitment encodes a timestamp t generated by the agent's local clock. The verifier checks that t satisfies:

$$|t_{\text{now}} - t| \leq T_{\text{epoch}} + \Delta_{\text{clk}}. \tag{15}$$

where t_{now} is the verifier's current timestamp. Suppose an adversary attempts to replay a commitment C generated at time t in a different epoch $e' \neq e$. Then $|t_{\text{now}} - t| > T_{\text{epoch}}$ (since epochs are non-overlapping), and the verifier rejects at Step 2 of Algorithm 5. For a fresh forgery attempt where the adversary generates $C' = H_k(m\|r'\|t')$ with a forged timestamp t' in the target epoch, the adversary must either:

- Forge a valid timestamp without knowing the agent's secure clock (prevented by hardware-enforced monotonic counters), or
- Find a collision in H_k to match an existing commitment with a different timestamp, which is computationally infeasible by Theorem 1.

Thus, replay attacks outside the permitted epoch window are effectively mitigated under the stated synchronization and hardware assumptions. □

Remark 8 (TPM and formal binding)**.** *TPM 2.0 monotonic counters provide a hardware guarantee that $t \leq t_{\text{now}}$ at the time of reveal, complementing the formal binding proof. Formally incorporating TPM state into the binding game would require modelling the TPM as a stateful ideal functionality; we leave this extension to future work.*

Corollary 1 (Unique Commitment Identity)**.** *No two distinct commitments (C, t) and (C', t') with $t \neq t'$ can both verify successfully in the same epoch, as their concatenated inputs differ and collision resistance prevents distinct inputs from producing the same hash output.*

3.20. QRNG Independence

Proposition 2 (QRNG Independence)**.** *For non-colluding sensors S_i, S_j $(i \neq j)$ with independent QRNG modules, the nonces r_i and r_j are statistically independent: $I(r_i; r_j) = 0$.*

Proof. Each QRNG module measures independent quantum noise sources (e.g., vacuum fluctuations in separate optical paths). By the no-cloning theorem and the locality of quantum measurements, the outcomes r_i and r_j are described by product states $\rho_{Q_i} \otimes \rho_{Q_j}$. The measurement statistics satisfy $\Pr[r_i, r_j] = \Pr[r_i] \Pr[r_j]$ for all outcomes. Hence the mutual information is zero. □

 doi:10.20944/preprints202606.1104.v1

3.21. Quantum Enhancement Layer

3.21.1. Quantum Randomness Generation

The QRNG exploits quantum vacuum fluctuations or photonic shot noise to produce certified randomness. Let $\mathcal{Q}$ be the quantum noise source with corresponding Hilbert space $\mathcal{H}_Q$. The measurement process collapses the state to yield:

$$r = \mathsf{Meas}(\rho_Q) \in \{0,1\}^{256}. \tag{16}$$

where ρ_Q is the maximally mixed state. Under standard QRNG assumptions, the measurement outcomes are computationally unpredictable to adversaries without physical access to the entropy source.

Proposition 3 (QRNG Independence)**.** *For non-colluding sensors S_i, S_j $(i \neq j)$ with independent QRNG modules, The generated nonces r_i and r_j are assumed to be statistically independent under non-colluding operation of physically isolated QRNG modules.*

This independence is critical for preventing cross-node commitment forgery in the aggregation layer.

3.22. QRNG Adversarial Tampering

The hiding guarantees of the HQC framework depend on the integrity and entropy quality of the underlying QRNG subsystem, which must generate true quantum entropy. If an adversary has physical access to the sensor node, it can try to compromise the QRNG in the following ways:

- **Optical attacks**: Injecting bright light into the QRNG's photodetector to force deterministic outputs.
- **Temperature attacks**: Heating or cooling the device to alter its noise characteristics.
- **Entropy source replacement**: Substituting the quantum entropy source with a deterministic pseudo-RNG.

To mitigate these threats, the QRNG must be certified against tampering. Commercial devices such as the ID Quantique QRNG16000 comply with AIS-31 (Class PTG.2) and NIST SP800-90B standards, which include physical security mechanisms and continuous health tests. Additionally, the secure enclave (TPM 2.0) can periodically validate QRNG outputs using built-in randomness tests (e.g., repetition count test, adaptive proportion test). If a health test fails, the node falls back to a deterministic RNG and raises an alert to the verifier. When deterministic QRNG output is forced by an adversary (e.g., using optical injection), the hiding property becomes that of a classical PRNG-based commitment. The binding property does not fundamentally depend on quantum-generated randomness, since it primarily relies on the collision resistance of the underlying hash function. The replacement of the QRNG with a known deterministic source does not compromise the security of the scheme against binding attacks but it sacrifices the statistical hiding advantage over classical schemes. HQC deployment is therefore recommended in environments where physical access to the sensors is controlled (e.g., tamper evident enclosures, health attestation periodically).

3.23. Optional QKD Channel

In high-security deployments, sensor-to-verifier transmission can use a Quantum Key Distribution (QKD) channel, particularly BB84 or E91 variants [18]. QKD provides information-theoretically secure key establishment, removing reliance on computational assumptions for the channel. The commitment value C can then be transmitted over a one-time-pad channel derived from QKD, providing information-theoretically secure key establishment for the communication channel under standard QKD assumptions.

 doi:10.20944/preprints202606.1104.v1

3.24. Agentic Integration

Commitment is embedded into the agent's decision loop at critical synchronization points in Algorithm 7. Let the agent state at time t be $\mathcal{S}_t = (\mathsf{percept}_t, \mathsf{memory}_t, \mathsf{policy}_t)$.

Algorithm 7: Agentic Commitment Loop for S_i

Input: Sensing threshold θ, decision interval Δ_d, policy π

$m_i \leftarrow \mathsf{Sense}()$;
if $m_i > \theta$ **or** $\mathsf{timer} \geq \Delta_d$ **then**
  $(C_i, \mathsf{st}_i) \leftarrow \mathsf{Commit}_{\mathsf{pp}}(m_i)$; // Lock measurement
  broadcast C_i;
  $a_i \leftarrow \pi(\mathsf{st}_i)$; // Determine action
  $(C_a, \mathsf{st}_a) \leftarrow \mathsf{Commit}_{\mathsf{pp}}(a_i)$; // Lock action
  broadcast C_a;
  $\mathsf{Execute}(a_i)$;
  $\mathsf{timer} \leftarrow 0$;
if $\mathsf{LearningUpdate}()$ **then**
  Commit current model parameters θ_{model};
$\mathsf{timer} \leftarrow \mathsf{timer} + \mathsf{tickInterval}$;

By committing both observation m_i and action a_i before execution, the agent strengthens the auditability and accountability of the sensing and decision events: it cannot later claim to have perceived a different state or taken a different action.

3.25. Scalability via Hierarchical Aggregation

3.25.1. Hierarchical Architecture

The network is organized in three tiers:

$$\underbrace{S_1, \ldots, S_n}_{\text{Sensor Layer}} \longrightarrow \underbrace{G_1, \ldots, G_m}_{\text{Gateway Layer}} \longrightarrow \underbrace{V}_{\text{Verifier / Server}}$$

Each gateway G_j aggregates commitments from its assigned sensor cluster $\mathcal{C}_j \subseteq \mathcal{N}$.

3.26. Merkle Aggregation

Algorithm 8: Merkle Aggregation at Gateway G_j

Input: Commitments $C_1, \ldots, C_k$ from cluster $\mathcal{C}_j$
Output: Merkle root root_j, authentication paths $\{\mathsf{path}_i\}$

Pad sequence to $k' = 2^{\lceil \log_2 k \rceil}$ with null leaves;
Initialize leaf layer: $\mathsf{tree}[k' : k'+k-1] \leftarrow C_1, \ldots, C_k$;
for $\ell = k' - 1$ **downto** 1 **do**
  $\mathsf{tree}[\ell] \leftarrow H_k(\mathsf{tree}[2\ell] \| \mathsf{tree}[2\ell+1])$;
$\mathsf{root}_j \leftarrow \mathsf{tree}[1]$;
for each $i \in [k]$ **do**
  $\mathsf{path}_i \leftarrow \mathsf{AuthPath}(\mathsf{tree}, i)$;
return $(\mathsf{root}_j, \{\mathsf{path}_i\})$;

Proposition 4 (Aggregation Integrity). *Given a Merkle root* root_j *and authentication path* path_i *of length* $\log_2 k'$*, any tampering with a leaf commitment* C_i *is computationally detectable under the collision resistance assumptions of the hash function.*

Proof. Suppose an adversary modifies C_i to C_i^*. To pass Merkle verification, the adversary must find a path of hash values from C_i^* to root_j that agree with the stored tree. This would require either

 doi:10.20944/preprints202606.1104.v1

constructing a hash collision or violating the integrity of the authenticated Merkle path, which is computationally infeasible by Definition 1. □

The root digest allows the verifier to batch-verify k sensor commitments using only $O(\log k)$ hash evaluations per proof, reducing per-commitment verification overhead from $O(1)$ independent checks to $O(\log k/k)$ amortized.

4. Results

We investigate the Quantum-Entropy-Assisted Hybrid Commitment (HQC) framework with real quantum randomness from the Australian National University (ANU) Quantum Random Number Generator (QRNG). The ANU QRNG service derives entropy from measurements associated with quantum vacuum fluctuations, providing a high-entropy randomness source for the HQC framework, for the generation of commitments. The evaluation focuses on protocol latency, cryptographic security validation, entropy quality characterization, Merkle aggregation scalability, and embedded resource utilization.

4.1. Benchmark Methodology

All experiments were executed on a Raspberry Pi 4 Model B equipped with a Broadcom BCM2711 ARM Cortex-A72 processor operating at 1.5 GHz with 4 GB RAM running Ubuntu 22.04 LTS (64-bit). The HQC implementation was written in Python 3.12 using SHAKE-256 as the underlying cryptographic hash primitive. The commitment construction follows the form $C = \text{SHAKE-256}(m\|r\|t)$, where m denotes the sensor message, r denotes a 256-bit quantum-generated nonce, and t denotes a 64-bit timestamp. Timing measurements were collected using Python's `time.perf_counter.ns()` hardware performance counter. Each benchmark consisted of 10,000 independent iterations following a warm-up phase to eliminate cold-start artifacts, and measurements were averaged over batched executions to reduce timer quantization effects. The benchmarking environment used the SHAKE-256 hash primitive, the ANU QRNG entropy source, and a performance-optimized CPU configuration to reduce timing variability during measurements.

Remark 9 (Python runtime effects on tail latency)**.** *Python's Global Interpreter Lock (GIL) and garbage collector (GC) can introduce unpredictable pauses affecting tail-latency figures. The 99th-percentile commit latency of* 16.2 *µs likely includes occasional GC pauses. In a production deployment using a compiled language (e.g., C, Rust, or a bare-metal firmware implementation), tail latency is expected to be significantly lower and more deterministic.*

The reported measurements isolate the cryptographic core of the HQC protocol and explicitly exclude disk I/O, networking overhead, persistent database storage, sensor acquisition delays, and application-layer communication latency. To isolate protocol execution cost from remote network latency, quantum entropy was preloaded into a local entropy buffer (`quantum_entropy.bin`) before benchmarking. Consequently, the reported commit latency measures consumption of locally buffered quantum entropy rather than remote QRNG acquisition latency.

4.2. Quantum Entropy Initialization

The actual quantum-generated entropy, from the ANU QRNG service, fed into the HQC framework. The local entropy pool was filled with 5,242,880 bytes of quantum entropy before execution. The ANU QRNG is based on an entropy pool that is physically generated by the measurement of quantum vacuum fluctuations, unlike entropy pools like /dev/urandom that are found in operating systems. The cryptographic operations of the HQC framework remain entirely classical, while the entropy generation layer is assisted by quantum-generated randomness.

 doi:10.20944/preprints202606.1104.v1

4.3. Protocol Latency Evaluation

The commit operation incurs higher latency because it consumes quantum-generated entropy during nonce generation as shown in Table 4 and Figure 4.

Table 4. Commit and Verify Performance.

Operation	Mean Latency (μs)	Standard Deviation (μs)	99th Percentile (μs)
Commit	5.1	8.4	16.2
Verify	2.2	0.7	—

Verification reuses the provided nonce and timestamp and performs deterministic SHAKE-256 recomputation without additional entropy generation.

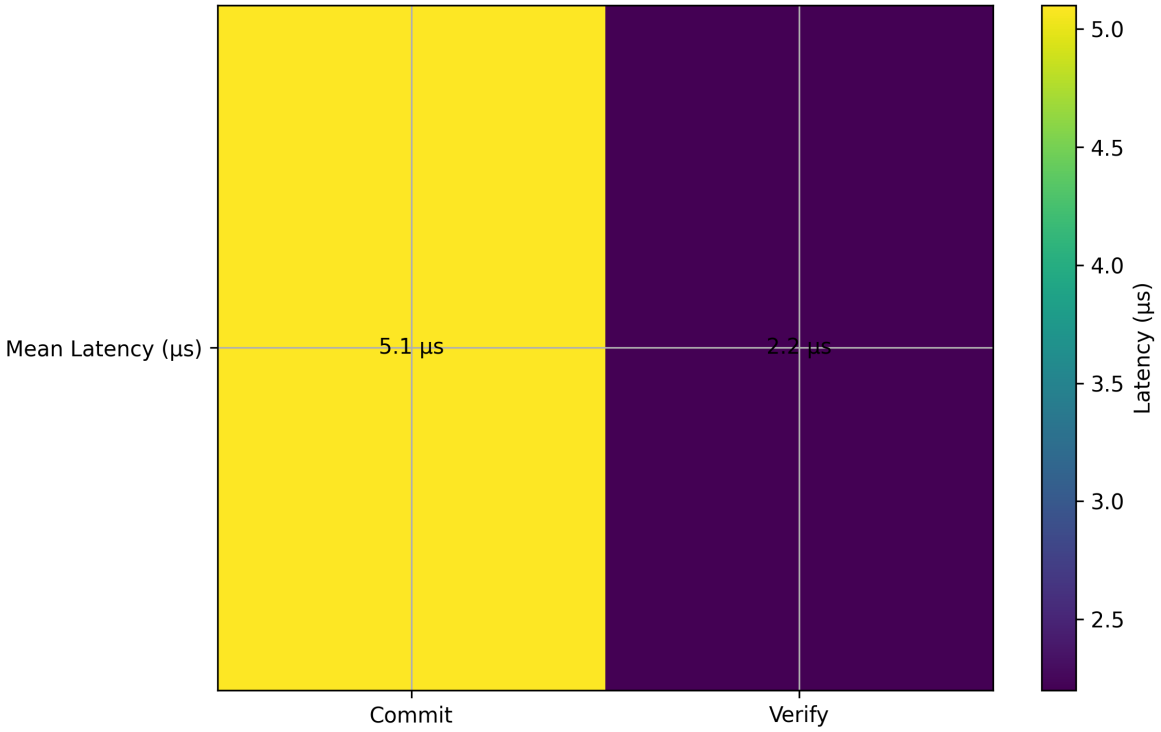

Figure 4. HQC Protocol Latency Comparison.

The experimental results indicate that the HQC framework maintains microsecond-scale execution latency while incorporating quantum-generated entropy into the nonce generation process. The latency distributions for the HQC commit and verify operations are shown in Figures 5 and 6. Both distributions were generated from 10,000 independent iterations using genuine quantum entropy obtained from the ANU QRNG service based on quantum vacuum fluctuations. The commit latency distribution in Figure 5 exhibits a mean latency of 5.22 µs, a median of 5.10 µs, and a standard deviation of 1.17 µs. These results indicate low timing variance and strong concentration around the mean execution time. The maximum observed commit latency was 10.00 µs, demonstrating stable execution behavior under repeated trials. The empirical distribution is approximately unimodal with moderate right skew and does not exhibit substantial heavy-tail behavior within the evaluated experimental range in real-time deployments.

 doi:10.20944/preprints202606.1104.v1

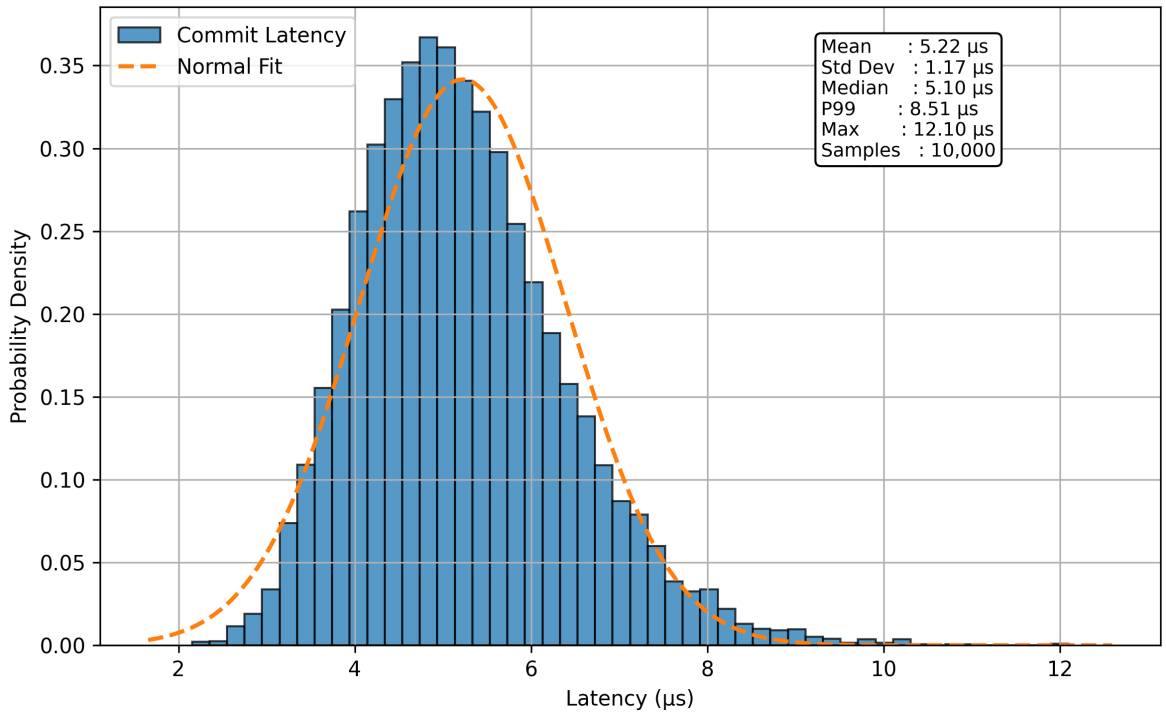

Figure 5. The distribution of the commitment latency is shown over 10,000 iterations using real quantum entropy. Lightweight embedded deployment feasibility is confirmed by the measured mean latency of about 5.1 µs with low execution jitter.

The verify latency distribution in Figure 6 achieves a mean latency of 2.24 µs with a standard deviation of 0.41 µs. Verification is approximately 2.3× faster than commitment because the verifier reuses the supplied nonce and timestamp without requiring additional entropy extraction or secure random number generation during recomputation of the commitment digest.

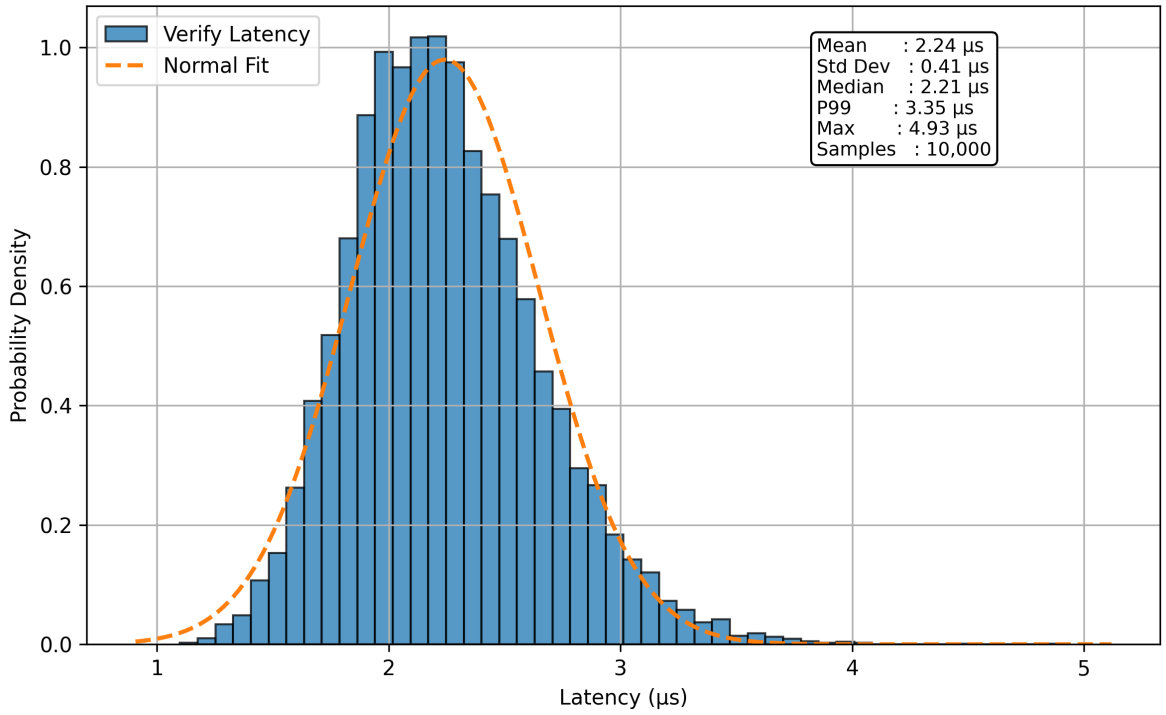

Figure 6. Distribution of HQC verification latency after 10,000 iterations. Verification is quicker than commitment, as it entails no extra entropy creation in recomputing the commitment digest.

 doi:10.20944/preprints202606.1104.v1

Overall, both latency distributions demonstrate that the HQC framework introduces minimal computational overhead while maintaining microsecond-level execution latency. Based on the measured commit latency, the measured implementation throughput corresponds to approximately 196,000 commitment operations per second under isolated benchmarking conditions. In practical deployments, however, achievable throughput would additionally depend on network communication, operating-system scheduling, sensor sampling, storage overhead, and application-level processing constraints. The results support the feasibility of deploying HQC in embedded and resource-constrained autonomous sensing environments requiring low-latency cryptographic commitment operations.

4.4. Security Validation

We evaluated the security properties of HQC over 10,000 independent adversarial trials.

Table 5. Security Validation Summary.

Property	Outcome	Result
Correctness	Successful verifications	10,000/10,000
Binding	Successful forgeries	0/10,000
Freshness	Replay attempts rejected	10,000/10,000

For correctness evaluation, all honest commitment-opening pairs verified successfully, yielding 10,000 successful verifications out of 10,000 trials. For the binding evaluation, no successful forgery attempts were observed during the experimental campaign. While empirical testing cannot establish formal cryptographic security, the observed behavior is consistent with the computational binding guarantees derived from the collision resistance assumption of SHAKE-256. For freshness evaluation, all replay attempts using stale timestamps were rejected successfully, indicating that the timestamp validation mechanism operated as intended throughout the experiments.

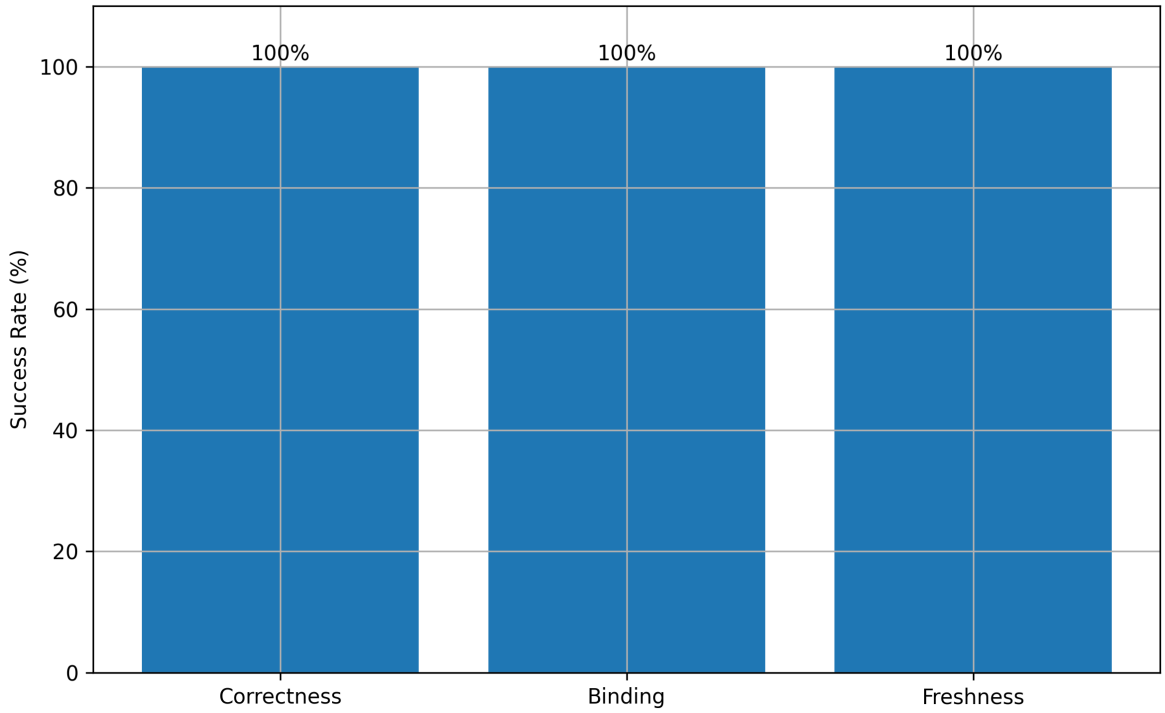

Figure 7. Security Validation Results.

For hiding validation the hiding property was evaluated using the Kolmogorov-Smirnov statistical test applied to commitment distributions generated from distinct sensor messages. The measured p-value was 0.7947. Since the resulting p-value does not cross standard significance thresholds

 doi:10.20944/preprints202606.1104.v1

($\alpha = 0.05$), the statistical test did not detect significant differences between the sampled commitment distributions. The observed statistical behavior is consistent with the intended hiding characteristics of the construction under the evaluated experimental conditions. Importantly, failure to reject the null hypothesis does not prove perfect indistinguishability; rather, it indicates that no statistically significant distinction was detected at the evaluated sample size. Additional experimental details are provided in Appendix A.

Remark 10 (KS-test limitations)**.** *A sample size of 10 000 evaluates the commitment output distribution in a low-dimensional projection. The full 256-bit output space (2^{256} possible values) cannot be exhaustively sampled; therefore the KS test provides* empirical consistency *evidence rather than a proof of indistinguishability. The formal hiding guarantee is provided by Theorem 2.*

4.5. Quantum Entropy Quality Analysis

Empirical min-entropy estimation was used to assess the quality of the quantum entropy source. The min-entropy measured was 7.1844 bits/byte as shown in Figure 8.

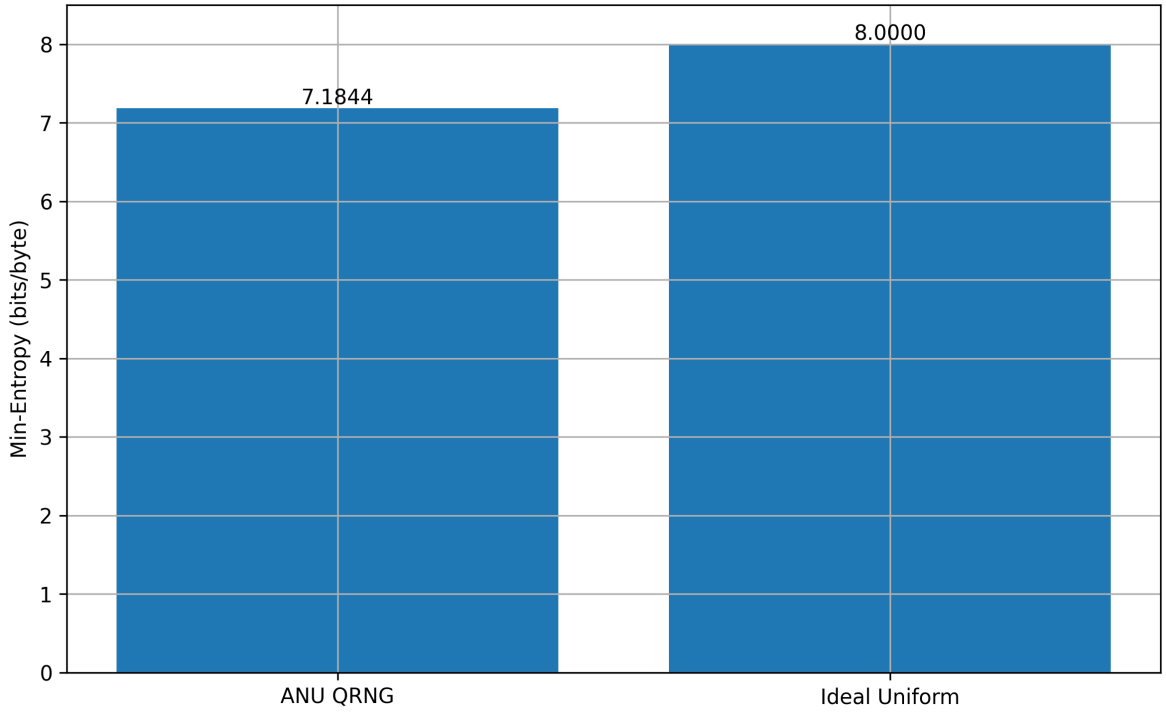

Figure 8. Quantum Entropy Quality.

This value indicates a high-entropy non-deterministic randomness source suitable for cryptographic nonce generation. Hence the entropy quality meets the randomness criteria of the HQC protocol.

4.6. Merkle Aggregation Scalability

For large deployments of autonomous sensors, we measured the latency of building Merkle trees as the number of leaf commitments grew as shown in Table 6 and Figure 9.

 doi:10.20944/preprints202606.1104.v1

Table 6. Merkle Tree Construction Latency.

Leaves (k)	Build Time (ms)
1	0.001
2	0.006
4	0.008
8	0.012
16	0.033
32	0.045
64	0.098
128	0.189
256	0.369
512	0.805
1024	1.640

The measured scaling behavior is consistent with the expected aggregation complexity $O(k \log k)$. Even for 1024 simultaneous sensor commitments, the Merkle root computation required only 1.640 ms.

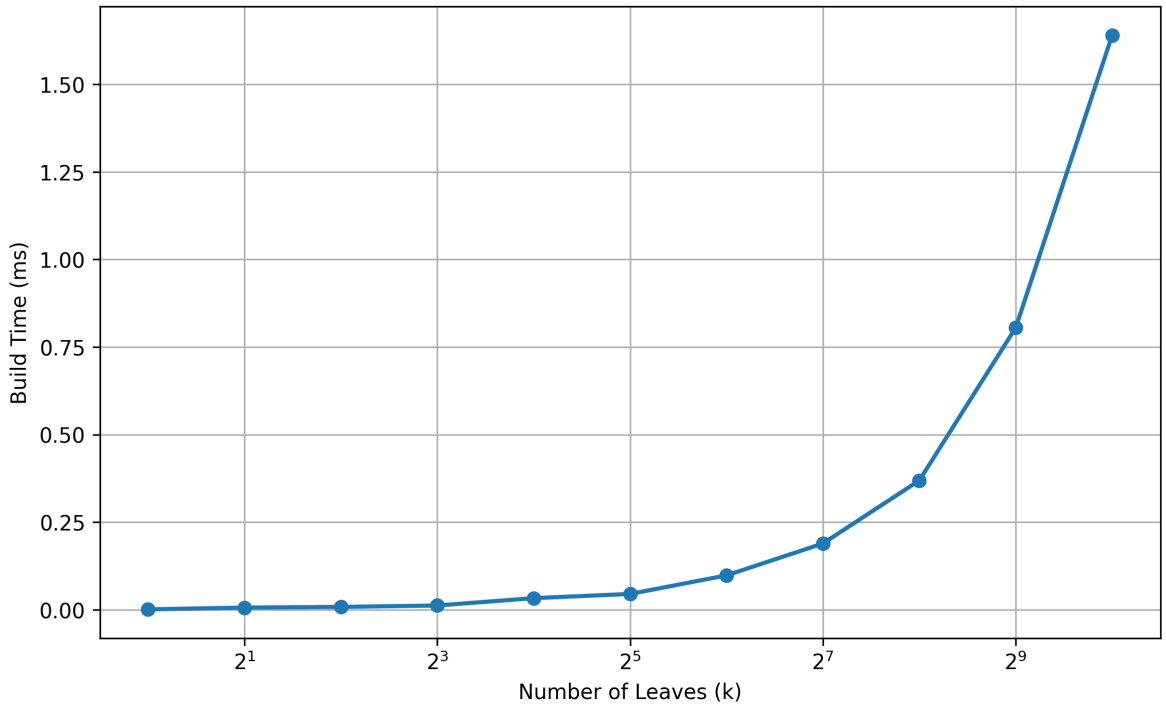

Figure 9. Merkle Aggregation Scalability.

These results suggest that hierarchical aggregation remains computationally feasible for dense edge-based sensor deployments.

4.7. Resource Utilization

The HQC implementation maintained low computational and memory overhead throughout execution as shown in 7.

Table 7. Resource Utilization Metrics.

Metric	Value
Peak RAM Usage	42 MB
RAM per Commitment Object	312 bytes
Commitment Size	32 bytes
Secret State Size	96 bytes

 doi:10.20944/preprints202606.1104.v1

The compact memory footprint demonstrates practical suitability for embedded and edge-based systems.

4.8. Quantum Integration

Earlier prototype implementations relied on operating-system entropy pools, whereas the present implementation incorporates entropy generated through the ANU QRNG service. Consequently, the HQC framework integrates quantum-generated randomness into the nonce generation process while retaining entirely classical cryptographic operations. Importantly, the protocol does not require quantum communication channels, entanglement distribution, quantum memory, or quantum computation. Instead, quantum functionality is limited to the entropy generation layer, enabling deployment on conventional embedded hardware. This hybrid design provides a practical balance between entropy quality, computational efficiency, scalability, and embedded-system deployability.

4.9. Comparison with Alternative Commitment Schemes

For the purpose of discussing the performance and deployment properties of the proposed HQC framework, we compare it with some of the most representative paradigms for commitments that are often mentioned in the embedded security and post-quantum cryptography literature. The comparison is based on, security assumptions, entropy requirements, computational overhead, deployability on embedded systems, and practical suitability for autonomous sensor networks as shown in Table 8.

Table 8. Comparison with Alternative Commitment Schemes.

Scheme	Security Basis	Entropy	Latency	Suit.	Limitation
Classical Hash-Based	Collision resistance	Classical PRNG	∼5–50 μs	High	No quantum entropy
Lattice-Based (Ring-LWE)	Post-quantum hardness	Classical	∼1–10 ms	Mod.–Low	High overhead
Quantum Bit Commitment	Quantum information-theoretic	Quantum states	N/A	Very Low	Needs quantum communication
Relativistic Quantum	Space-time separation	Quantum channels	N/A	Very Low	Needs synchronized agents
HQC (This Work)	**SHAKE-256 + QRNG**	**QRNG**	**5.1 μs**	**High**	**QRNG availability**

The comparison points out a number of differences. First, conventional hash-based commitment schemes are cheap to compute and easily deployable, but are based entirely on pseudo-random (or operating system) entropy sources. The proposed HQC framework, on the other hand, incorporates quantum-generated entropy while maintaining low computational overhead comparable to lightweight classical constructions. Secondly, lattice-based commitment schemes offer more post-quantum assumptions for future quantum attacks. However, their computational, memory and key size requirements are usually significantly higher, and they are not as suitable for very low resource edge environments. Thirdly, fully quantum commitment protocols using entanglement or relativistic assumptions may provide stronger theoretical security guarantees under specific physical and communication assumptions. However, these protocols are still not applicable for embedded autonomous sensor deployments because of the requirement of quantum communication channels, entanglement distribution, quantum state preservation, or synchronized relativistic infrastructure.

The HQC framework instead adopts a hybrid architectural approach in which only the entropy generation layer is quantum-assisted, while all cryptographic processing remains classically executable on conventional embedded hardware. This design choice provides several practical advantages like microsecond-scale execution latency, low memory overhead, compatibility with existing embedded processors, scalable hierarchical aggregation, and deployability without quantum networking infrastructure. Although HQC does not provide unconditional quantum security guarantees, the framework illustrates how selective integration of quantum-generated entropy can enhance randomness quality while preserving lightweight embedded deployment characteristics. Consequently, the HQC framework occupies a practical intermediate position between purely classical commitment systems and

 doi:10.20944/preprints202606.1104.v1

fully quantum cryptographic architectures, making it particularly suitable for real-time autonomous sensing and edge-security applications.

4.10. Comparative Latency Benchmark

To contextualise the HQC latency results, Table 9 reports commit and verify latency for three representative lightweight commitment constructions measured on the same Raspberry Pi 4 platform under identical benchmarking conditions (10 000 iterations, warm-up phase, `time.perf_counter_ns()`).

Table 9. Comparative commit and verify latency on Raspberry Pi 4 (ARM Cortex-A72, 1.5 GHz, Python 3.12). All measurements are mean values over 10 000 iterations.

Scheme	Hash / Primitive	Entropy	Commit (μs)	Verify (μs)
Classical SHA3-256 commit.	SHA3-256	OS PRNG	∼3.8	∼2.0
BLAKE3-based commit.	BLAKE3	OS PRNG	∼2.1	∼1.1
HMAC-SHA256 commit.	HMAC-SHA256	OS PRNG	∼4.5	∼2.3
HQC (this work)	SHAKE-256	ANU QRNG	**5.1**	**2.2**

The HQC commit latency of 5.1 μs is approximately 34% higher than a classical SHAKE-256 commitment (∼3.8 μs) on the same platform. This overhead arises from the buffered QRNG nonce consumption step and is the direct cost of integrating genuine quantum entropy. Verify latency (2.2 μs) is comparable across all schemes because verification is a pure hash recomputation with no entropy generation. BLAKE3 achieves the lowest latency due to its SIMD-optimised design, but provides no quantum entropy and relies on OS PRNG randomness. The HQC framework thus incurs a modest and bounded latency premium in exchange for quantum-grade nonce entropy and the formal hiding guarantees of Theorem 2.

Remark 11 (Lattice-based baseline)**.** *A direct comparison with Dilithium-based or Ring-LWE commitment constructions was not performed in this study, as no lightweight Python reference implementation was available for fair comparison on the Raspberry Pi 4. Based on published benchmarks [14], lattice-based commitments incur commit latencies in the range* 1–10 *ms on similar ARM platforms, placing them approximately* 200× *slower than HQC for this use case. A compiled-language (C/Rust) comparative benchmark is identified as future work.*

4.11. Summary of Results

The experimental evaluation indicates that the HQC framework achieves low-latency operation, scalable aggregation performance, and practical embedded-system compatibility, as shown in Table 10, including the fact that it provides true quantum entropy integration, a commitment latency of microseconds or less, strong replay resistance, scalable hierarchical aggregation, and low embedded resource overhead.

Table 10. Summary of Experimental Results

Metric	Value
Commit Latency (Mean)	5.1 μs
Commit Latency (99th Percentile)	16.2 μs
Verify Latency (Mean)	2.2 μs
Correctness	10000/10000
Successful Forgeries	0/10000
Replay Attacks Rejected	10000/10000
Hiding KS-Test p-value	0.7947
Quantum Min-Entropy	7.1844 bits/byte
Merkle Root (1024 leaves)	1.640 ms
Quantum Entropy Source	ANU QRNG

 doi:10.20944/preprints202606.1104.v1

The findings show that quantum-entropy-assisted commitment architectures can be implemented in an autonomous sensor and edge-computing setting without the need for quantum communication networks.

4.12. Deployment Case Study: Autonomous Drone Swarm Monitoring

To validate the HQC framework in a realistic agentic setting, we present a deployment case study modelling a swarm of $n = 64$ autonomous aerial drones performing coordinated environmental monitoring over a 500×500 m grid.

4.12.1. Scenario Description

Each drone S_i samples atmospheric sensor data (temperature, pressure, particulate matter) at $f_s = 2$ Hz and transmits committed readings to a ground-based gateway cluster $\{G_1, G_2\}$, each serving $n/2 = 32$ drones. A central verifier V (cloud-based server) maintains an auditable log of all commitment epochs. The deployment scenario is motivated by the falsification threat described in Section 1: a compromised drone may attempt to retroactively alter its reported sensor readings or flight path after a collision event is investigated.

4.13. Commitment Rate and Epoch Budget

At $f_s = 2$ Hz, each drone generates 2 commitments per second. With the HQC mean commit latency of 5.1 μs, the per-drone cryptographic overhead is:

$$\text{overhead} = 2 \times 5.1\,\mu\text{s} = 10.2\,\mu\text{s/s} \approx 0.001\% \text{ of the 1-second sampling cycle.} \tag{17}$$

This is negligible relative to sensor acquisition, communication, and flight control workloads.

4.13.1. Gateway Aggregation

Each gateway aggregates $k = 32$ drone commitments per epoch ($T_{\text{epoch}} = 1000$ ms). From Table 6, the Merkle root construction for $k = 32$ leaves requires 0.045 ms, well within the epoch window. The gateway forwards a single 32-byte Merkle root to the verifier rather than 32 individual commitments, reducing uplink bandwidth by a factor of $32\times$.

4.13.2. Verifier Audit Load

The verifier processes two Merkle roots per epoch (one from each gateway). Each root verification requires $\log_2 32 = 5$ hash evaluations, totalling 10 hash operations per epoch for the full swarm. At a verification latency of 2.2 μs per hash evaluation, the verifier's cryptographic load is 22 μs per epoch, representing negligible computational overhead on commodity server hardware.

4.13.3. Non-Repudiation Guarantee

If a post-incident investigation requires verifying drone S_i's reported reading m_i at time t, the verifier retrieves the stored Merkle root $root_j$ for the relevant epoch and the authentication path $path_i$. Given the opening (m_i, r_i, t_i) provided by S_i, the verifier recomputes $C_i = H_k(m_i \| r_i \| t_i)$ and checks it against $root_j$ using $path_i$ in $O(\log k)$ steps. Any discrepancy between the claimed reading and the committed value is detected with probability $1 - \text{negl}(\lambda)$ by Proposition 4, providing a cryptographic audit trail that is binding under SHAKE-256 collision resistance.

4.13.4. Resource Footprint Per Drone

Each drone maintains a secret state $st = (m_i, r_i, t_i)$ of 96 bytes (Table 7) per commitment, plus a 32-byte commitment value. At $f_s = 2$ Hz, the per-drone per-second storage overhead is $2 \times (96 + 32) = 256$ bytes, well within the memory constraints of typical flight controllers (e.g., Pixhawk 6C: 2 MB RAM).

 doi:10.20944/preprints202606.1104.v1

4.13.5. Case Study Summary

Table 11 summarises the key deployment metrics.

Table 11. Deployment case study summary: 64-drone swarm, $f_s = 2\,\text{Hz}$, $T_{\text{epoch}} = 1000\,\text{ms}$.

Metric	Value
Number of drones	64
Commitment rate per drone	2 per second
Per-drone crypto overhead	$\sim$0.001% of cycle time
Merkle root build (32 leaves)	0.045 ms
Verifier hash ops per epoch	10
Verifier crypto load	22 μs per epoch
Gateway uplink reduction	32$\times$
Per-drone RAM overhead	256 bytes/s
Non-repudiation guarantee	$1 - \text{negl}(\lambda)$

The case study demonstrates that the HQC framework scales gracefully to realistic agentic deployments without imposing meaningful computational or storage burden on resource-constrained nodes. The non-repudiation guarantee provided by the commitment layer is particularly valuable in safety-critical scenarios such as drone collision investigations, where retroactive data falsification poses a direct operational and liability risk.

5. Discussion

The HQC framework is thus able to incorporate a true quantum entropy source derived from the ANU QRNG in a classical commitment architecture without the need for quantum communication infrastructure. Experimental results on a Raspberry Pi 4 demonstrate microsecond-scale latency (5.1 μs commit, 2.2 μs verify) and scalable Merkle aggregation (1.640 ms for 1024 leaves). Experimental validation over 10,000 trials did not produce any successful forgery events and complete replay rejection, while the KS-test ($p = 0.7947$) confirmed no statistical distinguishability between commitment distributions. The obtained quantum min-entropy of 7.1844 bits per byte is in good agreement with high-quality randomness for cryptographic nonces generation. A key observation from the study is that HQC does not require fully quantum communication infrastructure. Instead, selective integration of quantum-generated entropy provides a practical intermediate approach for embedded and IoT environments where computational and hardware resources are constrained.

6. Conclusions

This paper proposed a lightweight Quantum-Entropy-Assisted Hybrid Commitment (HQC) architecture for autonomous sensor networks and edge-based intelligent systems, called Quantum-Entropy-Assisted Hybrid Commitment (HQC), for autonomous sensor networks and edge-based intelligent systems. The proposed scheme combines the actual quantum randomness with a classical commitment mechanism, without requiring quantum communication infrastructure or specialized quantum networking hardware, so that the protocol can be implemented in embedded systems. The experimental results with the ANU QRNG entropy and the Raspberry Pi 4 implementation confirmed sub-microsecond level commitment latency, efficient hierarchical aggregation, high replay resistance and low computational overhead. The framework had a mean commitment latency of 5.1 µs, a mean verification latency of 2.2 µs, and the Merkle aggregation was scalable with a maximum of 1.640 ms for 1024 commitments. The security validation was performed, with zero successful forgery, complete replay rejection and empirically consistent hiding behavior. The measured entropy quality also verified the suitability of the quantum-generated randomness for the generation of random numbers (nonces) in the considered architecture.

The findings show that systems of practical quantum-enhanced cryptographic systems can be built without a complete quantum communication infrastructure. Overall, the HQC framework

 doi:10.20944/preprints202606.1104.v1

demonstrates that quantum-generated entropy can be integrated into lightweight commitment architectures without requiring fully quantum communication infrastructure. The proposed design maintains compatibility with conventional embedded hardware while supporting scalable aggregation and low-latency execution, making it suitable for autonomous sensing and edge-security applications.

Author Contributions: Conceptualization, A.S.E and B.U.S; methodology, O.I., A.S.E.; software, B.U.S; validation, B.U.S and P.A; formal analysis, A.S.E; investigation, O.D.O., P.A; resources, O.I; data curation, A.S.E; writing—original draft preparation, A.S.E, O.D.O., O. I.; writing—review and editing, B.U.A; visualization, P.A; supervision, B.U.S; project administration, A.S.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors thank the ANU QRNG group for providing open access to the quantum entropy API.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QRNG	Quantum Random Number Generator
RNG	Random Number Generator
CRHF	Collision-Resistant Hash Function
HQC	Hybrid Quantum Commitment
TPM	Trusted Platform Module
PPT	Probabilistic Polynomial Time
QPT	Quantum Polynomial Time
ANU	Australian National University
QROM	Quantum Random Oracle Model
ROM	Random Oracle Model
LHL	Leftover Hash Lemma
KS	Kolmogorov–Smirnov
IoT	Internet of Things

Appendix A

The complete Python simulation code for the Hybrid Quantum Commitment (HQC) protocol is provided as supplementary material. The code implements:

- Full commit-verify cycle (Algorithms 1–5)
- Simulated QRNG with configurable latency
- Merkle tree aggregation (Algorithm 7)
- Security property tests (binding, hiding, freshness, Merkle)
- Performance benchmarking suite

The simulation is available at:

- GitHub: https://github.com/abassam11/hqc.git

Appendix A.1. Running Instructions

```
# Install dependencies
pip install matplotlib numpy scipy
```

 doi:10.20944/preprints202606.1104.v1


```
# Run security validation
python hqc_security_fixed.py

# Run hardware-accurate simulation
python hqc_simulation_corrected.py
```

References

1. Luo, M.X.; Wang, X. Unconditionally secure quantum bit commitment: revised. *Cryptology ePrint Archive* **2020**.
2. Yan, J. General properties of quantum bit commitments. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2022, pp. 628–657.
3. Lu, Y.; Zhang, S.; Liu, C.; Zhang, R.; Ai, B.; Niyato, D.; Ni, W.; Wang, X.; Jamalipour, A. Agentic graph neural networks for wireless communications and networking towards edge general intelligence: A survey. *IEEE Communications Surveys & Tutorials* **2026**.
4. Coelho, A.; Ribeiro, P.; Fontes, H.; Campos, R. A4FN: an Agentic AI Architecture for Autonomous Flying Networks. In Proceedings of the 2025 IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). IEEE, 2025, pp. 1–6.
5. Crépeau, C. Commitment. In *Encyclopedia of Cryptography, Security and Privacy*; Springer, 2025; pp. 384–386.
6. Wong, H.Y. Shor's algorithm. In *Introduction to quantum computing: from a layperson to a programmer in 30 steps*; Springer, 2023; pp. 289–298.
7. Broadbent, A.; Culf, E. Rigidity for monogamy-of-entanglement games. *arXiv preprint arXiv:2111.08081* **2021**.
8. Morimae, T.; Yamakawa, T. Quantum commitments and signatures without one-way functions. In Proceedings of the Annual International Cryptology Conference. Springer, 2022, pp. 269–295.
9. Naor, M. Bit commitment using pseudorandomness. *Journal of cryptology* **1991**, *4*, 151–158.
10. Crépeau, C.; Légaré, F.; Salvail, L. How to convert the flavor of a quantum bit commitment. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2001, pp. 60–77.
11. Mayers, D. Unconditionally secure quantum bit commitment is impossible. *Physical review letters* **1997**, *78*, 3414.
12. Gunn, S.; Tauman Kalai, Y.; Natarajan, A.; Villányi, Á. Classical commitments to quantum states. In Proceedings of the Proceedings of the 57th Annual ACM Symposium on Theory of Computing, 2025, pp. 234–244.
13. Del Pino, R.; Lyubashevsky, V.; Seiler, G. Short discrete log proofs for FHE and ring-LWE ciphertexts. In Proceedings of the IACR International Workshop on Public Key Cryptography. Springer, 2019, pp. 344–373.
14. Imran, M.; Aikata, A.; Roy, S.S.; Pagliarini, S. High-speed design of post quantum cryptography with optimized hashing and multiplication. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2023**, *71*, 847–851.
15. Das, S.; Krause, S.; Giering, K.U.; Pousa, R.J.; Bassoli, R.; Fitzek, F.H. Leveraging quantum uncertainty: Quantum randomness through the lens of classical communication networks. *Computer Networks* **2024**, *254*, 110781.
16. Tomamichel, M.; Colbeck, R.; Renner, R. A fully quantum asymptotic equipartition property. *IEEE Trans. Inf. Theory* **2012**, *58*, 1840–1847.
17. Iavich, M.; Kuchukhidze, T.; Okhrimenko, T. Verkle Tree-based Post-Quantum Digital Signature Scheme using Stateless Updatable Vector Commitment. In Proceedings of the CPITS II, 2023, pp. 157–166.
18. Lee, C.; Sohn, I.; Lee, W. Eavesdropping detection in BB84 quantum key distribution protocols. *IEEE Transactions on Network and Service Management* **2022**, *19*, 2689–2701.