OTP-Grade ITS Rekeying for Defence without Bulk Key Files or Specialised Quantum Links

Key Rotation from Public Quantum-Noise Epochs via a Self-Ratcheting Secret State

Adrian Neal
Oxford Scientifica

adrian.neal@oxfordscientifica.com

February 2026

Abstract

The one-time pad (OTP) provides Shannon-perfect secrecy [5], but is operationally constrained by distributing key material at scale. Prior q-stream work used a large quantum-random block augmented with recipient-specific embedded markers to locate key-generation material [4]. We update q-stream to a pure-noise model: the distributor transmits authenticated quantum-noise epochs, and recipients use a pre-established secret derivation key (DFK) to derive a fresh message key and refresh the DFK in a single ratcheting step. Under explicit assumptions on adversary side-information and state secrecy, we formalize a bounded information-theoretic claim: per-epoch derived keys are statistically indistinguishable from uniform to a passive eavesdropper, with bounds determined by the provisioned/injected secret entropy and any leakage exposure [1, 2]. Finally, we introduce QS-OTP, an operational OTP mode for manually initialized deployments in which a reservoir of preloaded DFK states enables one-time keystream use with low amortized bandwidth overhead while enforcing OTP-grade non-reuse semantics.

1 Introduction

Many high-assurance operational environments already accept manual initialization of secret material (e.g., key ceremonies, HSM injection, couriered keys, or tactical key distribution). However, these environments typically do *not* obtain scalable, automatic key rotation with information-theoretic (Shannon-style) assurances without distributing large key files. This paper positions q-stream v2 as a *post-deployment* key-rotation mechanism: a small (but possibly structured) initial secret enables automatic, epoch-based derivation of fresh traffic keys from transmitted quantum noise.

What changes in v2. In the earlier q-stream design, the distributor modified a QRNG block by inserting rotating recipient identifiers and metadata that revealed locations of key-generation material inside the block, and recipients searched the block to recover those locations and reconstruct key-generation material [4]. In v2, the distributor transmits only quantum noise epochs, and all secret keying material is derived locally using a secret DFK ratchet that simultaneously produces (a) working keys and (b) the next DFK state.

What this enables. This evolution enables a cleaner security posture: the system is no longer framed as "hiding" marker structure inside noise; rather it is framed as *local key derivation from high-entropy epochs under a secret ratcheting state*, permitting bounded information-theoretic claims under explicit assumptions.

2 Prior Work: q-stream v1 Summary (Embedding Markers into Noise)

This section concisely summarizes the earlier construction to clarify the evolution. In q-stream v1, the distributor generated a large random block R using a QRNG (e.g., $|R| = 2{,}097{,}152$ bits), then inserted recipient-specific rotating identifiers (rUIDs) and a small suffix encoding locations/lengths/order for key-generation material (kGen), producing a modified block R' transmitted to both parties [4]. Recipients searched for their rUIDs, decoded the embedded suffix, extracted kGen blocks, and locally generated a key (of message-length) for OTP-style XOR encryption (with an HMAC for integrity) [4]. The v1 paper also presented a "combinatorial hardness" argument based on placements of multiple kGen blocks inside a large QRNG block [4].

2.1 Limitations motivating v2

The v1 construction is naturally expressed as "structure embedded in noise + search," and its most prominent argumentation emphasizes combinatorial hardness. While valuable, this framing is not the cleanest way to express bounded information-theoretic guarantees for *automatic* ongoing key rotation. v2 is designed to make the information-theoretic statement explicit and modular.

3 System Model and Design Goals

3.1 Roles and channels

A distributor broadcasts a sequence of quantum-noise epochs $\{Q_t\}$. Recipients hold secret state and derive keys locally. Epochs must be integrity-protected (e.g., signature/MAC) to prevent substitution/replay, but secrecy is obtained from the secret state and entropy assumptions.

3.2 Design goals

- **Manual initialization, automatic rotation thereafter:** allow a trusted initial DFK load; then rotate without further secret distribution.
- **Bounded information-theoretic claims:** security statements in terms of statistical distance under explicit entropy conditions.
- **Operational simplicity:** transmit only noise epochs; avoid embedding recipient markers in the epoch payload.
- **OTP-grade non-reuse semantics:** support a mode in which keystream is never reused, enabling OTP-like encryption in practice.

4 Threat Model and Assumptions

4.1 Adversary capabilities

We assume a passive network adversary who observes all transmitted epochs $\{Q_t\}$ and ciphertexts, and can store them indefinitely. We separately consider active adversaries for integrity (epoch substitution/replay), handled by standard authentication of epochs.

4.2 Entropy and secrecy assumptions (explicit)

This paper uses *bounded* information-theoretic statements: they hold as long as (i) the secret state remains uncompromised and (ii) the construction consumes/injects sufficient adversary-unknown entropy per the chosen operational mode. We make no claim of "unconditional forever" security.

5 Protocol v2: Pure Noise Transmission with a Self-Ratcheting DFK

5.1 Notation

Let $Q_t \in \{0,1\}^N$ denote the t-th quantum-noise epoch (publicly transmitted but integrity-protected). Let $\mathsf{DFK}_t \in \{0,1\}^\lambda$ denote the recipient secret ratchet state at epoch t. Let ctx_t be context/domain separation data (epoch counter, direction, application label, recipient group, etc.).

5.2 Ratchet interface

We define a ratchet function F producing a working key and refreshed state:

$$(\mathsf{MEK}_t, \mathsf{DFK}_{t+1}) \leftarrow \mathsf{F}(\mathsf{DFK}_t, Q_t, ctx_t).$$

5.3 Explicit ratchet algorithm (v2) and QS-OTP reservoir mode

We present an explicit ratchet that implements secret-index path extraction with disjoint index sets for (i) pad/key derivation and (ii) state refresh, with strict non-reuse.

Index scheduling primitive. Let $\mathsf{Permute}(S, ctx, N)$ denote a deterministic procedure that outputs a permutation π of $\{0, \ldots, N-1\}$ derived from secret schedule seed S and context ctx. (Any implementation that is deterministic, keyed by secret state, and enforces non-reuse is acceptable; the paper's security claims do not require computational PRF assumptions about $\mathsf{Permute}$, only secrecy of S and non-reuse of outputs.)

Disjoint slicing. Given π, define:

$$I = (\pi[0], \ldots, \pi[\ell-1]), \qquad \tilde{I} = (\pi[\ell], \ldots, \pi[\ell+\lambda-1]),$$

ensuring $I \cap \tilde{I} = \emptyset$ by construction.

Algorithm 2 QS-OTP reservoir mode (m independent streams)

Require: Epoch $Q_t \in \{0,1\}^N$; reservoir $\{\mathsf{DFK}_t^{(i)} = (M_t^{(i)}, S_t^{(i)}, c_t^{(i)})\}_{i=1}^m$
Ensure: A batch of pads/keys $\{\mathsf{MEK}_t^{(i)}\}_{i=1}^m$ and refreshed reservoir $\{\mathsf{DFK}_{t+1}^{(i)}\}_{i=1}^m$
1: **for** $i \leftarrow 1$ to m **do**
2: $L \leftarrow$ ("QS-OTP", i) ▷ domain separation by stream index
3: $(\mathsf{MEK}_t^{(i)}, \mathsf{DFK}_{t+1}^{(i)}) \leftarrow$ Algorithm 1$(Q_t, \mathsf{DFK}_t^{(i)}, L)$
4: **Enforce:** use $\mathsf{MEK}_t^{(i)}$ at most once; never reuse $(i, c_t^{(i)}, \text{direction}, \text{epoch\_id})$
5: **end for**
6: **return** $\{\mathsf{MEK}_t^{(i)}\}_{i=1}^m$, $\{\mathsf{DFK}_{t+1}^{(i)}\}_{i=1}^m$

Algorithm 1 Q-Stream v2 secret-index ratchet (single stream)

Require: Epoch $Q_t \in \{0,1\}^N$; state $\mathsf{DFK}_t = (M_t, S_t, c_t)$ with $M_t \in \{0,1\}^\ell$, $S_t \in \{0,1\}^\sigma$; context label L
Ensure: Pad/key $\mathsf{MEK}_t \in \{0,1\}^\ell$; refreshed state DFK_{t+1}

$ctx_t \leftarrow (L, c_t, \text{direction}, \text{epoch\_id})$ ▷ public, unique per use
$\pi \leftarrow \mathsf{Permute}(S_t, ctx_t, N)$
$I \leftarrow (\pi[0], \ldots, \pi[\ell-1])$; $\tilde{I} \leftarrow (\pi[\ell], \ldots, \pi[\ell+\lambda-1])$
$X_t \leftarrow Q_t[I[0]] \| \cdots \| Q_t[I[\ell-1]]$
$\mathsf{MEK}_t \leftarrow X_t \oplus M_t$
$Y_t \leftarrow Q_t[\tilde{I}[0]] \| \cdots \| Q_t[\tilde{I}[\lambda-1]]$
$\tilde{M}_t \leftarrow \mathsf{MaskUpd}(M_t, S_t, ctx_t)$ ▷ secret-state re-masking
$M_{t+1} \leftarrow Y_t \oplus \tilde{M}_t$
$S_{t+1} \leftarrow \mathsf{SeedUpd}(S_t, M_t, Y_t, ctx_t)$ ▷ schedule evolution
$c_{t+1} \leftarrow c_t + 1$
return $(\mathsf{MEK}_t, \mathsf{DFK}_{t+1} = (M_{t+1}, S_{t+1}, c_{t+1}))$

In practice, Permute and update functions may be instantiated using standard conservative primitives (e.g., AES/ChaCha-based permutation generation) without changing the information-theoretic secrecy bound, which is driven by state unpredictability and exposure accounting; the choice primarily affects robustness against implementation leakage and schedule collisions.

5.4 A concrete instantiation (extract-then-derive)

To support clean separation and analysis, we adopt the pattern:

$$s_t \leftarrow \mathsf{Extract}(\mathsf{DFK}_t, Q_t, ctx_t), \tag{1}$$
$$\mathsf{MEK}_t \leftarrow \mathsf{Derive}(s_t, \text{“MEK”}, \ell), \tag{2}$$
$$\mathsf{DFK}_{t+1} \leftarrow \mathsf{Derive}(s_t, \text{“DFK”}, \lambda). \tag{3}$$

Algorithm 3 Per-epoch derivation and DFK refresh (v2)

Require: Public noise epoch $Q_t \in \{0,1\}^N$; secret state $\mathsf{DFK}_t \in \{0,1\}^\lambda$; context ctx_t
Ensure: Working key $\mathsf{MEK}_t \in \{0,1\}^\ell$; refreshed state $\mathsf{DFK}_{t+1} \in \{0,1\}^\lambda$

$s_t \leftarrow \mathsf{Extract}(\mathsf{DFK}_t, Q_t, ctx_t)$
$\mathsf{MEK}_t \leftarrow \mathsf{Derive}(s_t, \text{“MEK”}, \ell)$
$\mathsf{DFK}_{t+1} \leftarrow \mathsf{Derive}(s_t, \text{“DFK”}, \lambda)$
return $(\mathsf{MEK}_t, \mathsf{DFK}_{t+1})$

6 Security Definitions

6.1 Entities, transcripts, and side-information

Let $Q_t \in \{0,1\}^N$ denote the transmitted quantum-noise epoch at time t. Let $\mathsf{DFK}_t \in \{0,1\}^\lambda$ denote the secret ratchet state at time t. Let ctx_t denote public context/domain separation data (epoch counter, direction, stream index, etc.).

We distinguish between: (i) *integrity* of epochs (handled by signatures/MACs and replay protection), and (ii) *secrecy* of derived keys (the focus of this section).

We model the adversary's view at epoch t as side-information Z_t that includes Q_t, public metadata, all ciphertexts transmitted under keys derived up to time t, and optionally any *known-plaintext leakage* (e.g., protocol headers or exposed payload fragments).

6.2 Ratchet interface and derived objects

The ratchet function produces a working key and refreshed state:

$$(\mathsf{MEK}_t, \mathsf{DFK}_{t+1}) \leftarrow \mathsf{F}(\mathsf{DFK}_t, Q_t, ctx_t).$$

In QS-OTP mode, we additionally consider a family of m independent ratchet chains indexed by i:

$$(\mathsf{MEK}_t^{(i)}, \mathsf{DFK}_{t+1}^{(i)}) \leftarrow \mathsf{F}(\mathsf{DFK}_t^{(i)}, Q_t, ctx_t^{(i)}), \quad i \in \{1, \ldots, m\}.$$

Each $ctx_t^{(i)}$ domain-separates streams (index i, direction, application label, epoch counter).

6.3 Bounded information-theoretic key indistinguishability

We use a statistical (information-theoretic) notion. The "bound" is explicit: it depends on the adversary's uncertainty about the secret state given side-information.

Definition 1 (Conditional min-entropy of the secret state)**.** *For random variables* (DFK_t, Z_t)*, define the conditional min-entropy* $H_\infty(\mathsf{DFK}_t \mid Z_t)$ *in the standard way [2]. Intuitively, it measures how well an adversary that sees* Z_t *can guess* DFK_t*.*

Definition 2 (Per-epoch bounded ITS for a derived key)**.** *Fix an epoch* t*. Let* MEK_t *be derived via* $\mathsf{F}(\mathsf{DFK}_t, Q_t, ctx_t)$*. We say* MEK_t *achieves* bounded information-theoretic secrecy *at epoch* t *(with advantage at most* ε*) if*

$$\Delta\Big((\mathsf{MEK}_t, Z_t), (U_\ell, Z_t)\Big) \leq \varepsilon,$$

where U_ℓ *is uniform over* $\{0,1\}^\ell$ *and* $\Delta(\cdot,\cdot)$ *is statistical distance. The guarantee is* bounded *because* Z_t *may grow over time (e.g., due to known-plaintext exposure), reducing* $H_\infty(\mathsf{DFK}_t \mid Z_t)$ *and eventually invalidating the inequality.*

Remark 1 (Why this is the right notion for "operational ITS")**.** *This definition makes no computational assumption and does not promise "forever" security. It is the appropriate formalization for high-assurance systems where the DFK is assumed uncompromised and key material is consumed with strict non-reuse discipline.*

Remark 2 (No entropy expansion)**.** *Although multiple values may be derived in a single step (e.g., a working key and a refreshed state), we do not claim that this creates more information-theoretic secrecy than the available secret entropy budget. In particular, any ITS guarantees are stated per derived object (marginally), while the joint secrecy of* $(\mathsf{MEK}_t, \mathsf{DFK}_{t+1})$ *remains bounded by the adversary's residual uncertainty in the pre-state.*

6.4 QS-OTP non-reuse semantics and leakage accounting

QS-OTP targets OTP-grade *non-reuse* semantics (pad length matches payload length, no reuse). However, even an OTP system has a hard constraint: if an adversary learns plaintext, they learn the corresponding pad bits. Therefore, pad exposure must be accounted for explicitly.

Definition 3 (Pad exposure budget per DFK)**.** *In QS-OTP mode, define* $E_t^{(i)}$ *as the number of pad bits from stream* i *that become known to the adversary by time* t *(through known plaintext or explicit leakage). The effective secrecy remaining in stream* i *is bounded by the residual uncertainty in* $\mathsf{DFK}_t^{(i)}$ *given* Z_t *and the exposure* $E_t^{(i)}$*.*

Remark 3 (Single-use discipline)**.** *A conservative operational rule that preserves clean bounds is:* use each $\mathsf{DFK}^{(i)}$ to generate at most one ℓ-bit pad segment before it is refreshed, and do not reuse the same (i, t, ctx) tuple. *This provides OTP-grade non-reuse semantics while keeping leakage accounting simple.*

6.5 Forward secrecy notions (optional, scoped)

We separate two properties: (i) *forward secrecy of past derived keys* under future compromise of DFK states, and (ii) *post-compromise recovery*. This paper's primary claim is bounded ITS under non-compromise; we discuss compromise properties only to the extent the ratchet update is designed to avoid leaking prior states.

7 Security Arguments (Sketches)

7.1 Core intuition: secrecy comes from the secret reservoir, not from public noise

In v2, the transmitted epoch Q_t is treated as publicly observable but integrity-protected. Therefore, information-theoretic secrecy must come from the adversary's uncertainty about DFK_t. The role of Q_t is to provide high-entropy *material to be selected/combined* under the secret state, and to ensure freshness and strong separation across epochs.

In QS-OTP mode, the secrecy bound is particularly transparent: the system begins with a reservoir of m independent 256-bit states, and each state is used to generate one pad segment under strict non-reuse.

7.2 A sufficient condition for per-epoch uniformity (keyed balanced mapping)

A simple, defensible route to information-theoretic per-epoch uniformity is to use a family of functions $\{g_s : \{0,1\}^N \to \{0,1\}^\ell\}_{s\in\{0,1\}^\lambda}$ such that for any fixed input $x \in \{0,1\}^N$, the distribution of $g_S(x)$ is uniform (or ε-close) when S is uniform. Intuitively, s acts as a secret selector that induces a uniform output even when x is known.

Definition 4 (Balanced keyed extraction (sufficient condition))**.** *A keyed mapping family g is* balanced *if for every fixed $x \in \{0,1\}^N$ and uniform $S \in \{0,1\}^\lambda$, $g_S(x)$ is exactly uniform over $\{0,1\}^\ell$ (or within ε statistical distance of uniform).*

Theorem 1 (Per-epoch bounded ITS from secret-keyed balanced extraction)**.** *Assume that for each epoch t, $\mathsf{MEK}_t = g_{\mathsf{DFK}_t}(Q_t, ctx_t)$ for a balanced family g (with domain separation encoded in ctx_t), and that the adversary's side-information Z_t does not enable guessing DFK_t better than probability $2^{-\lambda'}$, i.e., $H_\infty(\mathsf{DFK}_t \mid Z_t) \geq \lambda'$. Then*

$$\Delta\Big((\mathsf{MEK}_t, Z_t), (U_\ell, Z_t)\Big) \leq 2^{-\lambda'} + \varepsilon,$$

where ε captures any non-ideal balance of the family.

Proof sketch. Condition on Z_t. The adversary's view induces a posterior distribution over DFK_t. If DFK_t is still close to unpredictable (high conditional min-entropy), the mixture over $g_{\mathsf{DFK}_t}(Q_t, ctx_t)$ remains close to the mixture induced by a uniform secret key, which is uniform by balance. The residual advantage is bounded by the probability mass of the best guess of DFK_t plus any imbalance term. □

7.3 Concrete construction: secret-index path extraction from a noise epoch

We instantiate the keyed extraction family g using *secret-index pathing* over the transmitted epoch Q_t.

Epoch structure. Let $Q_t \in \{0,1\}^N$ be treated as an indexed bit array $Q_t[0], \ldots, Q_t[N-1]$. Let ℓ be the desired output length (e.g., $\ell = 256$).

DFK structure. Let $\mathsf{DFK}_t \in \{0,1\}^\lambda$ be interpreted as a secret seed that deterministically specifies: (i) a sequence of ℓ indices $I_t = (i_1, \ldots, i_\ell)$ with $i_j \in \{0, \ldots, N-1\}$, (ii) an ℓ-bit one-time mask $M_t \in \{0,1\}^\ell$, and (optionally) (iii) a short *update selector* for state refresh. All of these are derived from (DFK_t, ctx_t) using a deterministic parsing rule.

Extraction rule (key output). Define

$$\mathsf{MEK}_t = g_{\mathsf{DFK}_t}(Q_t, ctx_t) := \Big(Q_t[i_1] \,\|\, Q_t[i_2] \,\|\, \cdots \,\|\, Q_t[i_\ell]\Big) \;\oplus\; M_t, \tag{4}$$

where $\|$ denotes concatenation.

State refresh rule. Refresh the state by selecting a disjoint (or provably non-overlapping) index set $\tilde{I}_t$ of size λ and a mask $\tilde{M}_t \in \{0,1\}^\lambda$ from (DFK_t, ctx_t), and define:

$$\mathsf{DFK}_{t+1} := \Big(Q_t[\tilde{i}_1] \,\|\, \cdots \,\|\, Q_t[\tilde{i}_\lambda]\Big) \;\oplus\; \tilde{M}_t, \tag{5}$$

with the additional operational requirement that $(I_t, \tilde{I}_t)$ are non-overlapping and never reused for the same stream index and direction.

Why this family is balanced (uniformity from secret masks). For any *fixed* epoch Q_t and any fixed index sequence I_t, the extracted raw path bits $X_t := Q_t[i_1] \| \cdots \| Q_t[i_\ell]$ are a *fixed* ℓ-bit string from the adversary's perspective (if I_t is known). However, when M_t is unknown and uniform over $\{0,1\}^\ell$, $X_t \oplus M_t$ is exactly uniform over $\{0,1\}^\ell$. Thus, the mapping family g is balanced in the sense of Section 7.2 whenever the induced mask component M_t retains near-uniformity given the adversary view.

Remark 4 (What carries the entropy)**.** *In this construction,* information-theoretic secrecy comes from the hidden mask bits and/or hidden index selection contained in the secret state. *The public epoch Q_t provides a large substrate from which fresh state can be re-parameterized, but it does not itself provide secret entropy if fully observable by the adversary.*

7.4 DFK format for secret-index pathing (QS-OTP-ready)

We specify a concrete DFK structure that supports: (i) secret-index path extraction for MEK, (ii) state refresh from the same epoch Q_t without overlap, and (iii) QS-OTP reservoir parallelism.

State parameters. Let ℓ denote the pad/key segment length per use (e.g., $\ell = 256$ bits). Let λ denote the DFK state size (we take $\lambda = \ell$ for QS-OTP, i.e., a 256-bit state per stream). Let N be the epoch size in bits.

DFK as a tuple. For stream index i, define the ratchet state at epoch t as:

$$\mathsf{DFK}_t^{(i)} := \big(M_t^{(i)},\ S_t^{(i)},\ c_t^{(i)}\big),$$

where:

- $M_t^{(i)} \in \{0,1\}^\ell$ is a one-time mask used to balance the extracted path bits into a uniform pad segment.
- $S_t^{(i)} \in \{0,1\}^\sigma$ is a compact *index-schedule seed* used to deterministically generate non-overlapping index sets for MEK extraction and DFK refresh (defined below). σ may be small (e.g., 128–256 bits) and is treated as secret.
- $c_t^{(i)} \in \mathbb{N}$ is a monotonic local counter (or epoch pointer) included in $ctx_t^{(i)}$ to enforce uniqueness.

Index scheduling. Given $(S_t^{(i)}, ctx_t^{(i)})$, the endpoint deterministically produces two disjoint index sets:

$$I_t^{(i)} = (i_1, \ldots, i_\ell), \qquad \tilde{I}_t^{(i)} = (\tilde{i}_1, \ldots, \tilde{i}_\lambda),$$

with the constraint $I_t^{(i)} \cap \tilde{I}_t^{(i)} = \emptyset$ and global non-reuse of any index set for the same stream and direction. (Implementations may enforce this by deriving indices from a permutation of $\{0, \ldots, N-1\}$ keyed by $S_t^{(i)}$, then slicing disjoint segments for $I_t^{(i)}$ and $\tilde{I}_t^{(i)}$.)

MEK/pad derivation (per stream). Define:

$$X_t^{(i)} := Q_t[i_1] \| \cdots \| Q_t[i_\ell], \qquad \mathsf{MEK}_t^{(i)} := X_t^{(i)} \oplus M_t^{(i)}.$$

State refresh (per stream). Define:

$$Y_t^{(i)} := Q_t[\tilde{i}_1] \| \cdots \| Q_t[\tilde{i}_\lambda], \qquad M_{t+1}^{(i)} := Y_t^{(i)} \oplus \tilde{M}_t^{(i)},$$

where $\tilde{M}_t^{(i)}$ is a fresh mask derived deterministically from $(M_t^{(i)}, S_t^{(i)}, ctx_t^{(i)})$ and is treated as secret as long as the state remains secret.[1]

Finally update the schedule seed to ensure fresh, non-overlapping index generation:

$$S_{t+1}^{(i)} := \mathsf{Upd}(S_t^{(i)}, M_t^{(i)}, Y_t^{(i)}, ctx_t^{(i)}), \qquad c_{t+1}^{(i)} := c_t^{(i)} + 1.$$

The update function Upd is deterministic and designed to preserve secrecy of S under non-compromise. (We do not require Upd to be a PRF; rather, we require that the state is never revealed and that indices are never reused.)

QS-OTP reservoir initialization. QS-OTP initializes m independent streams:

$$\mathsf{DFK}_0^{(1)}, \ldots, \mathsf{DFK}_0^{(m)},$$

with independent masks $M_0^{(i)}$ and schedule seeds $S_0^{(i)}$ loaded via manual initialization/key ceremony. This makes the secrecy budget explicit and avoids relying on "entropy creation" from public epochs.

7.5 A tightened per-epoch ITS statement for secret-index path extraction

We now restate the sufficient condition of Theorem 1 for the concrete construction above.

Theorem 2 (Per-epoch bounded ITS for secret-index path extraction)**.** *Fix an epoch t. Suppose MEK_t is produced by Eq. (4). Let Z_t be the adversary side-information. If the induced one-time mask M_t satisfies*

$$\Delta\big((M_t, Z_t), (U_\ell, Z_t)\big) \le \varepsilon_M,$$

then

$$\Delta\Big((\mathsf{MEK}_t, Z_t), (U_\ell, Z_t)\Big) \le \varepsilon_M.$$

In particular, MEK_t is statistically indistinguishable from uniform conditioned on Z_t up to ε_M.

Proof sketch. Condition on any fixed $Z_t = z$. Eq. (4) expresses MEK_t as $X_t \oplus M_t$, where X_t is fixed given Q_t and the (deterministically derived) index sequence. XOR with an ε_M-close-to-uniform mask yields an ε_M-close-to-uniform result. □

[1] The role of $\tilde{M}_t^{(i)}$ is to prevent the refreshed mask from becoming a known function of Q_t alone. It can be implemented as a simple re-masking rule inside the state update, provided the masking material is not leaked.

7.6 QS-OTP specialization (m independent DFKs)

In QS-OTP mode, initialization provisions a reservoir of m independent states $\mathsf{DFK}_0^{(1)}, \ldots, \mathsf{DFK}_0^{(m)}$, each inducing its own mask $M_t^{(i)}$ and index sequence $I_t^{(i)}$. Provided each stream enforces strict non-reuse of (i, t, ctx) and avoids overlap of index sets used for $\mathsf{MEK}_t^{(i)}$ versus $\mathsf{DFK}_{t+1}^{(i)}$, Theorem 2 applies independently per stream. The overall security bound is therefore the maximum (or union) of the per-stream bounds, and the total available one-time pad material is explicitly bounded by the reservoir size and the permitted exposure (Section 6.4).

7.7 Why the claim is bounded (and how QS-OTP keeps it clean)

Theorem 1 makes the bound explicit: if Z_t grows so that $H_\infty(\mathsf{DFK}_t \mid Z_t)$ collapses, the guarantee collapses. In practice, the most important contributor to such growth is known-plaintext exposure under pad-derived encryption, which reveals pad bits.

QS-OTP keeps this clean by: (i) using a *reservoir* of independent $\mathsf{DFK}^{(i)}$ values, and (ii) enforcing *strict non-reuse* so that each derived pad segment is used once and then the state advances.

Operationally, the reservoir approach prevents a single state chain from being burdened by large cumulative exposure. Instead, exposure is spread across independent chains, preserving clean bounds per chain.

7.8 A boundedness theorem (pad exposure degrades the ITS guarantee)

In any OTP-style system, known plaintext reveals pad bits. QS-OTP therefore tracks exposure explicitly (Section 6.4). We now formalize how exposure reduces the remaining information-theoretic guarantee.

Definition 5 (Exposure model per stream)**.** *For QS-OTP stream i, let $E_t^{(i)}$ denote the number of pad bits from $\mathsf{MEK}_{<t}^{(i)}$ that become known to the adversary by time t (e.g., through known plaintext, protocol headers, or partial plaintext disclosure). Let Z_t denote all other side-information (epochs, metadata, ciphertexts).*

Theorem 3 (Residual secrecy bound from conditional min-entropy)**.** *Fix a QS-OTP stream i at epoch t. Suppose $\mathsf{MEK}_t^{(i)}$ is derived by the secret-index path construction (Eq. (4)) with a one-time mask $M_t^{(i)} \in \{0,1\}^\ell$. Then the best possible information-theoretic distinguishing advantage at epoch t satisfies:*

$$\Delta\Big((\mathsf{MEK}_t^{(i)}, Z_t), (U_\ell, Z_t)\Big) \;\le\; 2^{-H_\infty(M_t^{(i)} \mid Z_t)}.$$

Moreover, if exposure up to time t reveals $E_t^{(i)}$ independent bits of the mask (or equivalently constrains the state by $E_t^{(i)}$ bits), then

$$H_\infty\big(M_t^{(i)} \mid Z_t\big) \;\ge\; \ell - E_t^{(i)},$$

and hence

$$\Delta\Big((\mathsf{MEK}_t^{(i)}, Z_t), (U_\ell, Z_t)\Big) \;\le\; 2^{-(\ell - E_t^{(i)})}.$$

Proof sketch. The first inequality is a standard bound: if an adversary can guess a secret with probability p, its distinguishing advantage against a masked value is at most p. For the second inequality, each exposed pad bit reveals (at most) one bit of information about the mask/state, reducing the conditional min-entropy by at most one per independent bit of exposure. Substituting yields the stated bound. □

Remark 5 (Operational interpretation)**.** *For $\ell = 256$, if a stream experiences $E_t^{(i)} = 0$ pad-bit exposure prior to using $\mathsf{MEK}_t^{(i)}$, the statistical advantage bound is 2^{-256}. If $E_t^{(i)} = 64$ bits are effectively exposed, the bound becomes 2^{-192}. Thus, QS-OTP can remain extremely strong under realistic partial-known-plaintext models, but the accounting must be explicit.*

Remark 6 (Why the reservoir matters)**.** *Because QS-OTP uses m independent streams, exposure in one stream does not directly reduce the min-entropy of other streams. This is precisely why preloading many independent DFKs supports a clean bounded ITS statement at scale.*

7.9 Ratcheting: preserving secrecy of refreshed states (non-expansion)

The ratchet update $\mathsf{DFK}_{t+1} \leftarrow \mathsf{F}_{\text{state}}(\mathsf{DFK}_t, Q_t, ctx_t)$ must be designed so that: (i) DFK_{t+1} does not leak DFK_t to an observer of public transcripts, and (ii) domain separation prevents correlations between outputs (e.g., MEK_t vs. DFK_{t+1}).

Crucially, the update must be viewed as *state evolution*, not as entropy creation: when Q_t is public, iterating the ratchet on the same epoch cannot increase secret entropy. It can, however, preserve unpredictability of the current state under appropriate design and non-leakage.

Remark 7 (No "entropy from nowhere")**.** *Even if a noise epoch is extremely large, if it is fully known to the adversary then it contributes no secret entropy. Bounded ITS claims therefore rest on the (remaining) unpredictability of the secret state and the strict accounting of any leakage that reveals information about derived pads/keys.*

7.10 QS-OTP bandwidth efficiency does not weaken the ITS statement

QS-OTP's low bandwidth overhead follows from amortization: a single epoch Q_t supports m independent pad derivations in parallel. The secrecy statement remains per-stream and bounded by each stream's state secrecy and exposure accounting; bandwidth efficiency does not change the information-theoretic nature of the bound.

7.11 What a full formal proof would include (roadmap)

A complete proof would specify: (i) a concrete balanced keyed extraction family g suitable for implementation, (ii) the exact leakage model Z_t (including which plaintext fields are assumed known), (iii) domain separation requirements, and (iv) an explicit theorem relating cumulative pad exposure to the decay of $H_\infty(\mathsf{DFK}_t \mid Z_t)$ per stream in QS-OTP mode.

7.12 Security discussion: separating secrecy, integrity, and traffic analysis

Secrecy (bounded ITS) is about statistical key indistinguishability. The core q-stream v2 claim concerns secrecy of derived keys (or pads) against a passive eavesdropper given the public epochs $\{Q_t\}$ and public metadata. The formal statements in Section 6 are information-theoretic and *bounded*: they hold as long as the secret state remains uncompromised and as long as any pad exposure (e.g., known plaintext) is accounted for under the residual min-entropy bounds (Theorem 3).

Integrity and anti-replay are separate and mandatory. Like any stream/OTP-style encryption, q-stream requires robust integrity protection and replay prevention. Absent authenticity, an active adversary can substitute or replay epochs, inducing keystream reuse or forcing state desynchronization. These are not secrecy failures of the entropy model but protocol failures. Therefore, each epoch must be authenticated (signature/MAC) and bound to a monotonic counter and direction, and endpoints must reject replayed or out-of-order epochs [7, 6].

Traffic analysis is out-of-scope for OTP secrecy. OTP-grade confidentiality does not hide message timing, sizes, endpoints, or volume. q-stream v2 does not claim traffic-flow confidentiality by default. If required, standard defenses (padding, batching, cover traffic, mix networks, link encryption across controlled segments) can be layered independently of the q-stream key schedule.

Endpoint compromise and key disclosure. If an endpoint is compromised such that DFK state is extracted, bounded ITS claims no longer apply from that point onward. Forward secrecy of past traffic depends on whether past pad material and prior states were securely erased, and on whether the state update prevents recovery of prior state from current state. Post-compromise recovery requires injection of new secret entropy after compromise (outside the scope of QS-OTP unless additional private entropy sources are available).

Operational summary. q-stream v2 should be viewed as a high-assurance *key lifecycle mechanism*: manual (or ceremony-based) initialization establishes an explicit secret reservoir; thereafter, authenticated public epochs drive automatic key rotation with bounded information-theoretic secrecy, while integrity, replay protection, and traffic-analysis defenses are handled by standard, separable controls.

7.13 Non-reuse rules (QS-OTP compliance checklist)

QS-OTP requires:

1. Never reuse any $(i, ctx, \text{direction})$ tuple.
2. Never reuse an epoch identifier with the same stream index and direction.
3. Reject replayed or out-of-order epochs.
4. Use each pad segment $\mathsf{MEK}_t^{(i)}$ at most once; securely erase after use.
5. Ensure index sets for pad extraction and state refresh are disjoint per invocation.

8 QS-OTP: Operational One-Time-Pad Mode via a Preloaded DFK Reservoir

8.1 Motivation

OTP remains attractive in defense and critical infrastructure due to its stability and strong secrecy guarantees [5], but is historically constrained by distribution and storage of bulk key material. QS-OTP is intended for environments that already accept manual initialization, but want scalable automatic re-keying with OTP-grade non-reuse semantics and bounded ITS guarantees.

8.2 Operational model

At initialization, both endpoints are loaded with a reservoir of m independent DFK values:

$$\{\mathsf{DFK}_0^{(1)}, \mathsf{DFK}_0^{(2)}, \ldots, \mathsf{DFK}_0^{(m)}\},$$

e.g., $m = 1024$, with $\lambda = 256$-bit states per entry. At each epoch t, the distributor broadcasts a noise epoch Q_t. Each endpoint locally derives m independent pads (or pad segments) of length ℓ:

$$(\mathsf{MEK}_t^{(i)}, \mathsf{DFK}_{t+1}^{(i)}) \leftarrow \mathsf{F}(\mathsf{DFK}_t^{(i)}, Q_t, ctx_t^{(i)}), \quad i = 1, \ldots, m,$$

where $ctx_t^{(i)}$ domain-separates streams (index i, direction, application label, epoch counter).

8.3 OTP-grade non-reuse semantics

QS-OTP enforces:

- Each pad segment $\mathsf{MEK}_t^{(i)}$ is used at most once.
- No (i, t, ctx) tuple is reused.
- The reservoir index i ensures parallelizable keying without repeating a single state chain.

Encryption is XOR with an integrity tag (e.g., HMAC) computed under a separately derived integrity key.

8.4 Bandwidth and overhead

For a single epoch, the total locally derived OTP pad material is:

$$B_{\text{pad}} = m \cdot \ell \quad \text{bits.}$$

If the only transmission overhead unique to QS-OTP is the noise epoch Q_t of length $|Q_t| = N$ bits (plus small authentication metadata), the amortized overhead ratio is:

$$\rho = \frac{N}{m \cdot \ell}.$$

Worked example (m=1024, $\ell = 256$ bits). Then $B_{\text{pad}} = 1024 \cdot 256 = 262{,}144$ bits $= 32$ KiB of OTP pad per epoch. If $N = 4096$ bits, then $\rho = 4096/262{,}144 \approx 1.56\%$ overhead. (If instead one used the earlier v1 illustrative block size $N = 2{,}097{,}152$ bits, cited in the v1 paper, then $\rho = 8$, i.e., noise dominates payload; v2 is not required to retain that large N.) [4]

8.5 QS-OTP bandwidth: overhead vs. epoch size

In QS-OTP, a single noise epoch Q_t supports m parallel one-time pad segments of length ℓ. The total pad material per epoch is:

$$B_{\text{pad}} = m \cdot \ell.$$

For $m = 1024$ and $\ell = 256$, this yields:

$$B_{\text{pad}} = 1024 \cdot 256 = 262{,}144 \text{ bits} = 32 \text{ KiB}.$$

If the per-epoch overhead is dominated by the transmitted noise epoch of size $N = |Q_t|$ bits, the amortized overhead ratio is:

$$\rho = \frac{N}{m \cdot \ell}.$$

Table 1: QS-OTP amortized bandwidth overhead for $m = 1024$ streams and $\ell = 256$-bit pad segments (32 KiB pad per epoch)

Epoch size N (bits)	Epoch size (bytes)	Overhead ratio ρ	Overhead (%)
4,096	512	0.015625	1.5625%
16,384	2,048	0.0625	6.25%
65,536	8,192	0.25	25%
262,144	32,768	1.0	100%
1,048,576	131,072	4.0	400%
2,097,152	262,144	8.0	800%

The table highlights a practical design knob: when QS-OTP uses a preloaded DFK reservoir to derive pad material in parallel, the noise epoch can be kept compact while still amortizing overhead to low single-digit percentages.

8.6 Bounded ITS statement for QS-OTP

In QS-OTP, the security bound is explicit: the system begins with m independent secret states; each is consumed/advanced by the ratchet per epoch. Thus the total available statistically secure pad material is bounded by the provisioned secret entropy and the per-epoch extraction design (and any additional secret entropy injection, if present).

9 Comparison: v1 vs v2

9.1 Conceptual shift

v1: "embed recipient-specific markers and kGen pointers into a QRNG block; recipients search and reconstruct keys" [4].
v2: "transmit only noise; recipients derive and refresh keys via a secret ratchet state."

9.2 Claim shift

v1 emphasizes combinatorial placement hardness and marker rotation [4].
v2 emphasizes bounded information-theoretic statements under explicit entropy/secrecy assumptions.

10 Implementation Considerations

10.1 Epoch authentication and replay protection

Epochs should be signed or MACed and carry a monotonic counter. Replay protection is required to avoid keystream reuse.

10.2 Domain separation

All derived material must be domain-separated [3]: labels for MEK vs DFK refresh; stream indices in QS-OTP; direction (TX/RX); application binding.

10.3 Parameter selection guidance (choosing N, m, and ℓ)

QS-OTP exposes three primary knobs: epoch size $N = |Q_t|$ (bits), reservoir size m (number of independent DFK streams), and pad segment length ℓ (bits). A practical selection balances bandwidth overhead, operational cadence, and exposure accounting.

1) Start from the required protected throughput. Per epoch, QS-OTP yields:

$$B_{\text{pad}} = m \cdot \ell \quad \text{bits of one-time pad material.}$$

If the system requires protected throughput R (bits/s) and uses epochs at frequency f (epochs/s), then choose (m, ℓ) such that:

$$m \cdot \ell \;\geq\; \frac{R}{f}.$$

Equivalently, for a given (m, ℓ), the maximum sustained protected rate is:

$$R_{\max} = f \cdot m \cdot \ell.$$

2) Choose N to meet a target overhead ratio. The amortized epoch overhead (ignoring small authentication metadata) is:

$$\rho = \frac{N}{m \cdot \ell}.$$

A simple design rule is:

$$N \approx \rho_{\text{target}} \cdot m \cdot \ell.$$

For example, with $(m, \ell) = (1024, 256)$ (32 KiB pad per epoch), targeting $\rho_{\text{target}} \approx 1\%$ suggests $N \approx 0.01 \cdot 262{,}144 \approx 2{,}621$ bits (so a convenient choice such as 4096 bits yields $\sim 1.56\%$).

3) Select ℓ to simplify key lifecycle and exposure accounting. Smaller ℓ yields finer-grained pad management and reduces worst-case loss if a segment is mishandled, but increases per-message overhead in tagging/metadata and reservoir scheduling complexity. Larger ℓ reduces per-message management overhead and matches common cryptographic key sizes (e.g., 256 bits), but requires strict operational discipline because known plaintext reveals the corresponding pad bits. A conservative default in high-assurance environments is $\ell = 256$ with one segment per stream per epoch.

4) Select m (reservoir width) to match parallelism and mission duration. Increasing m increases pad per epoch linearly without increasing epoch size N, thus reducing bandwidth overhead ρ. Operationally, m also provides natural parallelism: different traffic classes, directions, or channels can be assigned distinct stream indices to reduce coordination and to isolate exposure budgets. The reservoir size is also an explicit finite "secrecy budget" under the bounded ITS model: larger m supports more independent one-time usage before rekeying ceremonies or replenishment.

5) Epoch cadence f and replay controls. Epochs should be authenticated and carry a monotonic identifier. Higher cadence (larger f) reduces the amount of pad material tied to any single epoch and can limit rollback/replay windows, but increases control-plane overhead. Lower cadence reduces control-plane overhead but increases the operational impact of any epoch loss (requiring buffering).

Profile guidance (illustrative). The following qualitative profiles summarize typical selections:

- **Tactical / low-bandwidth links (HF/VHF/UHF, contested SATCOM):** Choose compact epochs (e.g., $N \in [4096, 16384]$ bits) to minimize overhead. Prefer moderate m and $\ell = 256$ to simplify stream management. Use lower cadence with robust buffering and strong replay protection.
- **Mobile / intermittent connectivity (field networks, forward operating bases):** Prefer moderate N and increased m so pad material per epoch remains sufficient even under bursty delivery. Consider assigning stream indices by traffic class to isolate exposure budgets.
- **Core backbone / fixed infrastructure (base-to-base, protected fiber segments):** Bandwidth is less constrained; choose higher cadence for operational robustness and smaller per-epoch operational blast radius. Increase m to support multi-channel high throughput while keeping ρ low.

Engineering note. Authentication metadata (signatures/MACs, counters, and optional padding) adds a small fixed overhead per epoch and should be included in ρ for precise link budgeting. However, ρ remains dominated by N and the derived pad volume $m\ell$, and can therefore be engineered to low single-digit percentages with appropriate parameter choices.

10.4 Key hygiene

Secure storage of DFK (preferably HSM/TEE where available), deletion of used pad segments, and audit of non-reuse.

11 Use-Cases and Operational Deployment

- Defense and tactical comms that already accept initial key loading but require frequent re-keying.
- Critical infrastructure interconnects with long confidentiality horizons.
- Sovereign or alliance deployments with shared initialization procedures and strict key lifecycle controls.
- QS-OTP is intended for high-assurance confidentiality against passive capture where payload plaintext is not assumed fully known to the adversary; integrity and traffic-flow security are separate concerns."

11.1 Operational relevance to defense and high-assurance environments

QS-OTP targets a specific operational gap: environments that already accept manual key initialization (e.g., key ceremony, couriered material, KMI distribution, HSM/TEE injection), but do not have a scalable mechanism for frequent, automatic re-keying with information-theoretic assurances without distributing bulk one-time pad files. By preloading a finite reservoir of independent DFK states (e.g., $m = 1024$) and enforcing strict non-reuse, QS-OTP provides OTP-grade usage semantics—pad length matches payload length, and no pad segment is reused—while shifting the ongoing distribution burden to authenticated transmission of compact quantum-noise epochs. The information-theoretic guarantee is explicit and bounded by the provisioned secret reservoir and the system's leakage/exposure model, which aligns with how defense COMSEC systems already reason about key lifecycle, compromise, and accountability. In operational terms, QS-OTP can deliver high-rate, one-time keystream encryption with minimal bandwidth overhead (Table 1) and without the logistics of distributing large secret key files, while keeping integrity, replay protection, and traffic-flow defenses separable and enforceable as standard controls.

12 Conclusion

We presented q-stream v2 as a pure-noise transmission key-rotation mechanism that enables bounded ITS statements under explicit assumptions, and introduced QS-OTP mode using a preloaded DFK reservoir to achieve OTP-grade non-reuse semantics with minimal bandwidth overhead.

References

[1] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[2] Yevgeniy Dodis. On extractors, error-correction and hiding all partial information, 2008. Survey note (PDF).

[3] Hugo Krawczyk and Pasi Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869, May 2010.

[4] Adrian Neal. The quantum-safe key-distribution mechanism having non-conjectured hardness, while scalable for a vernam cipher, under shannon conditions. In Kohei Arai, editor, *Proceedings of the Future Technologies Conference (FTC) 2025, Volume 2*, pages 264–281, Cham, 2026. Springer Nature Switzerland.

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

[6] Douglas R. Stinson. Universal hashing and authentication codes. In *Advances in Cryptology — CRYPTO '91*. Springer, 1991. Also available via SpringerLink.

[7] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981.