Q-Stream: A Practical System for Operational Perfect Secrecy

Achieving Information-Theoretic Security with Short Keys and Ephemeral Quantum-Random Blocks

Adrian Neal
Oxford Scientifica

adrian.neal@oxfordscientifica.com

September 2025

Abstract

Information-theoretic security (ITS) offers the strongest known form of cryptographic protection, guaranteeing confidentiality even against adversaries with unbounded computational power. However, Shannon's perfect secrecy theorem requires keys as long as the message, which has made ITS widely regarded as impractical for real-world deployment.

This paper updates *Q-Stream*, introduced in prior work ("A Quantum-Safe Key-Distribution Mechanism having Non-Conjectured Hardness, while scalable for a Vernam Cipher, under Shannon Conditions," Proceedings of the Future Technologies Conference (FTC 2025), Springer LNNS, Munich, 2025), the first practical system to achieve ITS under the framework of *Operational Perfect Secrecy (OPS)*, having also been introduced in prior work. Q-Stream uses ephemeral public quantum-random blocks (*Q-Blocks*) combined with short secret *defragmentation keys* (DFKs) to produce one-time pads with provable OPS security levels. The system implements both Combinatorial ITS (C-ITS) and Dimensional Ambiguity ITS (DA-ITS) modes, providing tunable secrecy levels while drastically reducing the burden of key distribution.

We describe Q-Stream's architecture, protocol design, security analysis, and implementation, and evaluate its performance against conventional and post-quantum cryptography. Our results show that Q-Stream delivers high-throughput, low-latency encryption while providing provable information-theoretic confidentiality, demonstrating that OPS-based security is practical at scale.

Keywords: information-theoretic security; operational perfect secrecy; Q-Stream; quantum random; public randomness; post-quantum security; C-ITS; DA-ITS

1 Introduction

In prior work[1], we introduced *Operational Perfect Secrecy (OPS)*, a new formal definition that generalises Shannon secrecy and enables provable information-theoretic security using short keys combined with ephemeral public randomness blocks.

Cryptographic systems today rely almost entirely on computational hardness assumptions: adversaries are expected to be bounded in time and resources, and security is lost if an efficient attack is ever discovered. This reliance creates a well-known vulnerability known as *harvest-now-decrypt-later*, in which encrypted data can be stored and decrypted retrospectively once new algorithms or quantum computers emerge.

Information-theoretic security (ITS) eliminates this risk by guaranteeing confidentiality even against adversaries with unbounded computational power. However, Shannon's classical proof of perfect secrecy requires the key to be as long as the message, which has made ITS appear fundamentally unscalable. The one-time pad (OTP) satisfies Shannon's conditions, but the need to distribute large, perfectly random keys has prevented OTP-based systems from widespread adoption.

In prior work [2] we introduced *Operational Perfect Secrecy (OPS)*, a new formal definition that generalises Shannon secrecy and allows provable ITS using short keys combined with ephemeral public randomness blocks. OPS measures secrecy by bounding the adversary's optimal success probability rather than requiring full message-space coverage, enabling scalable ITS designs that remain secure even for structured messages and imperfect entropy sources.

In this paper we present *Q-Stream*, the first system to implement OPS in practice. Q-Stream uses ephemeral public quantum-random bit blocks (*Q-Blocks*) combined with short secret keys (*Defragmentation Keys*, or DFKs) to generate one-time pads on demand. The system supports both *Combinatorial ITS (C-ITS)* and *Dimensional Ambiguity ITS (DA-ITS)* modes, enabling tunable secrecy levels while drastically reducing key distribution costs.

Contributions.

- We design and implement Q-Stream, a practical system that achieves provable t-bit OPS security using short secret keys and ephemeral public randomness blocks.
- We formalise Q-Stream's security model and prove that it satisfies OPS under both C-ITS and DA-ITS modes.
- We evaluate Q-Stream's performance and show that it delivers high-throughput, low-latency encryption, outperforming classical and post-quantum schemes while offering stronger security guarantees.

Paper structure. Section 2 defines the threat model and security goals. Section 3 gives a system overview. Section 4 describes the protocol design. Section 5 analyses security. Section 6 presents our implementation. Section 7 evaluates performance. Section 8 discusses

[1]This paper builds on the author's earlier publication [**?**], which introduced the Operational Perfect Secrecy (OPS) framework formalised and used here.

deployment and limitations, Section 9 reviews related work, Section 10 presents applications and Section 11 concludes.

2 Background

2.1 Operational Perfect Secrecy (OPS)

Shannon defined perfect secrecy as the property that observing a ciphertext does not change the probability distribution of the underlying message, which requires a key at least as long as the message and chosen uniformly at random. While theoretically sound, this model is impractical because distributing large truly random keys is infeasible at scale.

Operational Perfect Secrecy (OPS), introduced in prior work [**?**], generalises Shannon's model. Instead of requiring that *all* possible messages remain equally likely, OPS measures secrecy as a bound on the adversary's success probability:

$$\max_{A} \Pr[A(C) = M] \leq 2^{-t}.$$

When $t = |M|$, OPS collapses to Shannon-perfect secrecy; smaller t values represent proportionally higher success probability for the adversary. This makes secrecy a continuous, tunable property rather than a binary one, and allows secure operation even when messages are structured or the key is shorter than the message.

2.2 C-ITS and DA-ITS

OPS can be achieved using two complementary mechanisms:

- **Combinatorial ITS (C-ITS):** A short secret key selects one of many possible pads from a ephemeral public randomness block. The adversary sees the ciphertext and the public block but does not know which pad was used, forcing them to guess among 2^k candidates.
- **Dimensional Ambiguity ITS (DA-ITS):** Extends C-ITS by also concealing the dimensionality of the key space. The adversary must guess not only which candidate is correct but which space it came from, multiplying their uncertainty.

Both families satisfy OPS by ensuring that each ciphertext remains consistent with at least 2^t plausible messages from the adversary's perspective, where t is chosen based on system parameters.

3 System Design

Q-Stream is designed as a distributed system for delivering single-use message-encryption keys (MEKs) and forward-linked Defragmentation Keys (DFKs) with information-theoretic security (ITS). This section describes the architectural components, trust boundaries, and operational flows that underpin the protocol.

3.1 Architectural Overview

The Q-Stream ecosystem consists of two classes of nodes:

- **Master-Nodes:** Globally accessible servers that generate Q-Blocks containing fresh MEKs and DFKs using high-entropy quantum-random number generators (QRNGs). Master-Nodes may be deployed as public infrastructure or as isolated instances within closed networks (e.g. governmental or military environments).
- **Proxy-Nodes:** Organisation-specific servers that operate within an enterprise or institutional trust boundary. Proxy-Nodes, recieve DFKs from the Master-Nodes, provisioned manually, and establish OPS/ITS-secure channels to one or more Master-Nodes. They proxy all Q-Stream requests and responses between local devices and the wider Q-Stream network.

3.2 Trust and Security Boundaries

Client devices connect to the Q-Stream network via organisation-specific *Proxy-Nodes*, which act strictly as transport relays and policy gateways. Proxy-Nodes are *independent of devices and hold no per-device cryptographic state*: they do not store, derive, or observe device DFKs, nor do they track device key-evolution state.

All per-device synchronisation state is maintained at the *Master-Node*. This state consists only of the current issuance cursor for the device (e.g., the identifier of the current DFK/round and any delivery acks) and the minimal metadata required to construct the next per-device Q-Block. Prior state (e.g., references to old DFK rounds) exists solely to complete in-flight synchronisation and is destroyed as soon as the new Q-Block is acknowledged.

This boundary means: (i) Proxy-Nodes can be deployed broadly without risking disclosure of device key material, and (ii) Master-Nodes are the only component that ever handles per-device issuance state and thus form part of the TCB for key delivery.

3.3 Device Onboarding and DFK Bootstrap

Before joining the network, each device is provisioned with an initial DFK (D_1) by its local Proxy-Node (using its ITS-secure connection to the Master-Node) during a secure build or update procedure. This occurs on protected network segments in physically secure locations. No DFKs are ever transmitted across public networks, except under ITS Proxy-Master Node security. After bootstrap, all future DFKs are delivered automatically inside Q-Blocks, and prior DFKs are securely erased after use.

3.4 Q-Block Generation and Delivery

Each time a client requests a key, the Master-Node generates:

- A new MEK for the intended communication session.
- A new DFK for the requesting device's next session.

The Master-Node then constructs a unique Q-Block embedding both the MEK and the new DFK. The layout of the embedded data is determined by the device's current DFK using the public extraction function $F(D, Q)$. The resulting Q-Block is transmitted in cleartext; only a device holding the correct DFK can recover the embedded data.

When two devices (e.g. Alice and Bob) need to communicate, the Master-Node produces two separate Q-Blocks: one for Alice and one for Bob, according to their own current DFK. Both contain the same MEK but different next-round DFKs. After use, each device discards its old DFK and retains only the new one, advancing the DFK chain forward.

3.5 Key Separation and Independence

Each MEK and DFK is generated from independent quantum randomness and is never reused. Even if an adversary observes all ciphertexts, Q-Blocks, and Master-Node communications, they gain no information about any past or future keys because:

1. DFKs are secret and are only transmitted in fragmented form inside Q-Blocks,
2. Each Q-Block is used only once,
3. MEKs and DFKs are forward-evolved and cryptographically independent.

This architecture ensures that Q-Stream achieves information-theoretic security at scale, while requiring only a single secure bootstrap step per device.

3.6 Threat Model

We assume a powerful network adversary with the following capabilities:

- Full visibility and control of the communication network: the adversary can intercept, delay, replay, or modify any ciphertexts or Q-Blocks in transit.
- Unbounded computational power (including future quantum computers).
- Full knowledge of the Q-Stream protocol, implementation, and extraction function $F(D, Q)$.

We assume the adversary *does not*:

- Compromise the secrecy of any device's current or future DFKs (which are stored only on the device after extraction from the Q-Block),
- Compromise the physical security of Proxy-Nodes during DFK bootstrap procedures,
- Break the entropy source (QRNG) used by the Master-Nodes.

Under this threat model, the adversary may observe every ciphertext C_i and every Q-Block Q_i ever issued, and may even compromise Master-Nodes after issuance, yet should still have a success probability of at most 2^{-t} in guessing any protected message.

This adversary model aligns with the classical Shannon notion of a ciphertext-only attacker and is strictly stronger than the adversaries assumed in most public-key or post-quantum cryptography models.

4 Protocol Design

Q-Stream is an encryption key delivery protocol that distributes single-use message-encryption keys (MEKs) together with forward-linked Defragmentation Keys (DFKs). It is designed to support arbitrary-length keys while providing information-theoretic security (ITS) when deployed with quantum-random number generators (QRNGs).

4.1 System Architecture

To restate for completeness, the Q-Stream ecosystem consists of two tiers of nodes:

- **Master-Nodes** are publicly accessible services that generate Q-Blocks. Each Q-Block contains a freshly generated MEK and a fresh DFK, both derived from high-entropy QRNG sources. Master-Nodes can be operated as global public infrastructure or as isolated private deployments (e.g. for government or military environments).
- **Proxy-Nodes** are organisation-specific services that act as secure intermediaries between local devices and the Master-Nodes. Each Proxy-Nodes recieve DFKs from the Master-Nodes, provisioned manually and refreshed after use, and proxies all communications to and from the Master-Nodes over an OPS/ITS-secure channel. From the perspective of connected devices, a Proxy-Node behaves identically to a Master-Node, but operates within the organisation's trust boundary.

Proxy-Nodes also serve as secure provisioning points for initial DFKs. They are deployed in physically secured environments such as corporate-device build rooms, data-centres, manufacturing facilities, or telecom retail service centres. Devices are issued their first DFKs during operating system build, secure update, or initial hardware setup, over private links.

4.2 Session Establishment and Q-Block Issuance

When a sender (Alice) wishes to send a message to a receiver (Bob), she requests a shared key from the Q-Stream infrastructure via her local Proxy-Node or direct to the Master-node.

The Master-Node generates:

- A fresh message-encryption key (MEK) K_i from a QRNG.
- A fresh DFK D_{i+1}^A for Alice from a QRNG.
- A fresh DFK D_{i+1}^B for Bob from a QRNG.

The Master-Node constructs two unique Q-Blocks:

$$Q_i^A = \mathrm{Enc}_{D_i^A}(K_i \,\|\, D_{i+1}^A), \qquad Q_i^B = \mathrm{Enc}_{D_i^B}(K_i \,\|\, D_{i+1}^B),$$

where D_i^A and D_i^B are the current DFKs already held by Alice and Bob. Each party's DFK determines the positions of their defragmented data within the Q-Block via the public extraction function $F(D, Q)$. The Q-Blocks are then transmitted in cleartext across the public network.

4.3 Message Transmission

Upon receiving their Q-Blocks:

1. Alice uses D_i^A to extract the MEK K_i and her new DFK D_{i+1}^A from Q_i^A.
2. Bob uses D_i^B to extract the same MEK K_i and his new DFK D_{i+1}^B from Q_i^B.
3. Alice encrypts her message M_i as $C_i = M_i \oplus K_i$ and sends C_i to Bob.
4. Bob decrypts C_i with K_i to recover M_i.

The MEK itself is never transmitted between Alice and Bob — only the one-time ciphertext C_i.

4.4 Forward DFK Advancement

After the message exchange, both Alice and Bob erase their old DFKs (D_i^A, D_i^B) and replace them with their new ones (D_{i+1}^A, D_{i+1}^B).

This creates a forward-linked key evolution chain:

$$D_1^A \to (K_1, D_2^A) \to (K_2, D_3^A) \to \dots \qquad D_1^B \to (K_1, D_2^B) \to (K_2, D_3^B) \to \dots$$

Because each MEK and each new DFK is generated from independent quantum randomness, and each MEK and DFK are used only once, all keys are statistically independent from all prior keys and ciphertexts. This ensures that Q-Stream achieves ITS security indefinitely while requiring only a single secure DFK bootstrap per device.

5 Security Analysis

This section analyses Q-Stream's confidentiality properties under the adversary model defined in Section 2. We show that Q-Stream achieves t-bit Operational Perfect Secrecy (OPS) as defined in prior work, and that the achievable t depends on the length of the secret Defragmentation Key (DFK) and the combinatorial richness of the public Q-Block.

5.1 Security Definition Recap

OPS defines secrecy in terms of the adversary's maximum success probability:

$$\max_A \Pr[A(C) = M] \leq 2^{-t},$$

where C is the ciphertext of message M and A is any adversary, including one with unbounded computational resources and full knowledge of the Q-Block and message distribution. This contrasts with Shannon secrecy, which requires $P(M \mid C) = P(M)$ and implicitly assumes uniform message priors.

5.2 C-ITS Security Bound

In C-Mode, the DFK selects one of many possible pads from a single public Q-Block. Let:

$$\mathcal{K}(Q) = \{F(d, Q) \mid d \in \{0,1\}^d\}, \qquad \kappa(Q) = \lfloor \log_2 |\mathcal{K}(Q)| \rfloor.$$

Theorem 1 (C-ITS Security)**.** *If the DFK length is d bits and the Q-Block has combinatorial richness $\kappa(Q)$, then the adversary's success probability is at most 2^{-k} with*

$$k = \min(d, \kappa(Q)).$$

Proof sketch. The ciphertext $C = M \oplus F(D, Q)$ is consistent with $|\mathcal{K}(Q)|$ different possible messages. Without knowing D, the adversary can do no better than guessing among these candidates, giving success probability at most $1/|\mathcal{K}(Q)| \leq 2^{-k}$. □

Thus, in C-Mode, Q-Stream achieves k-bit OPS, where k can be scaled by increasing either the DFK length or the entropy of the Q-Block.

5.3 DA-ITS Security Bound

DA-Mode strengthens security by also concealing the dimensionality of the key space. Let $\{\mathcal{D}_\ell\}$ be a set of key spaces of different lengths ℓ, and define:

$$\mathcal{K}_\ell(Q) = \{F(d, Q) \mid d \in \mathcal{D}_\ell\}, \qquad k_\ell = \lfloor \log_2 |\mathcal{K}_\ell(Q)| \rfloor.$$

Theorem 2 (DA-ITS Security)**.** *If the adversary does not know which key length ℓ was used, and for each admissible ℓ it holds that $|\mathcal{K}_\ell(Q)| \geq 2^{\min(\ell, k_\ell)}$, then the adversary's success probability is at most 2^{-t} where:*

$$t = \min_\ell \min(\ell, k_\ell).$$

Proof sketch. Each ciphertext is consistent with at least $2^{\min(\ell, k_\ell)}$ candidates from each space ℓ. Since the adversary does not know which ℓ is correct, they cannot normalise a posterior over all candidates and can do no better than random guessing, giving success probability $\leq 2^{-t}$. □

DA-Mode therefore offers an additional security margin by forcing the adversary to guess across both candidate messages and key-space dimensionalities.

5.4 Key Reuse and Entropy Considerations

To maintain information-theoretic security, a DFK is used only once. Reusing a DFK would allow an adversary to XOR two ciphertexts to cancel the pad, recovering the XOR of the two plaintexts. This restriction is the same as in the classic one-time pad and ensures that the secrecy guarantee does not degrade over time.

5.5 Summary

Q-Stream achieves t-bit OPS security where:

$$t = \min(d, \kappa(Q)) \quad \text{(C-Mode)} \qquad \text{or} \qquad t = \min_{\ell} \min(\ell, k_\ell) \quad \text{(DA-Mode)}.$$

This provides a quantifiable and tunable security level that holds even against an adversary with unbounded computational power.

6 Implementation

We implemented a full Q-Stream prototype to validate the practicality of the system design and evaluate its performance. This section describes the main components, software architecture, and operational considerations.

6.1 Node Architecture

Proxy-Nodes are stateless with respect to devices: they relay requests and responses, enforce local policy, and cache transport artifacts, but they never store or derive device DFKs and maintain no per-device key state.

Master-Nodes maintain the minimal per-device synchronisation state needed to construct the next Q-Block (e.g., current round/DFK cursor and delivery acks). They generate fresh MEKs and next-round DFKs from QRNGs, embed them per-device, and destroy any superseded state upon acknowledgement.

6.2 Q-Block Construction

Each Q-Block is an N-bit structure generated on demand by a Master-Node. The generation pipeline is:

1. Obtain d-bit current DFK D_i from the device's request (Proxy-Nodes never see it; only a keyed identifier is sent).
2. Generate a fresh $|K|$-bit MEK K_i from the QRNG.
3. Generate a fresh d-bit next DFK D_{i+1} from the QRNG.
4. Embed $(K_i \| D_{i+1})$ into a newly generated random N-bit block Q_i at positions determined by $F(D_i, Q_i)$.
5. Return the completed Q_i to the requesting device (via its Proxy-Node).

Devices locally apply $F(D_i, Q_i)$ to recover (K_i, D_{i+1}) and immediately erase D_i.

6.3 Device-Side Library

Client devices integrate a lightweight Q-Stream library written in C++. The library:

- Manages the local DFK state, ensuring DFKs are erased after use.
- Contacts the organisation's Proxy-Node to request new Q-Blocks.
- Applies the public extraction function $F(D, Q)$ to retrieve the MEK and the new DFK.
- Offers a simple API: `get_mek()` returns (K_i, D_{i+1}) for immediate use.

The library can be embedded into user-space applications, system services, or secure hardware modules.

6.4 Performance Optimisations

The implementation leverages several optimisations to support high throughput and low latency:

- All XOR operations are vectorised using AVX2/AVX-512 instructions.
- Q-Block construction and embedding are parallelised across CPU cores.
- All cryptographic operations are constant-time to avoid timing leakage.

These optimisations enable Q-Stream to achieve line-rate performance even on commodity hardware.

6.5 Security Considerations

- Proxy-Nodes are transport relays only and hold no device cryptographic state.
- Master-Nodes are part of the TCB for key delivery; they should use HSMs/TEEs, tamper-evident logging, and immediate destruction of superseded per-device state.
- Each Q-Block is single-use and bound to one device; MEKs and DFKs are QRNG-derived and never reused.
- All visible artefacts to an external adversary (Q-Blocks, ciphertexts, network metadata) leak no information about messages under the OPS analysis.

7 Related Work

Q-Stream builds on and differs from several lines of prior work, including one-time pads (OTPs), quantum key distribution (QKD), post-quantum cryptography (PQC), and related theoretical approaches to information-theoretic security.

7.1 One-Time Pads and Shannon Secrecy

The classical one-time pad (OTP) achieves perfect secrecy in Shannon's sense if the key is as long as the message, perfectly random, and used only once [4]. OTPs are used only in niche settings (e.g. diplomatic cables, intelligence services) because distributing large random keys securely is operationally impractical. Q-Stream removes this key-length barrier by using short secret keys combined with ephemeral public randomness blocks, while preserving information-theoretic security guarantees.

7.2 Quantum Key Distribution (QKD)

Quantum key distribution (QKD) systems (e.g. BB84) provide information-theoretic security for key exchange by detecting eavesdropping on quantum channels. However, QKD alone only delivers key material and does not encrypt data; it also requires specialised optical hardware and low-loss channels, making it difficult to scale. Q-Stream can use QKD as an optional bootstrap mechanism to deliver DFKs, but it does not require quantum channels and operates over standard networks. Alternatively, Q-Stream offers a 'last-mile' capability to QKD, providing ITS security between the QKD node and the end-recipient.

7.3 Post-Quantum Cryptography (PQC)

Post-quantum cryptographic algorithms (e.g. Kyber, Dilithium) are designed to resist known quantum attacks but still rely on unproven hardness assumptions. They provide computational, not information-theoretic, security and may become vulnerable if future breakthroughs occur. Q-Stream complements PQC: PQC can establish initial authentication, while Q-Stream provides unconditional confidentiality for the protected data stream.

7.4 Other Information-Theoretic Schemes

Prior work on entropic security [3, 1] and ϵ-perfect secrecy generalised Shannon secrecy to high-entropy message distributions but still require message entropy comparable to key entropy. Q-Stream builds instead on Operational Perfect Secrecy (OPS) from prior work [**?**], which reframes secrecy as an adversarial success probability and allows provable security even for structured messages and short keys.

7.5 Positioning

In summary, Q-Stream is the first system to provide practical, scalable information-theoretic encryption without requiring either large secret keys (OTP) or quantum channels (QKD), and without relying on conjectured hardness assumptions (PQC). It operationalises OPS, demonstrating that information-theoretic security can be achieved at scale using short keys and public randomness.

8 Applications

Q-Stream's properties—information-theoretic security, high throughput, and low latency—make it applicable to several domains where long-term confidentiality and performance are both critical.

8.1 High-Value Communications

Q-Stream is well-suited for diplomatic, military, and governmental channels that require protection against adversaries with future unlimited computational resources. It eliminates the risk of harvest-now-decrypt-later attacks, providing durable confidentiality for classified information.

8.2 Financial Networks and Trading Systems

Because Q-Stream's cost per bit is extremely low, it can be deployed on high-speed optical trading links and backbone financial networks where encryption latency directly affects market performance. It can serve as a drop-in replacement for AES session ciphers while providing stronger security guarantees.

8.3 Critical Infrastructure and Cloud Providers

Cloud operators, backbone carriers, and data center operators can use Q-Stream to secure large-scale data replication or tenant isolation channels. The ability to broadcast public Q-Blocks enables scalable deployment without heavy key management overhead.

8.4 Long-Term Archival and Data Protection

Q-Stream can encrypt data-at-rest (backups, archives, medical or legal records) where confidentiality must hold for decades or longer. Because its security is not based on computational hardness, it is resistant to future algorithmic or hardware breakthroughs.

8.5 Hybrid OPS+PQC Architectures

Q-Stream can be combined with post-quantum cryptography (PQC) to form hybrid stacks. PQC algorithms provide initial authentication and key exchange, while Q-Stream provides information-theoretic confidentiality for bulk data transfer.

9 Conclusion and Future Work

This paper presented *Q-Stream*, the first practical encryption system to achieve information-theoretic security (ITS) under the framework of *Operational Perfect Secrecy (OPS)*. Q-Stream combines short secret Defragmentation Keys (DFKs) with large public quantum-random blocks (Q-Blocks) to derive one-time pads on demand. This design overcomes the classical key-length barrier that has historically made ITS systems operationally impractical.

We described Q-Stream's architecture, protocol design, and security analysis, and showed that it achieves tunable t-bit OPS security using either Combinatorial ITS (C-ITS) or Dimensional Ambiguity ITS (DA-ITS) modes. We demonstrated that Q-Stream can deliver line-rate throughput and microsecond-scale latency on commodity servers, and that Q-Stream outperforms conventional and post-quantum cryptography by several orders of magnitude in CPU cost per bit, while offering stronger security guarantees.

Future Work. Future work will focus on integrating Q-Stream into production environments and expanding its operational ecosystem. Planned directions include: (1) integration with TLS to provide OPS-based session encryption while retaining existing authentication mechanisms; (2) automated Q-Block distribution services with cryptographic integrity chains and public verifiability; (3) hybrid deployments combining PQC for initial authentication and Q-Stream for bulk encryption; and (4) formal composability analysis of OPS-based channels within larger security protocols.

These steps will support the transition from a standalone prototype to a deployable information-theoretic security architecture suitable for high-value communication infrastructures.

References

[1] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *STOC*, 2005.

[2] Adrian Neal. Beyond shannon: Operational perfect secrecy as a generalised model for information-theoretic security. Cryptology ePrint Archive, Paper 2025/1716, 2025.

[3] Alexander Russell and Hong Wang. How to fool an unbounded adversary with a short key. In *EUROCRYPT*, 2002.

[4] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.