
WHAT CAN VERIFIABLE DECAPSULATION TESTS CERTIFY?

PASS BOUNDS AND FAULT-RECOGNITION LIMITS

FOR FO-BASED KEMs

José Luis Delgado
Universitat Oberta de Catalunya
jdelgado13@uoc.edu

Abstract

Black-box tests for Fujisaki–Okamoto decapsulation observe the sampled execution seen by the harness, whereas the reencryption computation itself is visible only through the values that reach final key derivation. We study confirmation-code-augmented KEM variants under an honest-reference harness in which the reference encapsulation fixes a hidden final-key point $\langle \text{good}, B, W \rangle$, with W the confirmation witness. For a q -localized system under test, acceptance is bounded by honest correctness error, adversarial aliasing, final-key freshness defects, a hit on the localized suffix list $Q_G(B)$, and $2^{-\kappa}$. A one-query construction from any predictor of W matches this bound up to the fresh-key coincidence term, so the list-hit event is the black-box obstruction measured by the harness.

The list-hit term is bounded either by a cUP-faithful harness certificate, which transfers source confirmation-code unpredictability with a q -loss, or by an average conditional min-entropy bound, with separate RawEnt and TailEnt hypotheses for short diagnostic and truncation-tail codes. The same model proves a dependency-cone lower bound for non-certification claims. When the black-box observation of an honest-support harness factors through the confirmation-observable final-key target, every operation outside the support-active cone has a coupled erasure implementation with the same transcript distribution; over any implementation class containing that erasure, soundness and completeness errors of an execution certifier satisfy $\alpha + \beta \geq 1$. The ML-KEM and HQC case studies distinguish theorem-covered positive rows, finite-catalog artifact rows, and non-certification rows that carry a cone-inactivity certificate. The security of the standard KEM lines is the construction-level security supplied by the cited source analyses.

1 Introduction

Post-quantum KEMs built from the Fujisaki–Okamoto transform rely on a decapsulation check that is essential for security. A decapsulation algorithm decrypts the received ciphertext, rederives the encryption randomness, recomputes the ciphertext, compares the result to the received ciphertext, and then derives the final shared secret. The check is security-critical, but an ordinary input-output test observes only the sampled path. On honestly generated ciphertexts, an implementation that omits part of the check may still return the expected shared secret.

Glabush, Günther, Hövelmanns, and Stebila introduced verifiable decapsulation for FO-based KEMs [13]. Their construction adapts the confirmation-code methodology of Fischlin and Günther [7]: encryption produces an internal confirmation value, decapsulation recomputes it, and the value is tied to the subsequent key material. In the KEM setting, this makes the relevant reencryption computation part of the ordinary shared-key outcome: a fault that misses the computation produces the wrong final input except on the usual guessing events.

After the final key derivation includes a confirmation value, the pass probability measured by a test depends on the augmented KEM variant, the fault class, the peer model, and the sampled input

distribution. Removing the confirmation value from the final key input, changing branch selection on malformed ciphertexts, and applying the same endpoint mutation on both sides give different harness-indexed statements. For a fixed KEM variant, fault class, and harness, our theorem bounds the corresponding pass probability through a localized final-key query list. The source notions are the cUP and fCOR games of verifiable decapsulation; the harness layer reformulates them as a list-cUP testing statement with an explicit SUT view and support.

Main results. The central result is a formal theory of confirmation-code testing as list-cUP evidence under harness-indexed views. A canonical honest-reference instance fixes a hidden final-key point

$$x^* = \langle \text{good}, B, W \rangle$$

and a reference key $K = G_K(x^*)$, where B is the localized transcript prefix and W is the hidden confirmation witness. A tested decapsulator may be randomized, adaptive, stateful, and may know the public data and the decapsulation input prescribed by the harness. Its localized final-key queries define a list

$$Q_G(B) = \{w : \mathcal{S} \text{ queried } G_K(\langle \text{good}, B, w \rangle)\}.$$

If adversarial aliases and final-key freshness defects are accounted for, then

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}} + \epsilon_{\text{fresh}} + \Pr[W \in Q_G(B)] + 2^{-\kappa}.$$

Thus every meaningful positive testing claim reduces to bounding the list-hit term. The reduction achieves black-box tightness in the sense that, for every single-candidate predictor of the hidden witness from the SUT view, there exists a (1)-localized SUT whose honest-reference pass probability is exactly the predictors success probability plus the fresh-key coincidence term. The rest of the paper supplies the ways to bound this unavoidable term: source cUP under a cUP-faithful harness certificate, conditional entropy, and diagnostic RawEnt/TailEnt assumptions for short or tail-derived codes.

To bound the list-hit term, the source-security route defines a harness list-cUP game and proves that a q -candidate list adversary loses a factor q relative to a single-candidate cUP bound. Under a source bound ϵ_{cUP} , this yields

$$\text{Pass}^{\text{href}} \leq \delta_{\Pi^C} + \epsilon_{\text{alias}} + \epsilon_{\text{fresh}} + q\epsilon_{\text{cUP}} + 2^{-\kappa}.$$

The entropic route proves that if W has average conditional min-entropy λ given a view that determines the candidate list, then

$$\text{Pass}^{\text{href}} \leq \delta_{\Pi^C} + \epsilon_{\text{alias}} + \epsilon_{\text{fresh}} + q2^{-\lambda} + 2^{-\kappa},$$

with an explicit smoothing loss when only smooth min-entropy holds.

Because the sampled peer and support are part of the experiment, the harness fixes the interpretation of these bounds. Honest-self tests can accept a self-consistent endpoint that erases the code from both encapsulation and decapsulation, even though the same endpoint fails against a sound reference peer except with correctness error and final-key guessing. Support-equivalent endpoints have the same pass probability only on the tested support. A retained range of L bits is only a capacity bound and implies no lower bound on conditional min-entropy without a separate unpredictability claim. The testing theorems in this paper are classical ROM statements; QROM security claims are inherited only from the cited construction-level verifiable-decapsulation and FO analyses.

Contributions.

1. We define honest-reference and honest-self testing experiments for confirmation-code-augmented FO decapsulation, including canonical final-key inputs, hidden reference evaluations, localized final-key query lists, adversarial aliasing, final-key freshness, and correctness bad events.

2. We prove a tight list-hit detectability theorem. Every q -localized SUT satisfies

$$\text{Pass}^{\text{href}} \leq \delta + \epsilon_{\text{alias}} + \epsilon_{\text{fresh}} + \Pr[W \in Q_G(B)] + 2^{-\kappa}.$$

Moreover, the list-hit term is unavoidable: any single-candidate predictor of W from the SUT view induces a one-query SUT whose pass probability is the predictor success probability plus fresh-key coincidence.

3. We prove the dependency-cone lower bound for black-box honest-reference certification. If the observation of an honest-support harness factors through the confirmation-observable final-key point, then no operation outside the support-active dependency cone of that point can be certified by a black-box test over an implementation class containing the coupled erasure. Equivalently, over erasure-closed classes, every such operation admits an erasure fault with identical transcript distribution, so nontrivial execution certification is information-theoretically impossible.
4. We prove the two positive routes that meet this lower bound. The source-security route defines a harness list-cUP game with a q -loss from ordinary cUP under a cUP-faithful harness certificate. The entropic route derives a $q2^{-\lambda}$ list-hit bound from average conditional min-entropy of the witness, including smoothing.
5. We prove a diagnostic-code theorem for truncated hashes of hidden raw witnesses. This theorem covers short artifact codes such as ML-KEM-CD1; those variants use the source cUP bound only when a source-equivalence certificate is supplied.
6. We map the dependency cone to ML-KEM and the August 2025 salted-SFO HQC line. For the source ML-KEM code with $|S| = 2$, only $2(k + 1)$ selected uncompressed coefficients enter the source-code cone; the remaining $254(k + 1)$ coefficients are outside that code cone unless exposed by a separate harness target. For HQC-CD1, the retained tail cone contains 5, 8, 8 bits for HQC-1/3/5, leaving 0, 3, 29 tail bits outside the CD1 cone; CD* retains the full 5, 11, 37-bit tail range.
7. We provide a reproducible artifact with pinned baselines, derived variants, harnesses, mutant catalogs, frozen JSONL records, three compact summaries, and two plots. The experiments are interpreted as witnesses for theorem-relevant mutant behavior, with construction-level security for the standard KEM lines supplied by the cited source analyses. The dependency-cone theorem upgrades a negative row only when it is paired with a cone-inactivity certificate for the full confirmation-observable target. Decision-only, component-selective, non-retained-tail, and unselected-coefficient rows are reported either as finite-catalog observations, as source-code-cone boundaries, or as theorem-covered non-certification results according to the certificate available for the stated honest-support harness.

Instantiations and artifact. The model is instantiated on two augmented families: for ML-KEM, we use the confirmation-code line of the 2025 verifiable-decapsulation paper and evaluate an artifact diagnostic variant over the FIPS 203 algorithms; for the August 2025 public HQC specification [8], the salted flow yields a direct transcript

$$H(ek_{\text{KEM}}) || m || \text{salt}$$

and our augmented variants take their confirmation code from the truncation tail. We study CD1, which retains one byte when available, and CD*, which retains the full truncation-tail range budget.

The accompanying artifact, FO-FaultBench, available at <https://github.com/hypergalois/FO-FaultBench>, pins the mlkem-native and August 2025 HQC baselines, materializes the transformed variants, and runs the harnesses used by the theorem statements. The experiments check the stated harness behavior on concrete code: binding faults are detected under $D_{\text{honest_ref}}$, decision-only faults are support-dependent, symmetric endpoint faults separate $D_{\text{honest_self}}$ from $D_{\text{honest_ref}}$, and probabilistic code guessing follows the retained-code reference scale for the tested mutants.

2 Related Work

Glabush, Günther, Hövelmanns, and Stebila introduced verifiable decapsulation for FO-based KEMs [13]. Their paper adapts confirmation codes to FO-based KEMs and proves that, for the targeted faulty-but-benign implementations, missing reencryption can appear as a correctness failure. The construction builds on the verifiable-verification methodology of Fischlin and Günther [7], where a verification computation produces disposable confirmation information and ties that information to later protocol functionality. We keep the same construction and analyze the pass probability measured by a specified harness.

A concrete test harness may allow several final-key queries, run a self-consistent mutant as the peer, sample a restricted support, or use diagnostic codes shorter than the source confirmation code. The harness list-cUP, conditional min-entropy, support separation, and diagnostic-code theorems below formalize these differences as certification signals measured by that harness.

The FO-transform background used here is the modular analysis of Hofheinz, Hövelmanns, and Kiltz [18], together with later work on prefix hashing and public-key binding, which explains why FO inputs often include public-key-derived material in multi-user settings [6]. Work on decryption failures, explicit rejection, and failing plaintexts clarifies the distinction between the reencryption computation and the accept-or-reject behavior of the decapsulation oracle [14, 16, 15]. Recent work on alternatives to reencryption through rigidity and range-check style structure also separates the reencryption computation from the rejection interface [17]. The same separation appears in the harness statements below, where recomputation faults and branch-selection faults have different supports.

The HQC instantiation uses the salted FO analysis of Glabush, Hövelmanns, and Stebila [12], which formalizes the public salt used to mitigate multi-target collision attacks and ties the randomness and key derivations to public-key material, the message, and the salt. The public HQC specification dated 2025-08-22 adopts this salted implicit-rejection line and explicitly uses $H(ek_{\text{KEM}}||m||\text{salt})$ in the derivation of the key and reencryption randomness [8]. The experiments use the August 2025 specification as the main HQC object, with the earlier February line retained for drift checks.

For confirmation-code harnesses, the dependency-cone lower bound specializes observational equivalence to the declared target vector. Once the black-box observation factors through that vector, operations that are inactive for the target can be erased without changing the transcript distribution. The contribution is the target/cone grammar for confirmation-code testing, not the abstract indistinguishability fact by itself; the certificates below record the support, target, and implementation-class conditions under which a negative row becomes a non-certification theorem.

ML-KEM is standardized in FIPS 203 [21], and NIST SP 800-227 contains recommendations for implementing and using KEMs [1]. The claims below study the implementation signal created by confirmation-code binding in augmented variants derived from those baselines.

For the augmented line, construction-level provenance comes from the QROM security literature, including double-sided and measure-rewind styles of one-way-to-hiding, online and oracle-masked extraction, direct Kyber analyses, and FO lifting results [3, 19, 22, 5, 4, 2, 11, 10, 9, 20]. These results provide provenance for the standard FO security games cited in Section 5.9. The testing statements in this paper are classical ROM statements about pass probabilities under named testing harnesses.

3 Model

3.1 Canonical final-key point

We model the testing target as a confirmation-code-augmented FO KEM Π^C with algorithms $\text{KeyGen}, \text{Encaps}^C, \text{Decaps}^C$. All encodings in the model are length-delimited, domain-separated,

injective, and prefix-free unless an explicit aliasing error is stated. The final-key oracle G_K is a classical lazy-sampled random oracle returning κ -bit strings.

Definition 3.1 (Canonical confirmation-code FO test line). *A canonical honest-reference instance is a tuple*

$$\text{Inst} = (ek, dk, c, K, B, W, x^\star)$$

sampled as follows:

$$(ek, dk) \leftarrow \text{KeyGen}(1^\lambda), \quad (K, c, \tau) \leftarrow \text{Encaps}^C(ek).$$

The encapsulation transcript τ determines a localized transcript prefix

$$B = B(\tau) \in \{0, 1\}^*, \quad W = W(\tau) \in \mathcal{W}, \quad x^\star = \langle \text{good}, B, W \rangle.$$

The honest accepted key is

$$K = G_K(x^\star) \in \{0, 1\}^\kappa.$$

If the final-key encoding is not syntactically canonical, any adversarial collision with the canonical point is charged to the aliasing defect in Definition 3.2. The honest correctness error is

$$\delta_{\Pi^C} = \Pr[\text{Decaps}^C(dk, c) \neq K].$$

The harness withholds K , W , and $x^\star$ from the tested system except through declared public data and oracle access.

Definition 3.2 (Adversarial aliasing and final-key freshness). *Let Can be the canonical parser used for final-key oracle inputs, with undefined inputs mapped to $\perp$. For a run of a SUT $\mathcal{S}$, let $\mathcal{Q}_K^{\text{all}}$ be the set of all inputs queried by $\mathcal{S}$ to G_K . The adversarial aliasing event is*

$$\text{Alias}_{\mathcal{S}} = [\exists x \in \mathcal{Q}_K^{\text{all}} \setminus \{x^\star\} : \text{Can}(x) = \text{Can}(x^\star)].$$

Set

$$\epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) = \Pr[\text{Alias}_{\mathcal{S}}].$$

For a SUT class $\mathcal{C}$, write

$$\epsilon_{\text{alias}}(\mathcal{C}, \mathcal{H}) = \sup_{\mathcal{S} \in \mathcal{C}} \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}).$$

Let $\mathcal{E}$ be the event that honest correctness holds, no adversarial alias occurs, and the hidden point $x^\star$ is not queried. Let V be the complete SUT view at output time, including all declared oracle replies except $G_K(x^\star)$. If $\Pr[\mathcal{E}] = 0$, set $\epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) = 0$. Otherwise, the final-key freshness defect is

$$\epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) = \max \left\{ 0, \mathbb{E}_{v \leftarrow V | \mathcal{E}} \left[\max_{k \in \{0, 1\}^\kappa} \Pr[K = k \mid V = v, \mathcal{E}] \right] - 2^{-\kappa} \right\}.$$

For a class $\mathcal{C}$, define

$$\epsilon_{\text{fresh}}(\mathcal{C}, \mathcal{H}) = \sup_{\mathcal{S} \in \mathcal{C}} \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}).$$

The harness is secret-separating for $\mathcal{S}$ if $\epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) = 0$. It is $(\epsilon_{\text{alias}}, \epsilon_{\text{fresh}})$ -well formed for a class $\mathcal{C}$ if the corresponding suprema over $\mathcal{C}$ are bounded by those values.

In the salted HQC presentation of Section 4, the scheme-specific prefix is $B = H(ek_{\text{KEM}}) || m || \text{salt}$, whereas nonsalted lines use an empty or absent salt field. The witness W may be a source confirmation code, a tail-derived code, or a diagnostic code; its cryptographic interpretation depends on the theorems below.

3.2 Honest-reference testing

Definition 3.3 (Honest-reference harness). *For a possibly randomized, stateful system under test $\mathcal{S}$, the experiment $\text{HRef}_{\Pi^C, \mathcal{H}}(\mathcal{S})$ runs:*

1. *sample Inst as in Definition 3.1;*
2. *give $\mathcal{S}$ the decapsulation input prescribed by the harness, typically (dk, c, ek, aux) , but not K , W , or $x^\star$;*
3. *answer oracle queries using the same domain-separated random oracles as the reference line, while the reference evaluation $G_K(x^\star)$ is hidden unless $\mathcal{S}$ itself queries $x^\star$;*
4. *receive an output $\widehat{K}$ from $\mathcal{S}$;*
5. *output 1 iff $\widehat{K} = K$.*

Define

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) = \Pr[\text{HRef}_{\Pi^C, \mathcal{H}}(\mathcal{S}) = 1].$$

For a realized run, the localized final-key candidate set is

$$Q_G(B) = \{ w \in \mathcal{W} : \mathcal{S} \text{ queried } G_K(\langle \text{good}, B, w \rangle) \}.$$

The tested system is q_G -localized if $|Q_G(B)| \leq q_G$ in every run. The central hit event is

$$\text{Hit} = [W \in Q_G(B)].$$

Leakage is accounted for by the final-key freshness defect of Definition 3.2. This captures accidental exposure of the hidden key point through instrumentation, debug logs, shared oracle tables, deterministic seeds, transcript caches, or any other channel in the SUT view. A certified harness must give explicit reasons why $\epsilon_{\text{fresh}} = 0$, or must carry the stated freshness defect in the bound.

The harness model permits a broad tested system, the SUT may know dk , ek , c , the salt, and any auxiliary data explicitly declared by the harness, and it may be adaptive and stateful. The hidden confirmation witness W and the hidden reference key evaluation are excluded from the SUT input, except through the final-key query whose absence is being tested.

Definition 3.4 (Pass and detection probability). *For any harness $\mathcal{T} = (D, V)$, define*

$$\text{Pass}_{\Pi, \mathcal{S}, \mathcal{T}} = \Pr_{x \leftarrow D} [V(x, \mathcal{S}) = 1], \quad \text{Det}_{\Pi, \mathcal{S}, \mathcal{T}} = 1 - \text{Pass}_{\Pi, \mathcal{S}, \mathcal{T}}.$$

For a class $\mathcal{C}$, write

$$\text{Pass}_{\Pi, \mathcal{C}, \mathcal{T}} = \sup_{\mathcal{S} \in \mathcal{C}} \text{Pass}_{\Pi, \mathcal{S}, \mathcal{T}}.$$

3.3 Harness families

Across the theorem statements and artifact, five harness families distinguish the honest-reference signal from self-consistency, malformed-input, differential, and key-format behavior. The positive theorem is about $\mathcal{T}_{\text{href}}$; the other harnesses mark boundaries of what that test can certify.

Harness	Sampled object	Peer model	Main use in the paper
$\mathcal{T}_{\text{href}}$	reference-generated (ek, dk, c, K, B, W)	tested decapsulation against a sound peer	list-cUP detectability and binding mutants
$\mathcal{T}_{\text{hself}}$	tested endpoint's own (ek, dk, c, K)	faulty endpoint talks to itself	self-vs-reference separation
$\mathcal{T}_{\text{ct}}^\mu$	malformed ciphertext under mode μ	reference-vs-tested behavior on indexed support	branch-only and component-selective faults
$\mathcal{T}_{\text{diff}}^\mu$	same serialized input for both implementations	direct differential comparison	mode-selective signatures and drift checks
$\mathcal{T}_{\text{key}}^\nu$	key or layout perturbation under mode ν	parser and layout comparison	supplementary artifact coverage

Table 1: Harnesses used in the theorem statements.

Definition 3.5 (Honest-self harness). *The experiment $\text{HSelf}_{\Pi', \mathcal{H}}(\mathcal{S})$ samples key pairs, ciphertexts, and reference keys from the same endpoint Π' implemented by the SUT family. It accepts iff the decapsulated key returned by the tested endpoint agrees with that endpoint's own encapsulated key. This differs from $\text{HRef}_{\Pi^C, \mathcal{H}}$, which samples from the canonical reference Π^C and compares against $G_K(\langle \text{good}, B, W \rangle)$.*

Definition 3.6 (Support-equivalent SUTs). *Let $\mathcal{D}$ be a harness distribution over inputs $z = (ek, dk, c, \text{aux})$. Two randomized SUTs $\mathcal{S}_0, \mathcal{S}_1$ are Δ -support-equivalent on $\mathcal{D}$ if*

$$\Delta_{\text{TV}}((z, \mathcal{S}_0(z)), (z, \mathcal{S}_1(z))) \leq \Delta$$

for $z \leftarrow \mathcal{D}$.

Definition 3.7 (Reset-product testing). *A t -fold reset-product harness samples independent instances $\text{Inst}_1, \dots, \text{Inst}_t$, uses independent domain-separated oracle namespaces $G_K^{(1)}, \dots, G_K^{(t)}$, resets the SUT state between runs, and accepts iff all t runs accept.*

3.4 Fault vocabulary

Faults are named by the operation they omit or alter:

$$\mathcal{O} = \{O_\theta, O_{\text{reencrypt}}, O_{\text{code}}, O_{\text{compare}}, O_{\text{reject}}, O_{\text{bind}}, O_{\text{select}}\}.$$

Here O_θ and $O_{\text{reencrypt}}$ refer to deriving reencryption randomness and recomputing the ciphertext, O_{code} to producing the confirmation code, O_{bind} to including the mandated transcript inputs in the final key derivation, and the other entries to comparison, rejection, and key selection.

Class	Typical examples	Main harness behavior
Recomputation faults	skip reencryption, recompute only part of the ciphertext, overwrite the confirmation code	visible under $\mathcal{T}_{\text{href}}$ only when the omitted work is covered by the code or the ordinary key path
Binding faults	omit W , omit $H(ek_{\text{KEM}})$, omit salt, confuse transcript inputs	visible under $\mathcal{T}_{\text{href}}$ when they miss the hidden final-key point
Decision faults	always select K_+ , ignore rejection, compare only one component	often invisible on honest support and visible only under malformed or differential modes
Parsing and format faults	wrong key offset, wrong serialized layout, missing salt field	studied through $\mathcal{T}_{\text{key}}$ and $\mathcal{T}_{\text{diff}}$
Symmetric endpoint faults	apply the same binding mutation to both encapsulation and decapsulation	can pass $\mathcal{T}_{\text{hself}}$ and fail $\mathcal{T}_{\text{href}}$

Table 2: Fault classes used in the paper. The positive theorem concerns localized final-key list hits; operation coverage requires a separate code-source argument.

4 Split Presentation of the Augmented Line

For the artifact and proofs, we use a split presentation of the confirmation-code-augmented FO construction. The presentation separates the randomness derivation used for reencryption from the final key derivation, matches the salted HQC syntax, and places the confirmation code at one final random-oracle point. The reject-key expression uses the standard rejection secret σ stored in the decapsulation key.

Encapsulation	Decapsulation
<pre> 01 sample m and salt 02 set $B \leftarrow H(ek_{\text{KEM}}) \parallel m \parallel \text{salt}$ 03 set $\theta \leftarrow G_\theta(B)$ 04 compute $(c_{\text{PKE}}, W) \leftarrow \text{EncC}(ek_{\text{KEM}}, m; \theta)$ 05 set $c_{\text{KEM}} \leftarrow (c_{\text{PKE}}, \text{salt})$ 06 set $K \leftarrow G_K(\langle \text{good}, B, W \rangle)$ 07 return (K, c_{KEM}) </pre>	<pre> 01 parse $c_{\text{KEM}} = (c_{\text{PKE}}, \text{salt})$ 02 compute $m' \leftarrow \text{Dec}(dk_{\text{KEM}}, c_{\text{PKE}})$ 03 set $B' \leftarrow H(ek_{\text{KEM}}) \parallel m' \parallel \text{salt}$ 04 set $\theta' \leftarrow G_\theta(B')$ 05 compute $(c'_{\text{PKE}}, W') \leftarrow \text{EncC}(ek_{\text{KEM}}, m'; \theta')$ 06 set $K'_+ \leftarrow G_K(\langle \text{good}, B', W' \rangle)$ 07 set $K_- \leftarrow J(H(ek_{\text{KEM}}) \parallel \sigma \parallel c_{\text{KEM}})$ 08 return $\text{ct_select}(c'_{\text{PKE}} = c_{\text{PKE}}, K'_+, K_-)$ </pre>

Figure 1: Simplified split presentation used in the paper. The final good key is evaluated at a canonical point $\langle \text{good}, B, W \rangle$, where B is the transcript prefix and W is the recomputed confirmation witness.

Definition 4.1 (Split presentation of verifiable salted FO). *Let $\text{PKE}^C = (\text{KeyGen}, \text{EncC}, \text{Dec})$ be a PKE scheme whose encryption algorithm outputs a ciphertext and a confirmation code. The artifact line, denoted $\text{VSFO}[\text{PKE}^C, G_\theta, G_K, H, J]$, is the KEM in Figure 1. The ciphertext is*

$$c_{\text{KEM}} = (c_{\text{PKE}}, \text{salt}),$$

the reencryption randomness is derived as

$$\theta = G_\theta(H(ek_{\text{KEM}}) \parallel m \parallel \text{salt}),$$

and the good key is derived as

$$K_+ = G_K(\langle \text{good}, H(ek_{\text{KEM}}) \parallel m \parallel \text{salt}, cd \rangle).$$

In the split presentation, G_θ controls the deterministic reencryption path and G_K receives the recomputed code after reencryption, which are the two oracle points used by the testing statement. A fault that misses cd lacks the honest final good-key query; the FO comparison and rejection logic is unchanged.

Remark 4.2 (Relation to the 2025 construction). *The construction follows the same confirmation-code principle as [13]. Source security terms from that work are transported to this split presentation only when the source-equivalence certificate of Definition 4.3 is satisfied. Diagnostic variants that change or truncate the witness are handled by the diagnostic entropy route instead. The explicit split matches the August 2025 HQC salted transcript $H(ek_{\text{KEM}})\|m\|\text{salt}$ and the canonical final point $G_K(\langle \text{good}, B, W \rangle)$ used in the testing theorem.*

Definition 4.3 (Source-equivalent split presentation). *Let Π^{src} be the source confirmation-code transform and let Π^{split} be a split presentation with separate G_θ and G_K domains. The split presentation is source-equivalent for a SUT class C and harness $\mathcal{H}$ if there is an efficiently computable bijection Φ on transcripts such that:*

- (S1) *(ek, dk, c, B, W) has the same distribution in Π^{src} and Π^{split} after applying Φ .*
- (S2) *The source final-key oracle input and the split final-key input are related by a domain-separated injective relabeling, so random-oracle tables can be coupled by renaming inputs.*
- (S3) *The G_θ queries of the split presentation are simulatable from the source cUP interface and do not reveal any additional function of W beyond the source transcript.*
- (S4) *The accept predicate and honest-reference key equality event coincide after the same transcript relabeling.*

Proposition 4.4 (Transfer across a source-equivalent split). *If Π^{split} is source-equivalent to Π^{src} in the sense of Definition 4.3, then cUP, list-cUP, and fCOR advantages for the two presentations are equal up to the explicit relabeling of oracle domains. In particular, a source cUP bound may be used in Theorem 5.10 for the split presentation only under this source-equivalence certificate.*

Proof. Run the source experiment and the split experiment with the same sampled transcript under the bijection Φ . Couple their random oracles by the injective domain relabeling from condition (S2), and answer all G_θ queries using the simulator from (S3). Conditions (S1) and (S4) give identical distributions of (B, W) and identical acceptance events. Therefore any adversary in one presentation is transformed into an adversary in the other with the same success probability, and conversely. $\square$

Remark 4.5 (Diagnostic splits are not source transfers). *The ML-KEM-CD1 and HQC-CD1/CD* artifact lines are not asserted to be source-equivalent to the source ML-KEM confirmation-code construction. They change the witness source or shorten it diagnostically. Their interpretation therefore goes through the hashed-witness theorem, ML-KEM-CD1 RawEnt, or HQC TailEnt, not through direct transfer of the source cUP theorem.*

Testing split. In an FO implementation, randomness derivation, good-key derivation, and transcript binding can be close together in the code and in the proof syntax. For the present testing question, the reencryption randomness and the final key derivation are separate oracle points. The confirmation code is used after reencryption has produced the value bound into the final key. A fault that skips a code-covered computation and therefore fails to determine W loses the exact value needed at the final key point; operations outside the witness dependency cone require separate harness or code-source evidence.

Rejection path. The ordinary rejection structure is unchanged: decapsulation decrypts, recomputes, compares, and selects between a good key and a reject key. The support-coincidence theorem therefore applies to the transformed line as stated. A branch-only fault that agrees with correct decapsulation on all honestly generated ciphertexts is a malformed-support question.

5 Detectability Results

5.1 Honest-reference list-hit bound

Theorem 5.1 (Honest-reference list-hit bound with adversarial bad events). *Let Π^C be a canonical confirmation-code FO line with final key length κ and correctness error δ_{Π^C} . For any possibly randomized, stateful SUT $\mathcal{S}$ and honest-reference harness $\mathcal{H}$,*

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \Pr[W \in Q_G(B)] + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + 2^{-\kappa}.$$

For a class $\mathcal{C}$, the same bound holds with the class suprema of ϵ_{alias} , ϵ_{fresh} , and the list-hit probability.

Proof. Let $\text{Corr} = [\text{Decaps}^C(dk, c) = K]$ be the honest correctness event, and let $\text{Alias}_{\mathcal{S}}$ be as in Definition 3.2. Put

$$\text{Bad} = \neg \text{Corr} \cup \text{Alias}_{\mathcal{S}}.$$

Then

$$\Pr[\text{Bad}] \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}).$$

Let $\text{Acc} = [\widehat{K} = K]$. Decompose

$$\Pr[\text{Acc}] \leq \Pr[\text{Bad}] + \Pr[\text{Acc} \wedge \neg \text{Bad} \wedge \text{Hit}] + \Pr[\text{Acc} \wedge \neg \text{Bad} \wedge \neg \text{Hit}].$$

The middle term is at most $\Pr[\text{Hit}] = \Pr[W \in Q_G(B)]$. For the final term, condition on $\mathcal{E} = \neg \text{Bad} \wedge \neg \text{Hit}$ and on a complete output-time view $V = v$. By the definition of ϵ_{fresh} , the average posterior guessing probability for K from that view is at most $2^{-\kappa} + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H})$. Therefore

$$\Pr[\text{Acc} \wedge \neg \text{Bad} \wedge \neg \text{Hit}] \leq 2^{-\kappa} + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}).$$

Combining the three terms proves the claim. $\square$

Theorem 5.2 (Optimality of the hidden-point reduction). *Let Π^C be a canonical honest-reference line with $\delta_{\Pi^C} = 0$, no adversarial aliases, and a secret-separating harness in the sense of Definition 3.2. Let V be any SUT view before localized final-key queries, and let A be any measurable single-candidate strategy that outputs a suffix $A(V) \in \mathcal{W}$. Then there exists a 1-localized SUT $\mathcal{S}_A$ such that*

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}_A) = \Pr[A(V) = W] + (1 - \Pr[A(V) = W])2^{-\kappa}.$$

Consequently, any black-box honest-reference theorem whose hypotheses do not bound the posterior predictability of W from the SUT view cannot give a nontrivial bound below the best single-candidate witness-prediction probability, up to the fresh-key coincidence term.

Proof. The SUT $\mathcal{S}_A$ computes $w = A(V)$, queries $G_K(\langle \text{good}, B, w \rangle)$, and outputs the oracle reply. If $w = W$, this query is the hidden reference point $x^\star$, and the output is exactly $K = G_K(x^\star)$. If $w \neq W$, then by canonical injectivity the queried oracle point is distinct from $x^\star$. Since the harness is secret-separating and no alias occurs, the returned value and K are independent uniform κ -bit strings. The conditional success probability in this case is therefore $2^{-\kappa}$. Averaging over the two cases yields the equality. The final statement follows by taking A to be the best single-candidate predictor available from the view. $\square$

Remark 5.3 (List-hit tightness). *Theorem 5.1 reduces harness acceptance to a list-hit term. Theorem 5.2 shows that the term is tight: a predictor for the hidden witness induces a one-query SUT with the corresponding pass probability, up to fresh-key coincidence. Source cUP, conditional min-entropy, hashed-witness entropy, and TailEnt are the assumptions used to bound this term for the different instantiations.*

Remark 5.4 (Scope of the list-hit bound). *The theorem is a list-hit statement. Unpredictability of a particular code source and coverage of omitted operations are separate hypotheses. Once correctness, adversarial aliasing, and final-key freshness are accounted for, honest-reference acceptance is controlled by the SUT's list of localized final-key suffix candidates and by final-key guessing.*

5.2 Harness list-cUP bridge

The verifiable-decapsulation paper defines confirmation-code unpredictability (cUP) and faulty implementation correctness (fCOR) for confirmation-code transforms [13]. In a test harness, a SUT may make several localized final-key queries; therefore the matching source notion is a list version of cUP.

Definition 5.5 (Harness list-cUP). *Fix a canonical line Π^C , a harness $\mathcal{H}$, and a restricted oracle interface $\mathcal{F}$ matching the cUP interface of the source line. The experiment*

$$\text{L-cUP}_{\Pi^C, \mathcal{H}, \mathcal{F}}^q(\mathcal{A})$$

samples $(ek, dk) \leftarrow \text{KeyGen}(1^\lambda)$ and

$$(K, c, \tau) \leftarrow \text{Encaps}^C(ek), \quad B = B(\tau), \quad W = W(\tau).$$

The experiment sends $\mathcal{A}$ the public and secret data allowed by the cUP-style experiment, including the harness data such as ek, dk, c , but not W and not K . It grants $\mathcal{A}$ access to $\mathcal{F}$, with the localized final-key oracle $G_K(\langle \text{good}, B, \cdot \rangle)$ removed or blinded. The adversary outputs a list $L \subseteq \mathcal{W}$ with $|L| \leq q$. The experiment outputs 1 iff $W \in L$. Define

$$\text{Adv}_{\Pi^C, \mathcal{H}, \mathcal{F}}^{\text{L-cUP}, q}(\mathcal{A}) = \Pr[\text{L-cUP}_{\Pi^C, \mathcal{H}, \mathcal{F}}^q(\mathcal{A}) = 1].$$

For a class $\mathcal{C}$, take the supremum over $\mathcal{A} \in \mathcal{C}$.

Definition 5.6 (cUP-faithful harness certificate). *Fix a canonical augmented line Π^C , a source cUP interface $\mathcal{F}$, a harness $\mathcal{H}$, a SUT class $\mathcal{C}$, and a localization bound q . A certificate*

$$\text{Cert}_{\text{cUP}} = (\text{Inst}, \text{Orc}, \text{View}, \text{Log})$$

proves that $(\mathcal{H}, \mathcal{F}, \mathcal{C}, q)$ is cUP-faithful if the following conditions hold for every $S \in \mathcal{C}$.

- (C1) **Instance equivalence.** *The tuple (ek, dk, c, B, W) sampled by $\mathcal{H}$ has the same distribution as the tuple sampled in the source cUP experiment for Π^C .*
- (C2) **View equivalence.** *The initial SUT view produced by $\mathcal{H}$ is simulatable from the source cUP public and secret inputs allowed by $\mathcal{F}$, and contains no additional function of W, K , or $x^\star$.*
- (C3) **Oracle equivalence.** *Every oracle reply seen by S in $\mathcal{H}$, except calls to the localized final-key family $G_K(\langle \text{good}, B, \cdot \rangle)$, is exactly simulatable using $\mathcal{F}$ and independent lazy sampling with the same domains.*
- (C4) **Final-key logging.** *Calls by S to $G_K(\langle \text{good}, B, w \rangle)$ are not answered by the list-cUP simulator as reference-key evaluations. They are logged as suffixes w by Log, and at most q distinct suffixes are logged in every run.*
- (C5) **State containment.** *The SUT receives no side channel, debug output, shared oracle table, deterministic seed, transcript cache, or log entry that is absent from the simulated source interface.*

(C6) **No extra witness path.** Any computation, leakage, cache, or oracle interaction through which $\mathcal{S}$ can determine W must be represented in the source cUP interface $\mathcal{F}$ and therefore counted in the source cUP advantage. If dk , auxiliary data, or public code give a SUT class an alternate way to compute W that is not simulated by $\mathcal{F}$, then that class is not cUP-faithful for this harness.

Remark 5.7 (Scope of the source-cUP route). The source-cUP route is not a statement about arbitrary decapsulation programs given dk . It applies to SUT classes whose complete view and allowed computations are simulatable in the source cUP experiment. An implementation that recomputes W , obtains it through an auxiliary channel, or reconstructs it by a path outside $\mathcal{F}$ is handled by the list-hit theorem itself, or by an entropy/diagnostic assumption, not by importing the source cUP bound.

Lemma 5.8 (Certified extraction of the list-cUP adversary). If $(\mathcal{H}, \mathcal{F}, C, q)$ is cUP-faithful, then every q -localized $\mathcal{S} \in C$ induces a list-cUP adversary $\mathcal{A}_{\mathcal{S}}$ whose output list L satisfies

$$(B, W, L) \equiv (B, W, Q_G(B))$$

in distribution.

Proof. The adversary $\mathcal{A}_{\mathcal{S}}$ receives the source cUP instance and runs the simulator specified by the certificate. By instance equivalence, the hidden pair (B, W) has the same distribution as in the honest-reference harness. By view and oracle equivalence, every message delivered to $\mathcal{S}$, except localized final-key answers, has the same distribution as in the harness. When $\mathcal{S}$ attempts a localized final-key query $G_K(\langle \text{good}, B, w \rangle)$, the adversary logs w and returns a fresh dummy value from an independently sampled table that is never used as the reference key. The resulting execution is the list-cUP experiment, with the final-key oracle family removed or blinded. State containment excludes additional leakage. The logged list is therefore precisely $Q_G(B)$ under the simulated run, and its joint distribution with (B, W) is preserved. $\square$

Lemma 5.9 (List-cUP to single-cUP loss). If the corresponding single-candidate cUP game has advantage at most ϵ_{cUP} for every adversary in the same simulation class, then

$$\text{Adv}^{\text{L-cUP}, q} \leq q\epsilon_{\text{cUP}}.$$

Proof. Let $\mathcal{A}$ output a list L with $|L| \leq q$. A single-candidate adversary $\mathcal{B}$ runs $\mathcal{A}$, and if $L \neq \emptyset$, outputs a uniformly chosen element of L . Conditioned on $W \in L$, the probability that $\mathcal{B}$ outputs W is at least $1/q$. Therefore

$$\Pr[\mathcal{B} \text{ wins cUP}] \geq \frac{1}{q} \Pr[W \in L].$$

The assumed cUP bound implies $\Pr[W \in L] \leq q\epsilon_{\text{cUP}}$, and taking the supremum yields the lemma. $\square$

Theorem 5.10 (Certified honest-reference testing from source cUP). Suppose $(\mathcal{H}, \mathcal{F}, C, q_G)$ admits a cUP-faithful harness certificate in the sense of Definition 5.6. Then every q_G -localized $\mathcal{S} \in C$ satisfies

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + \text{Adv}_{\Pi^C, \mathcal{H}, \mathcal{F}}^{\text{L-cUP}, q_G}(C) + 2^{-\kappa}.$$

If the source construction provides a single-candidate cUP bound $\text{Adv}_{\Pi^C, \mathcal{F}}^{\text{cUP}}(C) \leq \epsilon_{\text{cUP}}$, then

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + q_G\epsilon_{\text{cUP}} + 2^{-\kappa}.$$

Proof. Apply Theorem 5.1. By Lemma 5.8, $\Pr[W \in Q_G(B)]$ in the SUT experiment is exactly the list-cUP success probability of the induced adversary. The first bound follows by taking the class supremum. The second bound follows from Lemma 5.9. $\square$

Proposition 5.11 (Honest-reference harness and fCOR). *When the instance distribution, oracle interface, correctness convention, and accepted key are the same as in the source fCOR experiment,*

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) = \Pr[\text{fCOR}^{\Pi^C, \mathcal{S}} \Rightarrow 1].$$

The list-cUP bridge identifies the additional loss incurred by allowing the tested system to try a list of final-key suffixes.

Proof. Under the stated matching conditions, both experiments sample the same key pair, ciphertext, hidden accepted key, and oracle replies, and both accept exactly on equality with the honest encapsulated key. The probabilities are therefore identical. $\square$

5.3 Average conditional min-entropy

Definition 5.12 (Average conditional min-entropy). *Let $W \in \mathcal{W}$ be the hidden confirmation witness and let V be a random variable with respect to which a candidate set $Q(V) \subseteq \mathcal{W}$ is measurable. Define*

$$P_{\text{guess}}(W | V) = \mathbb{E}_{v \leftarrow V} \left[\max_{w \in \mathcal{W}} \Pr[W = w | V = v] \right].$$

The average conditional min-entropy is

$$\tilde{H}_{\infty}(W | V) = -\log_2 P_{\text{guess}}(W | V).$$

We say W has $(\lambda, \epsilon_{\text{sm}})$ -smooth average conditional min-entropy given V if there is an event Good with $\Pr[\neg \text{Good}] \leq \epsilon_{\text{sm}}$ such that $\tilde{H}_{\infty}(W | V, \text{Good}) \geq \lambda$.

Lemma 5.13 (List hit from average min-entropy). *Let $Q(V) \subseteq \mathcal{W}$ be V -measurable and satisfy $|Q(V)| \leq q$. If $\tilde{H}_{\infty}(W | V) \geq \lambda$, then*

$$\Pr[W \in Q(V)] \leq q2^{-\lambda}.$$

With $(\lambda, \epsilon_{\text{sm}})$ -smooth min-entropy,

$$\Pr[W \in Q(V)] \leq \epsilon_{\text{sm}} + q2^{-\lambda}.$$

Proof. For the unsmoothed case,

$$\Pr[W \in Q(V)] = \sum_v \Pr[V = v] \Pr[W \in Q(v) | V = v].$$

For each v ,

$$\Pr[W \in Q(v) | V = v] \leq |Q(v)| \max_w \Pr[W = w | V = v] \leq q \max_w \Pr[W = w | V = v].$$

Summing over v yields $qP_{\text{guess}}(W | V) \leq q2^{-\lambda}$. The smoothed case decomposes over Good and charges $\Pr[\neg \text{Good}] \leq \epsilon_{\text{sm}}$. $\square$

Theorem 5.14 (Entropic honest-reference detectability). *In the setting of Theorem 5.1, suppose $\mathcal{S}$ is q_G -localized and that, for a view V making $Q_G(B)$ measurable,*

$$\tilde{H}_{\infty}(W | V) \geq \lambda.$$

Then

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + q_G 2^{-\lambda} + 2^{-\kappa}.$$

If only $(\lambda, \epsilon_{\text{sm}})$ -smooth min-entropy holds, then

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{sm}} + q_G 2^{-\lambda} + 2^{-\kappa}.$$

Proof. Combine Theorem 5.1 with Lemma 5.13, using $Q(V) = Q_G(B)$. $\square$

5.4 Hashed diagnostic codes

Definition 5.15 (Hashed diagnostic code). Let $R \in \mathcal{R}$ be a hidden raw witness determined by the honest reference transcript. A diagnostic confirmation code of length L is

$$W = \text{Trunc}_L(H_{\text{cd}}(\langle \text{cd}, B, R \rangle)),$$

where H_{cd} is a domain-separated random oracle independent of G_K . The final accepted key is $K = G_K(\langle \text{good}, B, W \rangle)$. Let q_H be the number of H_{cd} -queries made by the SUT at points of the form $\langle \text{cd}, B, r \rangle$, and let q_G be the number of localized final-key suffix candidates.

Lemma 5.16 (Hashed-witness hit bound). Let V_0 be a view with respect to which the SUT strategy before H_{cd} -queries is measurable. If $\tilde{H}_\infty(R \mid V_0) \geq \rho$, then

$$\Pr[W \in Q_G(B)] \leq q_H 2^{-\rho} + q_G 2^{-L}.$$

With $(\rho, \epsilon_{\text{sm}})$ -smooth min-entropy,

$$\Pr[W \in Q_G(B)] \leq \epsilon_{\text{sm}} + q_H 2^{-\rho} + q_G 2^{-L}.$$

Proof. Let RawHit be the event that the SUT queries $H_{\text{cd}}(\langle \text{cd}, B, R \rangle)$. By Lemma 5.13 applied to the raw witness R ,

$$\Pr[\text{RawHit}] \leq q_H 2^{-\rho}$$

in the unsmoothed case, and gains the smoothing loss in the smooth case. Conditioned on $\neg \text{RawHit}$, the value $H_{\text{cd}}(\langle \text{cd}, B, R \rangle)$ has not been revealed to the SUT, so its L -bit truncation W is uniform and independent of $Q_G(B)$. Since $|Q_G(B)| \leq q_G$, the conditional hit probability is at most $q_G 2^{-L}$. Combining the cases proves the lemma. $\square$

Theorem 5.17 (Hashed diagnostic detectability). In the setting of Definition 5.15, suppose that the raw witness has $\tilde{H}_\infty(R \mid V_0) \geq \rho$ for a pre- H_{cd} -query view V_0 . Then

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + q_H 2^{-\rho} + q_G 2^{-L} + 2^{-\kappa}.$$

If R only has $(\rho, \epsilon_{\text{sm}})$ -smooth average conditional min-entropy given V_0 , add ϵ_{sm} to the right-hand side.

Proof. Combine Theorem 5.1 with Lemma 5.16. $\square$

5.5 Range, tail witnesses, and HQC

Lemma 5.18 (Range does not imply entropy). If a random variable C has support size at most 2^L , then $\tilde{H}_\infty(C) \leq L$. However, no positive lower bound on $\tilde{H}_\infty(C \mid V)$ follows from the range size alone.

Proof. Since $|\text{Supp}(C)| \leq 2^L$, some value has probability at least 2^{-L} , so $\tilde{H}_\infty(C) \leq L$. For the conditional claim, take $V = C$. Then for every v , the posterior guessing probability is 1, and $\tilde{H}_\infty(C \mid V) = 0$, while the range has not changed. $\square$

Definition 5.19 (HQC tail witness in the 2025 salted-SFO line). In the public HQC KEM specification dated 2025-08-22, encryption forms

$$u = r_1 + hr_2, \quad v = C.\text{Encode}(m) + \text{Truncate}(sr_2 + e, \ell),$$

and the KEM derives

$$(K, \theta) = G(H(ek_{\text{KEM}}) \| m \| \text{salt}).$$

The ciphertext is $c_{\text{KEM}} = (c_{\text{PKE}}, \text{salt})$. Let

$$\ell_{\text{tail}} = n - n_1 n_2$$

and define the tail witness

$$T = (sr_2 + e)[n_1 n_2, \dots, n - 1] \in \{0, 1\}^{\ell_{\text{tail}}}.$$

For a retained tail length $L \leq \ell_{\text{tail}}$, let

$$W = \text{Tail}_L(T).$$

Proposition 5.20 (HQC retained range budget). *For HQC-1, HQC-3, and HQC-5, the retained tail code $W = \text{Tail}_L(T)$ has range size at most 2^L with*

$$L \leq 5, \quad L \leq 11, \quad L \leq 37,$$

respectively. A one-byte tail diagnostic has retained range bits

$$\min(8, \ell_{\text{tail}}) = 5, 8, 8$$

for HQC-1/3/5.

Proof. By Definition 5.19, $T \in \{0, 1\}^{\ell_{\text{tail}}}$. Any retained L -bit projection has support size at most 2^L . The public parameter relation is

$$17669 - 46 \cdot 384 = 5, \quad 35851 - 56 \cdot 640 = 11, \quad 57637 - 90 \cdot 640 = 37.$$

□

Definition 5.21 (HQC tail conditional min-entropy assumption). *For a harness class $\mathfrak{H}$, SUT class C , retained length L , and error ϵ_{tail} , define*

$$\text{HQC-TailEnt}_{L, \epsilon_{\text{tail}}}^{\mathfrak{H}, C}$$

to be the assumption that, after sampling a canonical honest-reference HQC instance and exposing exactly the harness view V to every $S \in C$, excluding W , K , and the private reference evaluation $G_K(\langle \text{good}, B, W \rangle)$, one has

$$P_{\text{guess}}(W \mid V) \leq 2^{-L} + \epsilon_{\text{tail}}.$$

Equivalently,

$$\tilde{H}_{\infty}(W \mid V) \geq -\log_2(2^{-L} + \epsilon_{\text{tail}}).$$

Theorem 5.22 (HQC tail-test detectability under TailEnt). *Assume $\text{HQC-TailEnt}_{L, \epsilon_{\text{tail}}}^{\mathfrak{H}, C}$. For any q_G -localized SUT $S \in C$,*

$$\text{Pass}_{\text{HQC}^C, \mathcal{H}}^{\text{href}}(S) \leq \delta_{\text{HQC}^C} + \epsilon_{\text{alias}}(S, \mathcal{H}) + \epsilon_{\text{fresh}}(S, \mathcal{H}) + q_G(2^{-L} + \epsilon_{\text{tail}}) + 2^{-\kappa}.$$

Proof. The TailEnt assumption implies $P_{\text{guess}}(W \mid V) \leq 2^{-L} + \epsilon_{\text{tail}}$. The proof of Lemma 5.13 yields

$$\Pr[W \in Q_G(B)] \leq q_G(2^{-L} + \epsilon_{\text{tail}}).$$

Apply Theorem 5.1.

□

Proposition 5.23 (Necessity of TailEnt for tail-code detectability). *Fix an HQC tail-code harness with zero correctness error, no aliasing, and a secret-separating final-key point. Suppose there is a measurable predictor $A(V)$ for the retained tail code such that*

$$\Pr[A(V) = W] = \alpha.$$

Then there exists a 1-localized SUT $\mathcal{S}_A$ satisfying

$$\text{Pass}_{\text{HQC}^C, \mathcal{H}}^{\text{href}}(\mathcal{S}_A) = \alpha + (1 - \alpha)2^{-\kappa}.$$

Thus any claimed bound below this quantity must rule out such predictors. The assumption $\text{HQC-TailEnt}_{L, \epsilon_{\text{tail}}}^{\mathfrak{S}, C}$ does exactly this by imposing $\alpha \leq 2^{-L} + \epsilon_{\text{tail}}$ for every single-candidate predictor in the modeled view.

Proof. The SUT queries $G_K(\langle \text{good}, B, A(V) \rangle)$ and outputs the reply. If $A(V) = W$, the queried point is the hidden reference point. Otherwise it is a distinct canonical oracle point and the returned value equals the hidden key only with probability $2^{-\kappa}$. Averaging over the predictor's success event yields the equality. The last sentence is the single-candidate case of the TailEnt guessing bound. $\square$

Remark 5.24 (Reading range-budget rows). *A retained range of L bits says only that the code can contain at most L bits of uncertainty. It implies no lower bound on the posterior uncertainty left to a SUT. The unconditional HQC statement in this paper is therefore the range budget in Proposition 5.20. The conditional detectability statement is Theorem 5.22, which additionally assumes TailEnt. By Proposition 5.23, this split is necessary: if the SUT view admits a predictor for the retained tail code, then a one-query SUT can pass at that prediction rate up to the final-key coincidence term.*

5.6 Honest-self, support, and repetition

Theorem 5.25 (Symmetric code erasure passes self-tests and fails reference tests). *Let Π^{-W} be the endpoint obtained from Π^C by replacing both encapsulation and decapsulation final-key derivation*

$$G_K(\langle \text{good}, B, W \rangle)$$

with

$$G_K(\langle \text{good}, B \rangle).$$

Assume domain separation makes these two oracle inputs distinct for all legal W , and assume no oracle aliasing. Then

$$\text{Pass}_{\Pi^{-W}, \mathcal{H}}^{\text{hself}}(\Pi^{-W}) \geq 1 - \delta_{\Pi^{-W}},$$

while

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\Pi^{-W}) \leq \delta_{\Pi^C} + 2^{-\kappa}.$$

Proof. For honest-self testing, both sides use Π^{-W} . Except on the endpoint's own correctness failure, encapsulation and decapsulation both use $G_K(\langle \text{good}, B \rangle)$. For honest-reference testing, the hidden target is $G_K(\langle \text{good}, B, W \rangle)$, while the SUT outputs $G_K(\langle \text{good}, B \rangle)$. These are distinct oracle inputs and the erased-code endpoint never queries the hidden point. Apply Theorem 5.1 with $\Pr[\text{Hit}] = 0$, $\epsilon_{\text{alias}} = 0$, and $\epsilon_{\text{fresh}} = 0$. $\square$

Proposition 5.26 (Support equivalence preserves pass probability). *If $\mathcal{S}_0$ and $\mathcal{S}_1$ are Δ -support-equivalent on the honest-reference harness distribution $\mathcal{D}_{\text{href}}$, then*

$$\left| \text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}_0) - \text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}_1) \right| \leq \Delta.$$

Proof. Acceptance is an event on the joint distribution of the harness input and SUT output. Total variation distance bounds the difference in probability of any event. $\square$

Corollary 5.27 (Deterministic support coincidence). *If two deterministic SUTs output the same value on every input in the support of $\mathcal{D}_{\text{href}}$, then their honest-reference pass probabilities are equal.*

Corollary 5.28 (Decision-only faults on honest support). *If a decision-only fault agrees with correct decapsulation on every honestly generated reference ciphertext, then it has the same honest-reference pass probability as the correct line. Malformed or differential harnesses sample different supports and may expose the fault.*

Theorem 5.29 (Reset amplification). *In a t -fold reset-product harness, the instances, oracle namespaces, SUT state, and SUT coins of the individual runs are independent after reset. If run i , under its own marginal reset distribution, satisfies $\Pr[\text{Acc}_i] \leq p_i$, then*

$$\Pr[\text{Acc}_1 \wedge \dots \wedge \text{Acc}_t] \leq \prod_{i=1}^t p_i.$$

If $p_i \leq p$ for all i , the all-pass probability is at most p^t . Under the smooth entropic bound with uniform parameters, one may take

$$p = \delta_{\Pi^C} + \epsilon_{\text{alias}} + \epsilon_{\text{fresh}} + \epsilon_{\text{sm}} + q_G 2^{-\lambda} + 2^{-\kappa}.$$

Proof. The reset-product harness resets the SUT and uses independent instances with independent oracle namespaces, so the run distribution factors. The product bound follows immediately. $\square$

Remark 5.30 (Non-reset campaigns). *For a non-reset stateful campaign,*

$$\Pr[\text{all accept}] = \prod_{i=1}^t \Pr[\text{Acc}_i \mid \text{Acc}_1, \dots, \text{Acc}_{i-1}].$$

Exponentiation is valid only if the per-round hypotheses, including entropy and query localization, hold under this conditioning.

5.7 Dependency-cone lower bound for honest-reference testing

The preceding theorems show how a confirmation-code harness can prove a positive statement once the hidden witness is unpredictable or cUP-secure. We now prove the matching negative statement: over honest ciphertext support, a black-box honest-reference harness cannot certify execution of operations that are outside the dependency cone exposed by the confirmation witness and the harness observation, relative to implementation classes that contain the corresponding erasure faults.

Definition 5.31 (Intervened trace). *Fix an implementation $\mathcal{I}$, an input z , implementation coins ω , and a lazy-sampled oracle table $\mathcal{R}$. Let*

$$\text{Trace}_{\mathcal{I}}(z, \omega, \mathcal{R}) = (O, \prec, \text{val})$$

be the deterministic trace DAG induced by that run. For a set $U \subseteq O$ and a replacement vector

$$a_U \in \prod_{u \in U} \text{Dom}(u),$$

the intervened trace

$$\text{Trace}_{\mathcal{I}}(z, \omega, \mathcal{R})[U \leftarrow a_U]$$

is obtained by replacing the value of every node in U by the corresponding entry of a_U , then recomputing all descendants in topological order using the same local transition functions and the same oracle table $\mathcal{R}$. Non-descendants are unchanged.

Remark 5.32 (Trace abstraction). *The trace model is a typed, total abstraction of the black-box execution being certified. Control-flow predicates, branch selectors, parsing decisions, and error states that can affect the black-box target are represented as trace nodes or as components of the target vector. Dummy values used in an intervention must be valid inhabitants of the declared node domains. A crash, undefined behavior, or changed public output is covered only if it appears in the declared black-box observation; timing, memory layout, debug output, and instrumentation state are outside the theorem unless the harness explicitly adds them to the target.*

Definition 5.33 (Confirmation-observable target vector). *Let $\mathcal{H}_{\text{hon}}$ be an honest-reference harness whose input distribution $\mathcal{D}_{\text{hon}}$ is supported on honestly generated ciphertexts. The confirmation-observable target vector*

$$\text{Tgt}_{\text{cc}, \mathcal{H}}(\text{Trace})$$

contains exactly the trace values through which the black-box observation of $\mathcal{H}$ factors on honest support:

1. *the localized prefix B , the retained witness W , and the canonical final-key point $x^* = \langle \text{good}, B, W \rangle$;*
2. *all final-key-oracle inputs and declared oracle-query inputs that the harness is allowed to observe or emulate;*
3. *all declared public outputs of the SUT interface; and*
4. *the branch selector restricted to honest support, after quotienting out support-constant predicates.*

The harness is black-box over honest support if, for any two coupled runs with the same value of $\text{Tgt}_{\text{cc}, \mathcal{H}}$, the complete harness observation has the same distribution. The complete observation is written

$$\text{Obs}_{\mathcal{H}}(\mathcal{I}; z, \omega, \mathcal{R}),$$

and includes the visible input/output transcript, the declared oracle query log, and the verifier's accept bit, but not undeclared internal trace nodes, timing, memory layout, debug output, or instrumentation state.

Definition 5.34 (Support-active dependency cone). *Let $\mathcal{D}$ be a distribution over inputs and let Tgt be a target vector on traces. A set $U \subseteq \mathcal{O}$ is inactive for Tgt on $\mathcal{D}$, written*

$$U \perp_{\mathcal{D}} \text{Tgt},$$

if for every $z \in \text{supp}(\mathcal{D})$, every implementation coin string ω , every oracle table $\mathcal{R}$, and every replacement vector a_U ,

$$\text{Tgt}(\text{Trace}_{\mathcal{I}}(z, \omega, \mathcal{R})[U \leftarrow a_U]) = \text{Tgt}(\text{Trace}_{\mathcal{I}}(z, \omega, \mathcal{R})).$$

The support-active dependency cone is the complement of the inactive sets: an operation belongs to $\text{Cone}_{\mathcal{D}}(\text{Tgt})$ if changing it can change Tgt on some point of $\text{supp}(\mathcal{D})$, for some fixing of coins and oracle table. We call

$$\text{Cone}_{\text{cc}, \mathcal{H}} = \text{Cone}_{\mathcal{D}_{\text{hon}}}(\text{Tgt}_{\text{cc}, \mathcal{H}})$$

the support-active confirmation cone of $\mathcal{H}$.

For set-valued claims, inactivity must be certified for the whole set U . It is not inferred from singleton non-membership unless the trace semantics proves that simultaneous interventions on the set cannot create a new path to Tgt .

Definition 5.35 (Erasure fault and execution certifier). *For $U \subseteq \mathcal{O}$, a U -erasure implementation $\mathcal{I}^{-U}$ is an implementation that never executes operations in U , and instead supplies fixed dummy values at those locations before continuing with the remaining transition functions. Write $\text{Exec}_U(\mathcal{I}) = 1$ if $\mathcal{I}$ executes every operation in U on every input in the honest support, and $\text{Exec}_U(\mathcal{I}) = 0$ if it is a U -erasure implementation.*

A class C is U -erasure closed for a harness $\mathcal{H}$ if, for every $I \in C$ with $\text{Exec}_U(I) = 1$, at least one coupled U -erasure implementation constructed by Lemma 5.36 also belongs to C . Equivalently, C contains the erasure faults against which the certifier claims soundness.

A black-box n -sample certification rule Cert for U -execution sees only n independent observations from $\text{Obs}_{\mathcal{H}}$ and outputs 1 to certify that U was executed. It is (α, β) -sound and complete for U -execution on a class C if

$$\Pr[\text{Cert} = 1 \mid I] \leq \alpha \quad \text{for all } I \in C \text{ with } \text{Exec}_U(I) = 0,$$

and

$$\Pr[\text{Cert} = 1 \mid I] \geq 1 - \beta \quad \text{for all } I \in C \text{ with } \text{Exec}_U(I) = 1.$$

A nontrivial certifier has $\alpha + \beta < 1$.

Lemma 5.36 (Erasure coupling outside the cone). *Let $\mathcal{H}_{\text{hon}}$ be black-box over honest support. If*

$$U \perp_{\mathcal{D}_{\text{hon}}} \text{Tgt}_{\text{cc}, \mathcal{H}},$$

then for every implementation I there exists a U -erasure implementation I^{-U} such that

$$\text{Obs}_{\mathcal{H}}(I; z, \omega, \mathcal{R}) \equiv \text{Obs}_{\mathcal{H}}(I^{-U}; z, \omega, \mathcal{R})$$

for every $z \in \text{supp}(\mathcal{D}_{\text{hon}})$, every coin string ω , and every oracle table $\mathcal{R}$. Consequently the one-run observation distributions are identical:

$$\text{Obs}_{\mathcal{H}}(I) \equiv \text{Obs}_{\mathcal{H}}(I^{-U}).$$

Proof. Construct I^{-U} by replacing every operation in U by a fixed dummy value of the correct syntactic type and then executing the remaining code. For fixed $(z, \omega, \mathcal{R})$, the trace of I^{-U} is exactly an intervened trace $\text{Trace}_I(z, \omega, \mathcal{R})[U \leftarrow a_U]$ for the dummy vector a_U . By the inactivity hypothesis, the value of $\text{Tgt}_{\text{cc}, \mathcal{H}}$ is unchanged. Since $\mathcal{H}$ is black-box over honest support, its complete observation factors through this target vector. Hence the observations are identical pointwise for the coupled run. Averaging over z , coins, and the oracle table proves equality of observation distributions. $\square$

Theorem 5.37 (Dependency-cone non-certification lower bound). *Let $\mathcal{H}_{\text{hon}}$ be any black-box honest-reference harness over honest ciphertext support, and let $U \subseteq \mathcal{O}$ be inactive for the confirmation-observable target:*

$$U \perp_{\mathcal{D}_{\text{hon}}} \text{Tgt}_{\text{cc}, \mathcal{H}}.$$

Let C be an implementation class that contains an implementation I_1 with $\text{Exec}_U(I_1) = 1$ and a coupled U -erasure implementation $I_0 \in C$ with $\text{Exec}_U(I_0) = 0$ and the same black-box transcript distribution. Then every (α, β) -sound and complete black-box certifier for U -execution over C must satisfy

$$\alpha + \beta \geq 1.$$

In particular, the same conclusion holds for every nonempty U -erasure-closed class C containing an implementation that executes U .

More generally, if the erasure construction only achieves

$$\Delta_{\text{TV}}(\text{Obs}_{\mathcal{H}}(I), \text{Obs}_{\mathcal{H}}(I^{-U})) \leq \Delta$$

per run for a pair $I, I^{-U} \in C$ with $\text{Exec}_U(I) = 1$ and $\text{Exec}_U(I^{-U}) = 0$, then every n -sample certifier over C must satisfy

$$\alpha + \beta + n\Delta \geq 1.$$

Proof. For the exact statement, the observations of $\mathcal{I}_1$ and $\mathcal{I}_0$ have identical one-run distributions. Hence their n -sample product distributions are also identical. Let p be the probability that Cert outputs 1 under this common distribution. Completeness on $\mathcal{I}_1$ implies $p \geq 1 - \beta$, while soundness on $\mathcal{I}_0$ implies $p \leq \alpha$. Therefore $1 - \beta \leq \alpha$, equivalently $\alpha + \beta \geq 1$. Thus no certifier with $\alpha + \beta < 1$ exists.

For the approximate statement, total variation distance between n -fold product distributions is at most $n\Delta$. Therefore the certifier's output probabilities on the executing implementation $\mathcal{I}$ and the erasure implementation $\mathcal{I}^{-U}$ differ by at most $n\Delta$. Completeness and soundness give

$$1 - \beta \leq \Pr[\text{Cert} = 1 \mid \mathcal{I}] \leq \Pr[\text{Cert} = 1 \mid \mathcal{I}^{-U}] + n\Delta \leq \alpha + n\Delta.$$

Rearranging proves $\alpha + \beta + n\Delta \geq 1$. □

Corollary 5.38 (Cone boundary for code claims). *An honest-reference confirmation-code test can support an operation-execution claim only for operations in the support-active confirmation cone $\text{Cone}_{\text{cc}, \mathcal{H}}$, or for operations exposed by additional harness observations, instrumentation, malformed support, or differential support. For operations outside that cone, the strongest correct statement over erasure-closed classes, or over any class containing the coupled erasure, is a non-certification statement of Theorem 5.37.*

Proof. If U is outside the cone, then it is inactive for $\text{Tgt}_{\text{cc}, \mathcal{H}}$, and Theorem 5.37 rules out nontrivial black-box certification for any implementation class that contains the coupled erasure. If a test samples a different support, logs additional internal data, or adds a code source that depends on U , then the target vector and its cone have changed; the theorem no longer classifies U as outside the cone. □

Remark 5.39 (Indistinguishability from cone inactivity). *The only structural hypothesis is black-box factorization: the harness observation must depend on the implementation trace only through declared inputs, declared oracle-query logs, declared outputs, and the final-key test. That is the definition of the black-box honest-reference setting. The theorem constructs a U -erasure with the same transcript distribution whenever U is outside the support-active cone. The certification lower bound is a statement about implementation classes that include that erasure, because soundness is always relative to the class on which a certifier is specified.*

Definition 5.40 (Cone-inactivity certificate). *A cone-inactivity certificate for an artifact row consists of:*

1. *the omitted operation set U ;*
2. *the exact target vector components being checked;*
3. *the support condition, such as honest ciphertext support or a named malformed/differential support;*
4. *a trace argument proving*

$$U \perp_{\mathcal{D}} \text{Tgt}$$

for the whole set U ; and

5. *the implementation class over which the coupled erasure is included.*

Rows without such a certificate remain finite-catalog observations or source-code-cone boundaries, depending on the theorem invoked.

Row family	Omitted set U	Target components and support	Certificate status
HQC-CD1 non-retained tail bits	tail coordinates outside the retained CD1 projection	$B, W, x^\star$, declared final-key inputs, public outputs, and the support-constant branch selector on honest ciphertext support	set-level certificate when the black-box target logs only the retained tail projection; invalid if a harness exposes the full tail, a full comparison trace, or instrumentation state
ML-KEM unselected source coefficients	coefficients outside the selected source set S	source-code target defined by the selected coefficient serialization	certificate for the selected source-code cone only; a full $\text{Tgt}_{\text{cc}, \mathcal{H}}$ certificate must additionally exclude ordinary comparison outputs, declared query logs, and branch behavior
Always-select- K_+ on honest ciphertexts	decision node selecting between K_+ and K_-	honest ciphertext support, after quotienting support-constant predicates	certificate only on honest support; malformed or differential support changes the target and can make the branch observable
Component-selective comparison rows	comparison component not sampled by the row's retained code or mode	the named mode and honest-support target for that row	row-specific certificate required; singleton non-membership is insufficient for a set of comparisons unless simultaneous inactivity is proved

Table 3: Cone-inactivity certificates used to interpret negative rows. A row becomes theorem-covered non-certification only for the stated target, support, and erasure-containing implementation class.

5.8 Coverage supplied by the code source

Definition 5.41 (Trace obligations and code coverage). *Let a decapsulation trace be a labeled DAG*

$$\text{Trace} = (O, \prec, \text{val}),$$

where O is the set of operations, $\prec$ is the data-dependency relation, and $\text{val}(o)$ is the value produced by operation o . Let $W = \text{Code}(\text{Trace})$. For $U \subseteq O$, let $\pi_{\overline{U}}(\text{Trace})$ be the trace with all values produced only by operations in U removed.

We say U is (λ, ϵ) -covered by the code against a SUT class C and harness $\mathcal{H}$ if, for every $S \in C$ that omits all operations in U , there is a view V_U containing the SUT transcript such that $Q_G(B)$ is V_U -measurable and W has (λ, ϵ) -smooth average conditional min-entropy given V_U .

Theorem 5.42 (Covered omitted obligations are detected). *If U is (λ, ϵ) -covered by the confirmation code against $(C, \mathcal{H})$, then every q_G -localized SUT $S \in C$ omitting U satisfies*

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(S) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(S, \mathcal{H}) + \epsilon_{\text{fresh}}(S, \mathcal{H}) + \epsilon + q_G 2^{-\lambda} + 2^{-\kappa}.$$

Proof. By coverage, the omitted operations leave W with smooth average conditional min-entropy at least λ given the SUT view. Apply Theorem 5.14. $\square$

Proposition 5.43 (Factorization is the one-operation cone case). *Suppose*

$$W = f(\pi_{\{o\}}(\text{Trace}))$$

for some operation o , and suppose o also has no active path on honest support to any other value in $\text{Tgt}_{\text{cc}, \mathcal{H}}$. Then $o \notin \text{Cone}_{\text{cc}, \mathcal{H}}$, and no black-box honest-reference certifier over an o -erasure-closed class can certify execution of o with $\alpha + \beta < 1$.

Proof. The factorization says that intervening on o cannot change the witness W . The additional no-active-path condition says that intervening on o cannot change the other components of the confirmation-observable target vector either. Thus

$$\{o\} \perp_{\mathcal{D}_{\text{hon}}} \text{Tgt}_{\text{cc}, \mathcal{H}}.$$

Apply Theorem 5.37. □

Corollary 5.44 (No whole-reencryption conclusion without a cone certificate). *Let O_{reenc} be the set of operations in the reencryption path. An honest-reference confirmation-code test certifies only those operations in*

$$O_{\text{reenc}} \cap \text{Cone}_{\text{cc}, \mathcal{H}}$$

for which a positive list-hit, cUP, or entropy bound is supplied. For every operation

$$o \in O_{\text{reenc}} \setminus \text{Cone}_{\text{cc}, \mathcal{H}},$$

there exists an o -erasure implementation with identical black-box honest-reference transcript distribution on honest support. Hence no nontrivial black-box certificate of whole-reencryption execution follows from that harness alone over erasure-closed classes, or over any class containing the corresponding erasure implementation.

Proof. The positive part is Theorem 5.42 or the corresponding cUP/entropy bound applied to the operations inside the cone. The negative part is Theorem 5.37 applied to each operation outside the support-active confirmation cone. □

Lemma 5.45 (ML-KEM code-source coverage). *In the source ML-KEM confirmation-code line of [13], the code is made of selected uncompressed ciphertext coefficients from positions $S \subseteq \{1, \dots, 256\}$ in each recomputed u_i component and in the recomputed v component. For $|S| = 2$, this selects $2(k+1)$ coefficients, serialized as 12, 16, and 20 bytes for ML-KEM-512, ML-KEM-768, and ML-KEM-1024. Computing that source code through the specified procedure requires computing those selected uncompressed reencryption coefficients.*

The artifact ML-KEM-CD1 line uses a smaller diagnostic hashed code: one byte obtained by hashing 32 residue bytes from selected recomputed u - and v -component coefficients. Its formal interpretation is Theorem 7.2, a specialization of Theorem 5.17 under the named raw-residue entropy assumption. It is not covered by the source-paper ML-KEM cUP theorem.

Lemma 5.46 (HQC code-source coverage). *In the August 2025 HQC CD1 and CD* variants, the confirmation code is read from the residual v -path bits immediately before `vect_truncate`. Computing CD* through the specified procedure requires computing the retained tail bits of that pre-truncation v -buffer; computing CD1 requires the first $\min(\ell_{\text{tail}}, 8)$ retained bits. Full ciphertext comparison and other internal reencryption values require separate harness or code-source evidence.*

5.9 Source security and ROM/QROM scope

The detectability theorems proved here are classical ROM testing statements for classical SUTs with classical oracle queries, with construction-level security of the confirmation-code-augmented KEM line cited from the source verifiable-decapsulation and FO literature.

In the source notation, cUP is the game for predicting the confirmation code under restricted access to the internal confirmation function, and fCOR is the game in which a faulty-but-benign decapsulation outputs the honest encapsulated key. The source theorem states a bound of the form

$$\text{Adv}^{\text{fCOR}} \leq \delta + \text{Adv}^{\text{cUP}} + 2^{-\kappa}$$

Source game or term	Standard role	Use in this paper
cUP	unpredictability of the confirmation code for a faulty implementation with restricted confirmation-function access	single-candidate primitive lifted to list-cUP
list-cUP	probability that a candidate list contains the hidden witness	central hit term for honest-reference testing
fCOR	faulty implementation outputs the honest encapsulated key	matches honest-reference pass probability when interfaces coincide
IND-CCA	ordinary active security of the KEM	inherited from source augmented FO transforms, not reproved here
FFP and spreadness	bad events for simulating decapsulation in FO/QROM proofs	source security provenance
explicit/implicit rejection	invalid-ciphertext behavior of FO-KEM decapsulation	explains why branch-selection faults need their own harness support

Table 4: Security vocabulary used in the paper. Construction-level KEM security statements are cited from the source games.

for the confirmation-code-augmented transforms under the modeled final-key derivation oracle function [13, Theorem 16]. Our contribution is the harness-indexed list version: a concrete test induces a candidate list $Q_G(B)$, and the source cUP term controls that list through Theorem 5.10 when the simulation interface matches.

The classical query $\log Q_G(B)$ and the event that the hidden point $x^\star$ is queried are part of the testing model. A quantum SUT with superposition access has no such classical list; a QROM analogue would require a different theorem, replacing $\Pr[\text{Hit}]$ by an appropriate query-amplitude or one-way-to-hiding term for $x^\star$. References to QROM results give construction-level provenance for the underlying augmented FO KEM, not testing theorems for the harness model [13, Theorems 26, 27 and 30].

6 Standard Lines, Augmented Variants, and Mutants

The theorem applies to confirmation-code-augmented KEM variants, while the unmodified standardized KEM, the augmented variant, and the mutant tested against that variant carry different claims, and the artifact separates these three objects throughout the experiments.

Definition 6.1 (Output compatibility with a standard KEM). *An augmented line Π^C is output-compatible with a standard KEM Π if, for every standard key pair and every honestly generated standard ciphertext, the good-branch shared secret output by Π^C is identical to the shared secret output by Π , except with the stated correctness error.*

Proposition 6.2 (Confirmation-code binding is not output-compatible). *Let the standard good key be derived at a domain-separated oracle point x_{std} , and let the augmented good key be derived at $x^\star = \langle \text{good}, B, W \rangle$. If $x_{\text{std}} \neq x^\star$ except with aliasing probability ϵ_{alias} , then the augmented line is not output-compatible with the standard line. More precisely, on honest ciphertexts,*

$$\Pr[K_{\text{std}} = K_{\text{aug}}] \leq \epsilon_{\text{alias}} + 2^{-\kappa}.$$

Proof. Outside the aliasing event, the standard and augmented keys are outputs of the random oracle on distinct domain-separated inputs. They are therefore independent uniform κ -bit strings, and collide with probability $2^{-\kappa}$. Adding the aliasing event proves the bound. $\square$

Remark 6.3 (Deployment reading). *The augmented variants in this paper are test-mode or derived research lines. They are not drop-in output-compatible modes for FIPS 203 ML-KEM or for the public HQC*

Object	Example in the artifact	Claim made in this paper
Standard line	FIPS 203 ML-KEM and the public August 2025 HQC line	baseline implementation and source of the decapsulation algorithms
Confirmation-code-augmented variant	VSFO/CD1, VSFO/CD*, and the source-paper ML-KEM confirmation-code line	object to which the list-hit/list-cUP testing theorem applies
Artifact mutant	drop cd , overwrite cd , always select K_+ , compare only one component, symmetric endpoint mutation	tested faulty implementation under a named harness

Table 5: Three layers used in the paper. The augmented variants inherit algorithmic structure and formats from the baseline lines, but their shared secrets include confirmation-code material and are not output-compatible with the unmodified standards.

line. The standard implementations serve as baselines and sources of the decapsulation algorithms; the confirmation-code variants create a separate test signal by changing the good-branch key input.

FIPS 203 defines the standardized ML-KEM behavior, and the artifact VSFO/CD1 line inherits the ciphertext format of mlkem-native while changing the good-branch shared secret by hashing the recomputed diagnostic code with the ordinary key material. It is a test variant over FIPS 203 algorithms, not a FIPS 203-compatible output mode. The same distinction applies to the HQC augmented lines: the baseline is the public August 2025 specification, while CD1 and CD* are confirmation-code-augmented variants used to test the code-binding mechanism.

The standard lines appear in the experiments as baselines for support and branch behavior, and some representative reencryption omissions are already visible under honest-reference testing on those baselines. The positive theorem concerns the additional localized final-key signal introduced by binding a hidden confirmation witness into the good key of the augmented variant.

7 Instantiations

7.1 ML-KEM

The ML-KEM source construction is the confirmation-code line of [13], where the confirmation code is derived from selected recomputed uncompressed ciphertext coefficients and included in final key derivation. If the source cUP bound for the selected code is $\epsilon_{\text{cUP}}^{\text{ML-KEM}}$, and if the harness and SUT class satisfy the cUP-faithful certificate of Definition 5.6, then the harness list-cUP theorem yields, for a q_G -localized SUT in that simulated class,

$$\text{Pass}_{\Pi^C, \mathcal{H}}^{\text{href}}(\mathcal{S}) \leq \delta_{\Pi^C} + \epsilon_{\text{alias}}(\mathcal{S}, \mathcal{H}) + \epsilon_{\text{fresh}}(\mathcal{S}, \mathcal{H}) + q_G \epsilon_{\text{cUP}}^{\text{ML-KEM}} + 2^{-\kappa}.$$

For the selected-code line, the transfer requires both the source cUP theorem and the cUP-faithful harness certificate. SUT classes that can compute W through paths outside the source cUP interface fall outside this source-security route.

The source paper proposes short ML-KEM codes by selecting $|S| = 2$ coefficients from each recomputed ciphertext component. With the usual ML-KEM module ranks $k = 2, 3, 4$, this selects 12, 16, and 20 serialized code bytes for ML-KEM-512, ML-KEM-768, and ML-KEM-1024. Those source codes are the faithful ML-KEM instantiation of the cUP analysis cited above.

Cone accounting for the source ML-KEM code. The dependency-cone theorem assigns an exact interpretation to the source coefficient selection. With $|S| = 2$, the source code reads two selected

uncompressed coefficients from each recomputed ciphertext component $u_1, \dots, u_k, v$. The source-code cone therefore contains $2(k + 1)$ selected coefficient nodes. The full recomputed uncompressed u -and- v coefficient array has $256(k + 1)$ coefficient nodes, so $254(k + 1)$ coefficient nodes are outside this particular selected source-code cone. Membership outside the full $\text{Tgt}_{\text{cc}, \mathcal{H}}$ cone requires a separate target certificate: ordinary comparison outputs, declared query logs, public outputs, or branch behavior may expose those nodes under a larger harness target.

Parameter set	k	Total u/v coeffs	Selected coeffs	Outside source-code cone
ML-KEM-512	2	$256 \cdot 3 = 768$	$2 \cdot 3 = 6$	762
ML-KEM-768	3	$256 \cdot 4 = 1024$	$2 \cdot 4 = 8$	1016
ML-KEM-1024	4	$256 \cdot 5 = 1280$	$2 \cdot 5 = 10$	1270

Table 6: Dependency-cone accounting for the source ML-KEM confirmation code with $|S| = 2$. The selected coefficients are the code-source cone. The outside column is the boundary of this selected source-code target. A full black-box non-certification claim additionally requires a cone-inactivity certificate for $\text{Tgt}_{\text{cc}, \mathcal{H}}$, as in Definition 5.40.

Starting from the pinned mlkem-native v1.0.0 baseline used in the experiments, the artifact patch materializes a VSFO/CD1 variant. This derived line is a one-byte diagnostic hashed-code variant over the FIPS 203 algorithms, distinct from the full source-paper short-code construction and from a FIPS 203 output-compatible KEM. Its proof obligation is therefore Theorem 7.2, which specializes the hashed-witness theorem under the named ML-KEM-CD1 raw-residue entropy assumption. The source-paper ML-KEM cUP theorem does not cover this diagnostic line. The ciphertext format is inherited from the baseline, while the shared secret is derived from the recomputed diagnostic code as well as the ordinary ML-KEM key material.

The patched encapsulation computes the usual

$$\text{buf} = m \| H(pk), \quad kr = G(\text{buf}),$$

uses the second 32-byte half of kr as the IND-CPA coins, and calls `mlk_indcpa_enc_cd`. That routine returns the ciphertext and a one-byte confirmation code. The code is the first byte of ML-KEM’s hash H applied to a 32-byte witness: the first 16 bytes are compression residues from the first 16 coefficients of the recomputed u -component, and the last 16 bytes are compression residues from the first 16 coefficients of the recomputed v -component. The artifact good key is computed as $H(kr \| cd)$. Decapsulation decrypts, recomputes the same ciphertext and code, compares the ciphertext in the ordinary way, computes the same $H(kr \| cd)$ value for the good branch, and leaves the rejection branch unchanged.

Definition 7.1 (ML-KEM-CD1 raw-residue entropy assumption). *Let $R_{\text{ML}} \in \{0, 1\}^{256}$ be the 32-byte residue witness used by the artifact ML-KEM-CD1 line: the first 16 bytes are the selected recomputed u -component compression residues and the last 16 bytes are the selected recomputed v -component compression residues. Let*

$$W = \text{Trunc}_8(H_{\text{cd}}(\langle \text{cd}, B, R_{\text{ML}} \rangle)).$$

For a harness class $\mathfrak{H}$, SUT class C , and parameters $(\rho_{\text{ML}}, \epsilon_{\text{ML}})$, the assumption

$$\text{MLKEM-CD1-RawEnt}_{\rho_{\text{ML}}, \epsilon_{\text{ML}}}^{\mathfrak{H}, C}$$

states that, in the canonical honest-reference ML-KEM-CD1 experiment, after exposing the pre- H_{cd} -query SUT view V_0 but excluding R_{ML} , W , K , x^ , and the private oracle value $H_{\text{cd}}(\langle \text{cd}, B, R_{\text{ML}} \rangle)$, one has*

$$P_{\text{guess}}(R_{\text{ML}} \mid V_0) \leq 2^{-\rho_{\text{ML}}} + \epsilon_{\text{ML}}.$$

ML-KEM line	Confirmation code	Role in this paper
FIPS 203 ML-KEM source ML-KEM ^C [S]	none selected uncompressed u_i and v coefficients; 12/16/20 bytes for $ S = 2$	standards baseline and source of the unmodified algorithms faithful source-paper instantiation covered by the cited cUP analysis
artifact VSFO/CD1	one byte from a hash of 32 recomputed residue bytes	diagnostic non-standard variant governed by Theorem 7.2

Table 7: ML-KEM layers used in the paper. Only the augmented lines bind a confirmation code into the good key; the artifact line uses a smaller code than the source-paper short code.

Object	ML-KEM reading	Testing role
B	kr and the final-key domain material used by the patched good branch	localized prefix for the final key query
W in ML-KEM ^C [S]	selected recomputed uncompressed ciphertext coefficients	source confirmation code controlled by the cited cUP theorem
W in artifact CD1	one-byte hash of the recomputed compression-residue witness	diagnostic hidden code value interpreted through hashed-witness detectability
binding mutant	remove or overwrite the confirmation code before the final key derivation	direct localized-list-hit fault
decision mutant	accept on a branch condition that is unchanged on honest ciphertexts	support-coincidence example
symmetric endpoint mutant	remove the code from both encapsulation and decapsulation	self-vs-reference example

Table 8: ML-KEM interpretation of the abstract quantities.

The RawEnt assumption names a hypothesis about the SUT class and harness view; without that hypothesis, the artifact rows for ML-KEM-CD1 are finite-catalog diagnostic observations.

Theorem 7.2 (ML-KEM-CD1 diagnostic detectability). *Assume*

$$\text{MLKEM-CD1-RawEnt}_{\rho_{\text{ML}}, \epsilon_{\text{ML}}}^{\mathfrak{S}, C}.$$

For every SUT $S \in \mathcal{C}$ making at most q_H raw H_{cd} -queries and at most q_G localized final-key suffix queries,

$$\text{Pass}_{\text{MLKEM-CD1}, \mathcal{H}}^{\text{href}}(S) \leq \delta_{\text{MLKEM-CD1}} + \epsilon_{\text{alias}}(S, \mathcal{H}) + \epsilon_{\text{fresh}}(S, \mathcal{H}) + q_H(2^{-\rho_{\text{ML}}} + \epsilon_{\text{ML}}) + q_G 2^{-8} + 2^{-\kappa}.$$

Proof. The raw-query list has size at most q_H . The raw-residue entropy assumption and the list-hit lemma imply that the probability of querying the exact raw witness R_{ML} to H_{cd} is at most $q_H(2^{-\rho_{\text{ML}}} + \epsilon_{\text{ML}})$. Conditioned on not making that raw query, the value $H_{\text{cd}}(\langle \text{cd}, B, R_{\text{ML}} \rangle)$ is fresh and its first byte is uniform from the SUT view. A q_G -localized final-key suffix list then hits W with probability at most $q_G 2^{-8}$. Apply Theorem 5.1. $\square$

The ML-KEM experiments compare plain faults, transformed diagnostic binding faults, decision-only faults, and symmetric endpoint faults on the one-byte code-bearing artifact variant named CD1. That variant is a small, reproducible harness object for the theorem-relevant fault families, with a diagnostic security reading separate from the source confirmation-code theorem.

7.2 August 2025 HQC

For HQC, we use the public specification dated 2025-08-22 as the main public baseline line [8]. That line uses the salted implicit-rejection FO transform analyzed for multi-target security in [12]: the ciphertext

Step	August 2025 HQC baseline	Augmented HQC variant used here
Encaps randomness and key seed	$(K, \theta) \leftarrow G(H(ek_{\text{KEM}}) m \text{salt})$	derive the same reencryption transcript, with G_θ and G_K modeled as separated domains
PKE encryption	$c_{\text{PKE}} \leftarrow \text{HQC-PKE.Enc}(ek_{\text{KEM}}, m; \theta)$	$(c_{\text{PKE}}, cd) \leftarrow \text{HQC-PKE.EncC}(ek_{\text{KEM}}, m; \theta)$
Good key	K from the baseline derivation	$K_+ \leftarrow G_K(\langle \text{good}, H(ek_{\text{KEM}}) m \text{salt}, cd \rangle)$
Decaps recomputation	recompute m' , θ' , and c'_{PKE} , then compare	recompute m' , θ' , c'_{PKE} , and cd' , then bind cd' into K'_+
Ciphertext format	$(c_{\text{PKE}}, \text{salt})$	unchanged $(c_{\text{PKE}}, \text{salt})$
Shared-secret output	baseline shared secret	changed by confirmation-code binding, hence non-output-compatible with the baseline

Table 9: Pseudo-diff for the August 2025 public HQC line and the confirmation-code-augmented variants used in the artifact.

is salted, the encapsulation key is hashed into the derivations, and the derivation transcript used by the transform is

$$H(ek_{\text{KEM}})||m||\text{salt}.$$

In our augmented variants, the diagnostic confirmation code comes from the truncation tail of the v -path. The August 2025 salted-SFO line yields a direct range statement for that tail, formalized in Proposition 5.20. Tail-code detectability requires HQC-TailEnt, as stated in Theorem 5.22, or another explicit lower bound for the witness given the SUT view. We consider two tail-derived variants:

- CD1 retains one byte, truncated to the available tail;
- CD* retains the full truncation-tail range budget.

The artifact patch computes $\theta = G_\theta(H(ek_{\text{KEM}})||m||\text{salt})$ with the HQC domain byte for G_θ . The modified PKE encryption routine `hqc_pke_encrypt_cd` extracts the confirmation code from the residual bits of the v -path immediately before `vect_truncate`. For bit i of the retained code, the source bit is `PARAM_N1N2 + i` in the pre-truncation v -buffer, subject to the configured bit limit and output byte length. The good key is computed by `hash_k_confirmation`, which absorbs $H(ek_{\text{KEM}})$, m , salt, the retained confirmation code, and the HQC domain byte for G_K . Decapsulation decrypts, recomputes θ , reruns the same code-producing encryption routine, compares the PKE ciphertext, and uses the same code-bearing good-key hash for the good branch.

The earlier verifiable-decapsulation discussion of HQC uses its own confirmation-code analysis and independence assumptions for that setting. For the August 2025 salted transcript, we prove the retained range budget and state the conditional tail-entropy assumption needed to turn that range into a detectability bound.

August 2025 as anchor. During 2025, the public HQC line changed in ways that affect the baseline for the augmented variants. The February line is useful historical context, including for the implementation flaw that motivated part of this work, but the August public specification is the relevant line for the August 2025 HQC experiments. The salted transcript, the implicit-rejection flow, and the key layout are part of the baseline from which the augmented variants are derived.

Cone accounting for HQC tail variants. For HQC, the retained code cone is exactly the retained truncation-tail slice. On the August 2025 salted-SFO line, $\ell_{\text{tail}} = 5, 11, 37$ for HQC-1/3/5. The CD1 line retains 5, 8, 8 bits, so it leaves 0, 3, 29 tail bits outside its retained tail-code cone. The CD* line retains the full tail range and therefore leaves no tail bit outside the tail-code cone, although operations outside the tail path or outside the honest-support target remain subject to Theorem 5.37.

Parameter set	ℓ_{tail}	CD1 range bits	CD* range bits
HQC-1	5	5	5
HQC-3	11	8	11
HQC-5	37	8	37

Table 10: August 2025 HQC tail-derived range budgets. ℓ_{tail} is the number of residual truncation-tail bits in the code source. These are range bounds; conditional min-entropy is the separate hypothesis used in Theorem 5.22.

HQC aspect	February 2025 line	August 2025 line used here
Transform shape	earlier salted/HHK-style presentation	public salted implicit-rejection flow with explicit $H(ek_{\text{KEM}}) m \text{salt transcript}$
Public-key binding	narrower historical binding surface	all bytes of ek_{KEM} enter the key and randomness derivation through the public hash
Secret-key layout	historical appended layout	August 2025 public layout and compressed alternatives
Code source	truncation-tail opportunity present	truncation-tail opportunity still present and parameterized
Role in artifact	drift comparison	main public baseline

Table 12: Condensed view of the HQC drift relevant to this paper.

Parameter set	ℓ_{tail}	CD1 retained	outside CD1 tail cone	CD* retained	outside CD* tail cone
HQC-1	5	5	0	5	0
HQC-3	11	8	3	11	0
HQC-5	37	8	29	37	0

Table 11: Dependency-cone accounting for August 2025 HQC tail-derived codes. These are cone and range counts, not entropy claims. Tail-code detectability still requires HQC-TailEnt or another unpredictability bound for the retained bits, while non-retained tail bits give theorem-covered non-certification only when the row includes a cone-inactivity certificate for the black-box target, as in Table 3.

7.3 Instantiation Contributions

ML-KEM links the testing statement to a source confirmation-code construction derived from a standardized KEM and also supplies a small diagnostic hashed-code artifact line. The August 2025 HQC line has a public salted flow with a tail-code family whose retained range can be swept experimentally. The main theorem applies to augmented variants once the list-hit term is bounded by source cUP, by min-entropy, or by an explicit diagnostic assumption. The standard lines provide baselines and implementation context, and the HQC CD1 versus CD* comparison varies retained range and measured witness-guess behavior while keeping the KEM design goal fixed.

Object	August 2025 HQC reading	Testing role
B	$H(ek_{\text{KEM}}) m \text{salt}$	localized final-key prefix fixed by the salted reference trace
θ	$G_\theta(B)$	reencryption randomness that must be recomputed
W	retained truncation-tail code cd	hidden code entering the final good-key query
CD1	one-byte retained code, truncated to the available tail on HQC-1	small August 2025-line variant with a range budget; detectability requires TailEnt or another entropy claim
CD*	full retained truncation-tail code	full-range variant for the same family; range is not unpredictability

Table 13: August 2025 HQC interpretation of the abstract quantities.

Family	Hidden code value	Main role in this paper
ML-KEM	source confirmation code from the verifiable-decapsulation line; one-byte hashed diagnostic code in the artifact	source-paper instantiation plus diagnostic artifact variant over FIPS 203 algorithms
August 2025 HQC	truncation-tail diagnostic code in the salted public line	salted public baseline plus range-budget diagnostic variants under explicit TailEnt assumptions when used for detectability

Table 14: The two concrete instantiations used in the paper.

Artifact line	Final-key length κ	Reading
ML-KEM-CD1	256 bits	SHA3/SHAKE-derived 32-byte shared secret in the diagnostic augmented line
HQC-CD1	256 bits	32-byte shared secret in the augmented salted line
HQC-CD*	256 bits	same final-key length, with a larger retained code input

Table 15: Final-key length used for the $2^{-\kappa}$ term in the artifact variants. The term is a random-oracle-model final-key coincidence term.

8 Artifact

FO-FaultBench packages the experiment code, patch lines, mutant catalogs, frozen JSONL records, three derived summaries, and two plots for the confirmation-code decapsulation experiments. The package separates a read-only validation path from a rebuild path: `make check` validates the stored layout and parses the JSONL records, `make summary` prints a compact inventory, and the `fetch/build/run` targets acquire pinned sources and rerun the campaigns.

Baselines. The main baselines are `mlkem-native v1.0.0` and the August 2025 public HQC reference line. The HQC baseline is built with an explicit portability patch on the development host. The derived lines are materialized from local patches: `mlkem-vsfo-cd1`, `hqc-vsfo-cd1`, and `hqc-vsfo-cdstar`. The historical February 2025 HQC line is retained for standards-drift comparison.

Harnesses. The artifact implements $D_{\text{honest_ref}}$, $D_{\text{honest_self}}$, D_{ctmal} , D_{diff} , and D_{keyfmt} , using the same harness names as the theorem statements; every “detected” or “undetected” row is relative to the stated harness.

Mutants. The mutant catalogs contain deterministic recomputation and binding faults, decision-only faults, symmetric endpoint faults, probabilistic code-guessing faults, and the HQC range-budget sweep. Each catalog entry records the fault class, affected operation, expected harness behavior, and patch.

Reporting. Deterministic tables report pass/fail counts over the frozen seeds and parameter sets. Probabilistic rows report observed pass probability and a 95% interval or upper bound. For a fixed line–fault–harness row with x passes in n seeded harness runs, the point estimate is $\hat{p} = x/n$. Nonzero rows use the Wilson score interval with $z_{0.975} = 1.9599639845$,

$$\frac{\hat{p} + z^2/(2n) \pm z\sqrt{\hat{p}(1-\hat{p})/n + z^2/(4n^2)}}{1 + z^2/n}.$$

Zero-pass rows use the upper endpoint $1 - (0.025)^{1/n}$ of the two-sided 95% Clopper–Pearson interval. Repeated-test survival columns are computed as $\hat{p}^t$ under the reset-product interpretation of Theorem 5.29, using the zero-pass upper endpoint where applicable.

Reproducibility. The packaged snapshot contains eleven raw JSONL files, three Markdown summaries, two SVG plots, and a hash file over the stored package inputs. The read-only path uses `make check` and `make summary`; it validates the expected directories and files, parses every stored JSONL record, and reports the inventory without rewriting the stored results. The current snapshot contains 3592 raw records. The rebuild path uses `make fetch`, `make verify-upstreams`, `make build-baselines`, `make prepare-variants`, `make run-core`, and `make run-mutants`; it fetches pinned upstreams, verifies them against `artifact/baselines.json`, builds the baseline and derived implementations, and reruns the harness families. Reruns may create build products and append new JSONL records under `results/raw/`.

Each source-security row in the artifact records the interface classification used to interpret the corresponding harness result. Rows using the source ML-KEM confirmation code are eligible for cUP-faithful certification, whereas rows using CD1 or CD* diagnostic codes are interpreted through the diagnostic or TailEnt route.

Campaign	Main harnesses	Stored records
deterministic recomputation and binding	$D_{\text{honest_ref}}, D_{\text{ctmal}}, D_{\text{diff}}$	raw mutant and differential JSONL records, with the compact mutant summary
decision-only and comparison faults	$D_{\text{ctmal}}^\mu, D_{\text{diff}}^\mu, D_{\text{honest_ref}}$	raw mutant and differential JSONL records
symmetric endpoint faults	$D_{\text{honest_ref}}, D_{\text{honest_self}}$	raw honest-reference, honest-self, and mutant JSONL records
probabilistic code guessing	$D_{\text{honest_ref}}$	raw probabilistic-witness JSONL records and compact probability summary
HQC range-budget sweep	$D_{\text{honest_ref}}$	raw sweep JSONL records, compact sweep summary, and frozen plot
HQC standards drift	baseline/spec comparison	pinned baseline metadata and raw comparison records

Table 16: Main artifact campaigns.

Line	Status in package	Role
mlkem-native-v1.0.0	pinned, built, smoke-validated	plain ML-KEM baseline
mlkem-vsfo-cd1	derived by patch and validated	transformed ML-KEM code line
hqc-current-ref	pinned August 2025 public line with portability patch	August 2025 HQC baseline
hqc-vsfo-cd1	derived by patch and validated	August 2025 HQC one-byte code line
hqc-vsfo-cdstar	same derived line with full code retention	August 2025 HQC full-budget code line
hqc-historical-fb-2025-additional-128	acquired for comparison	standards-drift context

Table 17: Baseline and derived-line inventory used by the experiments.

Definition 8.1 (Artifact ROM-interface certificate). *An artifact row has a ROM-interface certificate if the row records the following structural correspondence between the concrete harness and the oracle model.*

- (R1) *Each concrete hash, XOF, or KDF call used as a modeled oracle call is assigned to a length-delimited, domain-separated namespace. Calls modeled as G_K , H_{cd} , or another oracle either use distinct namespaces or use an injective domain tag inside a single concrete primitive.*
- (R2) *The final-key point $x^\star = \langle \text{good}, B, W \rangle$ and the projection $Q_G(B)$ use the same serialization as the concrete harness.*
- (R3) *All SUT calls to the modeled final-key namespace are logged before projection to $Q_G(B)$, and the SUT has no access to a shared oracle table or unlogged cache for $G_K(x^\star)$.*
- (R4) *The SUT input excludes W , K , $x^\star$, the private reference evaluation $G_K(x^\star)$, deterministic witness-sampling seeds, and debug or trace data containing witness bytes.*
- (R5) *Any disabled item, noncanonical serialization, shared cache, or implementation leak affecting the hidden witness or final key is reported as an explicit aliasing or freshness defect for that row.*

Proposition 8.2 (Structural artifact-to-ROM correspondence). *If an artifact row satisfies Definition 8.1, then in the ROM interpretation induced by that certificate the logged localized final-key queries are exactly the modeled list $Q_G(B)$, and the row has $\epsilon_{\text{fresh}} = 0$ except for explicitly reported defects. The claim is a*

structural correspondence between the artifact interface and the ROM model; random-oracle ideality for the concrete SHA3/SHAKE primitives is the usual modeling assumption, not an empirical conclusion of the artifact.

Proof. By (R1) and (R2), concrete final-key calls have a unique modeled namespace and canonical projection to B -localized candidates. By (R3), every SUT call to that namespace appears in the logged list, and no shared table or cache exposes the SUT the private reference value without a logged query. By (R4), the SUT view omits the hidden witness, final key, final-key point, and witness-bearing sampling or debug data. Thus the row has no unreported path that increases the posterior guessing probability of $K = G_K(x^*)$ after conditioning on no list hit. Any violation is charged by (R5) as aliasing or freshness defect. $\square$

9 Experimental Results

Definition 9.1 (Artifact claim levels). *An experimental row has one of three claim levels.*

- (A1) **Finite-catalog claim.** *The row concerns exactly the listed mutants, seeds, parameter sets, and harnesses.*
- (A2) **Theorem-covered class claim.** *The row is paired with a formal class hypothesis such as cUP-faithfulness, conditional min-entropy, raw-witness entropy, TailEnt, support equivalence, or code-source coverage.*
- (A3) **No class claim.** *If neither (A1) nor (A2) is stated, the row is not evidence for unenumerated faults.*

Proposition 9.2 (Finite-catalog soundness of artifact rows). *Let $\mathcal{M}$ be the finite mutant catalog named by a table row, let $\mathcal{D}_n$ be the frozen seeded harness distribution used by the artifact, and let x_m be the number of accepting runs for mutant $m \in \mathcal{M}$ over n_m runs. The unconditional empirical statement of the row is exactly the collection of estimates $\hat{p}_m = x_m/n_m$ with the stated confidence-interval convention. It implies no statement about a mutant outside $\mathcal{M}$ unless an additional theorem-covered class claim is invoked.*

Proof. The replay script fixes the baseline, patch, mutant, parameter set, harness, seed schedule, and acceptance predicate. Therefore each x_m/n_m is an empirical estimate for that named finite experiment. For any mutant $m' \notin \mathcal{M}$, one can define a deterministic implementation that agrees with a tested mutant on all frozen inputs and differs on an untested input. The finite transcript distribution cannot distinguish these two implementations. Hence the finite row alone cannot imply any statement about m' . $\square$

The experimental section reports rows as empirical witnesses for theorem-relevant behavior under named harnesses and fixed catalogs. After Theorem 5.37, a negative honest-reference row has a formal interpretation when the omitted operation is outside the support-active confirmation cone and the implementation class contains the coupled erasure: no black-box honest-support certifier can certify that operation without changing the code source, adding instrumentation, or sampling a different support. A positive row is a finite-catalog observation unless it is linked to one of the theorem-covered routes: source cUP, conditional entropy, hashed-witness entropy, TailEnt, support equivalence, self/reference separation, or dependency-cone non-certification. Negative rows use the last route only when a cone-inactivity certificate of Definition 5.40 is supplied.

Artifact family	Honest-reference claim level	Additional harness claim	Reason
Source ML-KEM selected-code faults	eligible for theorem-covered positive status once source cUP and a cUP-faithful harness certificate are supplied	none needed for selected code nodes beyond that certificate	selected coefficients lie in the source confirmation cone
ML-KEM unselected coefficient omissions	selected-source-cone boundary; full non-certification only with a $\text{Tgt}_{cc, \mathcal{H}}$ cone certificate	requires larger selected set, instrumentation, malformed support, or differential support	outside the selected source-code cone by Table 6; see Table 3 for the extra full-target condition
ML-KEM-CD1 drop/overwrite diagnostic code	theorem-covered positive under hashed-witness theorem plus RawEnt, or finite-catalog if RawEnt is not asserted	none for the named deterministic mutants	hidden diagnostic code changes the final-key point
Plain reencryption omissions	finite-catalog unless a separate ordinary-path cone certificate is supplied	malformed/differential rows may be theorem-covered for the sampled mode	may be visible through ordinary output behavior, not through the confirmation-code cone alone
HQC-CD1 retained tail-bit faults	theorem-covered positive under HQC-TailEnt for retained bits	none for retained tail cone	retained bits enter the hidden final-key point
HQC-CD1 non-retained tail-bit omissions	theorem-covered non-certification only with the retained-tail cone certificate	use CD*, instrumentation, or a harness exposing those bits	outside CD1 retained tail cone by Tables 3 and 11
HQC-CD* retained tail-bit faults	theorem-covered positive under HQC-TailEnt for the full tail	none for tail bits	all tail bits enter the retained code cone
Always-select- K_+ decision faults on honest ciphertexts	theorem-covered non-certification only with the honest-support branch certificate	malformed/differential support establishes the positive branch claim	branch selector is support-constant on honest ciphertexts; see Table 3
Component-selective comparison faults on honest ciphertexts	theorem-covered non-certification only for rows with a mode-specific cone certificate	differential support establishes the mode-selective claim	comparison target must be certified inactive as a set for that mode
Symmetric endpoint faults	theorem-covered separation	honest-self and honest-reference must both be reported	peer model changes the sampled reference key
Witness-guess mutants	theorem-covered quantitative row for the named guessing interface	reset-product theorem for repeated independent trials	pass probability follows the retained-code list-hit scale

Table 18: Artifact claim levels after the dependency-cone theorem. Negative honest-reference rows are upgraded from mere empirical absences to theorem-covered non-certification statements only when the omitted operation set has a cone-inactivity certificate for the stated target, support, and erasure-containing implementation class.

9.1 Binding faults

Line	Mutant family	Fault class	$D_{\text{honest_ref}}$	D_{ctmal}	Reading
ML-KEM plain	reencryption omission	recomputation	detected (3/3)	undetected (0/3)	the listed recomputation mutants are visible on this honest support
ML-KEM VSFO/CD1	overwrite or drop cd	binding	detected (6/6)	undetected (0/6)	the transformed final key exposes the missing code
ML-KEM VSFO/CD1	always select K_+	decision-only	undetected (0/3)	detected (3/3)	malformed support samples this branch behavior
August 2025 HQC plain	reencryption omission	recomputation	detected (3/3)	undetected (0/3)	the August 2025 public line exposes this listed recomputation mutant on this support
August 2025 HQC VSFO/CD1	overwrite or drop cd	binding	detected (6/6)	undetected (0/6)	the salted line exposes the missing code
August 2025 HQC VSFO/CD1	always select K_+	decision-only	undetected (0/3)	detected (3/3)	malformed support samples this branch behavior

Table 19: Main deterministic mutant families over the frozen parameter-set tranche. Counts are detected families over tested rows for the stated harness.

Observation 9.3. *On both scheme families, the listed plain reencryption-omission mutants are detected under honest-reference testing on the frozen support. The transformed rows isolate the additional signal introduced by the confirmation-code path: faults that remove, overwrite, or fail to bind the recomputed code change the good-key input under $D_{\text{honest_ref}}$, whereas branch-only faults require malformed or differential support.*

9.2 CD1 versus CD*

Fault family on August 2025 HQC	CD1	CD*	Reading
overwrite cd	detected (3/3) under $D_{\text{honest_ref}}$	detected (3/3) under $D_{\text{honest_ref}}$	both variants bind the code into the final good key
drop cd from K_+	detected (3/3) under $D_{\text{honest_ref}}$	detected (3/3) under $D_{\text{honest_ref}}$	deterministic behavior is unchanged
always select K_+	undetected (0/3) under $D_{\text{honest_ref}}$	undetected (0/3) under $D_{\text{honest_ref}}$	the support limitation is unchanged
compare only u	mode-selective (0/3 on honest support)	mode-selective (0/3 on honest support)	both variants keep the same comparison surface

Table 20: August 2025 HQC CD1 and CD* on the deterministic mutant tranche. The variants differ quantitatively; the deterministic fault classes coincide.

9.3 Self-vs-reference tests

Line	Symmetric mutant	$D_{\text{honest_ref}}$	$D_{\text{honest_self}}$	Reading
ML-KEM VSFO/CD1	drop cd at both endpoints	detected (3/3)	undetected (0/3)	self-tests pass because both endpoints agree on the same faulty key input
August 2025 HQC VSFO/CD1	drop cd at both endpoints	detected (3/3)	undetected (0/3)	the salted line has the same peer-model separation
August 2025 HQC VSFO/CD*	drop cd_* at both endpoints	detected (3/3)	undetected (0/3)	the separation is independent of the retained code budget

Table 21: Self-consistent endpoint faults. These rows instantiate Theorem 5.25.

9.4 Probabilistic code guessing

Line	Bits	Trials	Passes	Est. pass prob.	95% CI / ub	Ref. scale
ML-KEM-512-CD1	8	4 096	14	0.003 4	[0.0020, 0.0057]	2^{-8}
ML-KEM-768-CD1	8	4 096	10	0.002 4	[0.0013, 0.0045]	2^{-8}
ML-KEM-1024-CD1	8	4 096	24	0.005 9	[0.0039, 0.0087]	2^{-8}
HQC-1-CD1	5	4 096	151	0.036 9	[0.0315, 0.0431]	2^{-5}
HQC-3-CD1	8	4 096	21	0.005 1	[0.0034, 0.0078]	2^{-8}
HQC-5-CD1	8	4 096	13	0.003 2	[0.0019, 0.0054]	2^{-8}
HQC-1-CD*	5	4 096	151	0.036 9	[0.0315, 0.0431]	2^{-5}
HQC-3-CD*	11	8 192	5	0.000 6	[0.0003, 0.0014]	2^{-11}
HQC-5-CD*	37	4 096	0	0	[0, 0.0009]	2^{-37}

Table 22: Probabilistic code-guessing mutants under $D_{\text{honest_ref}}$. The reference scale is 2^{-L} for the tested witness-guess mutant with an L -bit retained code. The interval column uses Wilson score intervals for nonzero counts and the exact Clopper–Pearson upper endpoint for zero counts.

Code bits	Bytes	Trials	Passes	Est. pass prob.	95% interval	Ref. scale
1	1	4 096	2 046	0.499 5	[0.4842, 0.5148]	0.500 0
2	1	4 096	1 045	0.255 1	[0.2420, 0.2687]	0.250 0
4	1	4 096	273	0.066 7	[0.0594, 0.0747]	0.062 5
8	1	4 096	21	0.005 1	[0.0034, 0.0078]	0.003 9
11	2	4 096	3	0.000 7	[0.0002, 0.0022]	0.000 5

Table 23: Selected rows from the August 2025 HQC range-budget sweep. The full artifact sweep covers $L = 1, \dots, 11$ and is plotted in Figure 2; intervals use the same convention as Table 22.

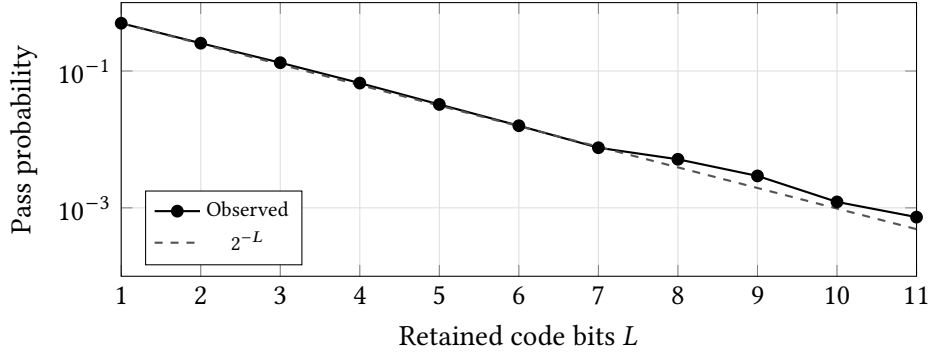


Figure 2: August 2025 HQC range-budget sweep under $D_{\text{honest_ref}}$. The observed points use the full artifact sweep; selected rows are listed in Table 23. For these witness-guess mutants, retaining more hidden code bits reduces the observed pass probability on the order of the 2^{-L} reference scale.

Observation 9.4. *The probabilistic rows are reported against the 2^{-L} reference scale for the witness-guess mutants tested here. The ML-KEM and HQC one-byte variants fall on the corresponding 2^{-8} or 2^{-5} scale, and the August 2025 HQC sweep tracks the same order of magnitude through $L = 11$. The HQC-5-CD* row has zero passes in 4096 trials; the table records the finite-run exact upper endpoint, and the 2^{-37} entry records the retained range budget for that row.*

9.5 Mode-selective comparison faults

Line	Fault	honest-ref	flip _u	flip _v	Reading
ML-KEM plain	compare only u	undetected (0/3)	undetected (0/3)	detected (3/3)	comparison coverage depends on the corrupted component
ML-KEM VSFO/CD1	compare only u	undetected (0/3)	undetected (0/3)	detected (3/3)	the comparison-mode boundary is unchanged
August 2025 HQC plain	compare only u	undetected (0/3)	undetected (0/3)	detected (3/3)	the August 2025 HQC baseline has the same signature
August 2025 HQC VSFO/CD1	compare only u	undetected (0/3)	undetected (0/3)	detected (3/3)	code binding leaves this comparison boundary unchanged

Table 24: Mode-selective differential signatures for the component-selective comparison fault.

9.6 Summary across campaigns

Fault family	Honest-reference behavior	Other harness behavior	Interpretation
plain reencryption omission	detected in representative rows	secondary malformed signal	some recomputation faults are already visible before adding a code
transformed binding faults	detected	secondary	the added code creates the honest-reference signal
branch-only decision faults	undetected	detected on malformed or differential support	support-coincidence limitation
component-selective comparison	undetected	detected only on matching modes	comparison coverage is mode-indexed
symmetric endpoint faults	detected against reference	undetected under self-tests	peer-model separation
probabilistic code guesses	quantitative	optional	survival scale follows the retained-code reference scale for the tested mutants

Table 25: Family-level summary of the frozen campaigns.

At campaign level, the transformed variants add evidence for the code-bearing good-key path under a named honest-correctness harness and a named support, with negative rows marking the expected boundaries of that harness.

10 Scope and Interpretation

The positive theorem covers honest-reference tests through localized final-key list hits, while branch-selection faults that agree with correct decapsulation on honest ciphertexts are handled by malformed and differential harnesses within their own supports.

Relative cone closure of the testing layer. Within the classical ROM model, honest ciphertext support, black-box observation boundary, and erasure-containing implementation class specified in Section 5.7, the honest-reference claim grammar is closed in the following sense. Inside the support-active confirmation cone, positive claims require a bound on the list-hit term, supplied here by cUP, conditional entropy, hashed-witness entropy, or TailEnt. Outside that cone, Theorem 5.37 shows that no black-box honest-support certifier can prove execution over an implementation class containing the coupled erasure. The resulting claim grammar matches the black-box honest-reference interface under the stated model and erasure-class conditions.

The statements apply to confirmation-code-augmented variants, with standard ML-KEM and the August 2025 public HQC line appearing as baselines and algorithmic anchors. The augmented variants keep the relevant ciphertext formats in the artifact, but their good-branch shared secrets include confirmation-code material. A standard implementation obtains this testing signal through an adopted augmented test mode, exported confirmation-code mode, or corresponding modified line.

ROM versus concrete hash testing. The $2^{-\kappa}$ term in Theorem 5.1 is a random-oracle-model statement for a lazy-sampled G_K . The artifact uses concrete SHA3/SHAKE implementations with domain separation. Rows interpreted through the theorem must satisfy the structural ROM-interface certificate of Definition 8.1, which records the namespace mapping, serialization, query logging, and absence of hidden witness leakage. The experimental rows therefore report concrete pass/fail behavior and

empirical confidence intervals for the tested mutants; they are not empirical measurements of the ROM fresh-key-coincidence term or proofs that the concrete primitives are random oracles.

The reproducible mutant campaign runs over concrete ML-KEM and August 2025 HQC code lines, with timing measurements kept as artifact context, which use the harness outcomes and the probabilistic pass estimates.

Security of the augmented KEM line is cited from the existing confirmation-code-augmented FO results, and the statements here concern testing evidence under named harnesses. For HQC, the paper proves a retained range budget and states the conditional TailEnt assumption needed to turn that budget into the entropic theorem. For ML-KEM, the source-paper multi-byte codes carry the cited cUP analysis through the list-cUP bridge; the artifact’s one-byte CD1 line is a hashed diagnostic specialization governed by Theorem 7.2.

The testing theorems are classical ROM statements about harnessed implementations, with IND-CCA security for ML-KEM and HQC and QROM guarantees for the underlying confirmation-code transform cited as construction-level claims. Short diagnostic codes are engineering instruments whose security meaning is captured by the list-hit, min-entropy, and hashed-witness bounds. A retained range of L bits is only a capacity bound; detectability requires conditional unpredictability of the retained witness relative to the SUT view.

11 Conclusion

Confirmation-code-augmented decapsulation connects correctness testing to the final key derivation by placing a hidden witness value in the good-branch oracle input, so a faulty decapsulator that agrees with a sound peer on honest ciphertexts must compute that witness, query it under the localized prefix, or pass through the residual freshness term. In the honest-reference harness, once correctness errors, aliasing, and freshness defects are separated, acceptance is bounded by a localized final-key list hit and a fresh-key coincidence. The theorem ties the peer model and the sampled support to the claim, since honest-self tests can mask symmetric endpoint faults, malformed and differential tests expose branch or component-specific behavior, and repeated campaigns amplify only under reset, fresh instances, and isolated oracle namespaces.

The list-hit term is controlled by source cUP, conditional min-entropy, or a hashed diagnostic witness, according to how the confirmation code is obtained and what the SUT view contains. Source cUP applies through a cUP-faithful interface with no extra witness path; conditional min-entropy yields the $q2^{-\lambda}$ bound; hashed diagnostic codes separate raw-witness discovery from truncated-code guessing; and TailEnt supplies the additional hypothesis needed to interpret an HQC tail range as a detectability statement. Under these hypotheses, the source ML-KEM confirmation-code line uses the cited cUP analysis through the cUP-faithful route, the one-byte ML-KEM artifact line is a hashed diagnostic experiment, and the August 2025 HQC lines differ by the number of retained tail bits in CD1 and CD*.

The support-active dependency cone identifies which operations a black-box honest-reference transcript can certify on a chosen support. For a black-box honest-reference harness on honest ciphertext support, operations outside the confirmation-observable target admit a coupled erasure implementation with the same transcript distribution over any implementation class containing that erasure. Over erasure-closed classes, no execution certifier for such operations can have soundness and completeness errors summing to less than one. A negative artifact row has a theorem-covered non-certification interpretation only when its omitted operation set has a cone-inactivity certificate for the stated target, support, and class. Operations outside that certified cone require a different target, such as a larger retained code, malformed or differential support, or instrumentation that exposes the missing dependency.

FO-FaultBench records this claim grammar on pinned ML-KEM and August 2025 HQC code lines. The stored campaigns show that the derived variants make the listed missing-code binding faults visible under honest-reference testing, that branch-only and component-selective faults fall on the supports

predicted by the cone analysis, that symmetric faults separate honest-self from honest-reference testing, and that probabilistic witness-guess mutants follow the retained-code scale on the tested distributions. The resulting testing theory for confirmation-code-augmented FO KEM variants, paired with the reproducible implementation package, specifies what a harness proves, which hypotheses bound the list-hit term, and which operations remain outside black-box certification until the harness target or support changes.

References

- [1] Gorjan Alagic et al. *Recommendations for Key-Encapsulation Mechanisms*. NIST Special Publication NIST SP 800-227. National Institute of Standards and Technology, Sept. 2025. DOI: 10.6028/NIST.SP.800-227. URL: <https://csrc.nist.gov/pubs/sp/800/227/final>.
- [2] Manuel Barbosa and Andreas Hülsing. *The security of Kyber’s FO-transform*. Cryptology ePrint Archive, Paper 2023/755. 2023. URL: <https://eprint.iacr.org/2023/755>.
- [3] Nina Bindel et al. “Tighter Proofs of CCA Security in the Quantum Random Oracle Model”. In: *Theory of Cryptography – 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part II*. Ed. by Dennis Hofheinz and Alon Rosen. Vol. 11892. Lecture Notes in Computer Science. Springer, 2019, pp. 61–90. DOI: 10.1007/978-3-030-36033-7_3. URL: https://doi.org/10.1007/978-3-030-36033-7_3.
- [4] Xiaohui Ding et al. “An Injectivity Analysis of Crystals-Kyber and Implications on Quantum Security”. In: *Information Security and Privacy – 27th Australasian Conference, ACISP 2022, Wollongong, NSW, Australia, November 28–30, 2022, Proceedings*. Ed. by Khoa Nguyen et al. Vol. 13494. Lecture Notes in Computer Science. Springer, 2022, pp. 332–351. DOI: 10.1007/978-3-031-22301-3_17. URL: https://doi.org/10.1007/978-3-031-22301-3_17.
- [5] Jelle Don et al. “Online-Extractability in the Quantum Random-Oracle Model”. In: *Advances in Cryptology – EUROCRYPT 2022 – 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 677–706. DOI: 10.1007/978-3-031-07082-2_24. URL: https://doi.org/10.1007/978-3-031-07082-2_24.
- [6] Julien Duman et al. “Faster Lattice-Based KEMs via a Generic Fujisaki-Okamoto Transform Using Prefix Hashing”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2021, pp. 2722–2737. DOI: 10.1145/3460120.3484819. URL: <https://doi.org/10.1145/3460120.3484819>.
- [7] Marc Fischlin and Felix Günther. “Verifiable Verification in Cryptographic Protocols”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2023, pp. 3239–3253. DOI: 10.1145/3576915.3623151. URL: <https://doi.org/10.1145/3576915.3623151>.
- [8] Philippe Gaborit et al. *Hamming Quasi-Cyclic (HQC)*. HQC specifications. Aug. 2025. URL: https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf.
- [9] Jiangxia Ge, Heming Liao, and Rui Xue. “Measure-Rewind-Extract: Tighter Proofs of One-Way to Hiding and CCA Security in the Quantum Random Oracle Model”. In: *Advances in Cryptology – ASIACRYPT 2024 – 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings, Part IV*. Ed. by Kai-Min Chung and Yu Sasaki. Vol. 15487. Lecture Notes in Computer Science. Springer, 2025, pp. 3–34. DOI: 10.1007/978-981-96-0894-2_1. URL: https://doi.org/10.1007/978-981-96-0894-2_1.
- [10] Jiangxia Ge, Tianshu Shan, and Rui Xue. *On the Fujisaki-Okamoto transform: from Classical CCA Security to Quantum CCA Security*. Cryptology ePrint Archive, Paper 2023/792. 2023. URL: <https://eprint.iacr.org/2023/792>.

- [11] Jiangxia Ge, Tianshu Shan, and Rui Xue. “Tighter QCCA-Secure Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model”. In: *Advances in Cryptology – CRYPTO 2023 – 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20–24, 2023, Proceedings, Part V*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 292–324. DOI: 10.1007/978-3-031-38554-4_10. URL: https://doi.org/10.1007/978-3-031-38554-4_10.
- [12] Lewis Glabush, Kathrin H"ovelmans, and Douglas Stebila. *Tight Multi-challenge Security Reductions for Key Encapsulation Mechanisms*. Cryptology ePrint Archive, Paper 2025/343. 2025. URL: <https://eprint.iacr.org/2025/343>.
- [13] Lewis Glabush et al. “Verifiable Decapsulation: Recognizing Faulty Implementations of Post-quantum KEMs”. In: *Advances in Cryptology – CRYPTO 2025 – 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2025, Proceedings, Part III*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16002. Lecture Notes in Computer Science. Springer, 2025, pp. 543–574. DOI: 10.1007/978-3-032-01881-6_17. URL: https://doi.org/10.1007/978-3-032-01881-6_17.
- [14] Kathrin H"ovelmans, Andreas H"ulsing, and Christian Majenz. “Failing Gracefully: Decryption Failures and the Fujisaki-Okamoto Transform”. In: *Advances in Cryptology – ASIACRYPT 2022 – 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 414–443. DOI: 10.1007/978-3-031-22972-5_15. URL: https://doi.org/10.1007/978-3-031-22972-5_15.
- [15] Kathrin H"ovelmans and Mikhail A. Kudinov. “Treating Dishonest Ciphertexts in Post-quantum KEMs – Explicit vs. Implicit Rejection in the FO Transform”. In: *Post-Quantum Cryptography – 16th International Workshop, PQCrypto 2025, Taipei, Taiwan, April 8–10, 2025, Proceedings, Part II*. Ed. by Ruben Niederhagen and Markku-Juhani O. Saarinen. Vol. 15578. Lecture Notes in Computer Science. Springer, 2025, pp. 325–350. DOI: 10.1007/978-3-031-86602-9_12. URL: https://doi.org/10.1007/978-3-031-86602-9_12.
- [16] Kathrin H"ovelmans and Christian Majenz. “A Note on Failing Gracefully: Completing the Picture for Explicitly Rejecting Fujisaki-Okamoto Transforms Using Worst-Case Correctness”. In: *Post-Quantum Cryptography – 15th International Conference, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings, Part II*. Ed. by Markku-Juhani Saarinen and Daniel Smith-Tone. Vol. 14772. Lecture Notes in Computer Science. Springer, 2024, pp. 245–265. DOI: 10.1007/978-3-031-62746-0_11. URL: https://doi.org/10.1007/978-3-031-62746-0_11.
- [17] Kathrin H"ovelmans et al. “(Un)breakable Curses – Re-encryption in the Fujisaki-Okamoto Transform”. In: *Advances in Cryptology – EUROCRYPT 2025 – 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4–8, 2025, Proceedings, Part II*. Ed. by Serge Fehr and Pierre-Alain Fouque. Vol. 15602. Lecture Notes in Computer Science. Springer, 2025, pp. 245–274. DOI: 10.1007/978-3-031-91124-8_9. URL: https://doi.org/10.1007/978-3-031-91124-8_9.
- [18] Dennis Hofheinz, Kathrin H"ovelmans, and Eike Kiltz. “A Modular Analysis of the Fujisaki-Okamoto Transformation”. In: *Theory of Cryptography – 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12–15, 2017, Proceedings, Part I*. Ed. by Yael Kalai and Leonid Reyzin. Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 341–371. DOI: 10.1007/978-3-319-70500-2_12. URL: https://doi.org/10.1007/978-3-319-70500-2_12.
- [19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. “Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model”. In: *Public-Key Cryptography – PKC 2019 – 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14–17, 2019, Proceedings, Part II*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11443. Lecture

Notes in Computer Science. Springer, 2019, pp. 618–645. DOI: 10.1007/978-3-030-17259-6_21. URL: https://doi.org/10.1007/978-3-030-17259-6_21.

- [20] Christian Majenz and Fabrizio Sisinni. “Provable Security Against Decryption Failure Attacks from LWE”. In: *Advances in Cryptology – CRYPTO 2024 – 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part II*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14921. Lecture Notes in Computer Science. Springer, 2024, pp. 456–485. DOI: 10.1007/978-3-031-68379-4_14. URL: https://doi.org/10.1007/978-3-031-68379-4_14.
- [21] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication NIST FIPS 203. National Institute of Standards and Technology, Aug. 2024. DOI: 10.6028/NIST.FIPS.203. URL: <https://csrc.nist.gov/pubs/fips/203/final>.
- [22] Tianshu Shan, Jiangxia Ge, and Rui Xue. “QCCA-Secure Generic Transformations in the Quantum Random Oracle Model”. In: *Public-Key Cryptography – PKC 2023 – 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7–10, 2023, Proceedings, Part I*. Ed. by Alexandra Boldyreva and Vladimir Kolesnikov. Vol. 13940. Lecture Notes in Computer Science. Springer, 2023, pp. 36–64. DOI: 10.1007/978-3-031-31368-4_2. URL: https://doi.org/10.1007/978-3-031-31368-4_2.

A Additional HQC Sweep

Bits	Bytes	Trials	Passes	Est. prob.	95% CI / ub	Ref. scale
1	1	4 096	2 046	0.499 5	[0.4842, 0.5148]	0.500 0
2	1	4 096	1 045	0.255 1	[0.2420, 0.2687]	0.250 0
3	1	4 096	542	0.132 3	[0.1223, 0.1430]	0.125 0
4	1	4 096	273	0.066 7	[0.0594, 0.0747]	0.062 5
5	1	4 096	133	0.032 5	[0.0275, 0.0384]	0.031 3
6	1	4 096	65	0.015 9	[0.0125, 0.0202]	0.015 6
7	1	4 096	31	0.007 6	[0.0053, 0.0107]	0.007 8
8	1	4 096	21	0.005 1	[0.0034, 0.0078]	0.003 9
9	2	4 096	12	0.002 9	[0.0017, 0.0051]	0.002 0
10	2	4 096	5	0.001 2	[0.0005, 0.0029]	0.001 0
11	2	4 096	3	0.000 7	[0.0002, 0.0022]	0.000 5

Table 26: Full August 2025 HQC range-budget sweep for $L = 1, \dots, 11$, with the same interval convention as the main probabilistic tables.