Domain-Separated Commitments under Adaptive Selective Opening

Anonymous Authors

Anonymous Institution
contact@anonymous.invalid

Abstract. Commitments are routinely created in batches and opened only after an adversary has inspected the full batch. The resulting selective-opening setting is subtle: ordinary hiding of each commitment does not by itself imply that unopened messages remain hidden. We give a compact, audit-oriented treatment of domain-separated commitments in the random-oracle model. A commitment binds a protocol identifier, session identifier, position, message, and fresh salt through an injective encoding. We prove adaptive selective-opening security for high-min-entropy message distributions by an identical-until-bad argument. The bound exposes exactly three resources: the number of oracle queries, the number of unopened commitments, and the residual entropy of an unopened message conditioned on all opened messages. We also prove computational binding, show why omitting the session or position field invalidates natural composition claims, and give concrete parameter guidance. The result is intentionally modest but reusable: it separates the information- theoretic guessing step from syntax and collision resistance, making both the proof and an implementation straightforward to review.

Keywords: commitment schemes · selective opening · random-oracle model · domain separation · min-entropy

1 Introduction

A commitment lets a sender fix a message while keeping it hidden until a later opening. The two usual requirements are hiding and binding: the commitment should reveal no message, and the sender should not be able to open it in two ways. These definitions are deliberately local. Modern applications, however, rarely create a single isolated commitment. Voting systems commit to many ballots, distributed protocols commit in concurrent sessions, and zero-knowledge compilers commit to vectors of correlated values.

In such uses an adversary first sees a batch and then chooses which positions to open. This order creates the *selective-opening* problem [6,1]. Per-message hiding is not a generic solution: the opened messages may be correlated with the unopened ones, and a simulator must reproduce a joint view selected adaptively. The problem is partly cryptographic and partly syntactic. If two sessions share an untyped hash domain, an opening produced in one context may accidentally be meaningful in another.

This paper studies a transparent special case. Messages come from a joint distribution having residual conditional min-entropy, and commitments use a random oracle with fresh salts and explicit domain separation. For session identifier sid and position i, the commitment is

$$C_i = \mathsf{H}\big(\mathsf{Enc}(\mathsf{Tag}, sid, i, m_i; r_i)\big), \tag{1}$$

where $r_i \xleftarrow{\$} \{0,1\}^\rho$. The simulator replaces each initially hidden message with a fixed dummy value while retaining the same public syntax. It answers adaptive openings using the real messages and programs the corresponding oracle points. The only observable failure occurs if the adversary queried such a point before it was programmed.

1.1 Contributions

Our goal is a proof that can be read beside an implementation. The main contributions are:

- a fully typed commitment syntax that binds the protocol version, session, vector position, message, and salt;
- a game-based adaptive selective-opening theorem whose loss is stated directly in terms of residual conditional min-entropy; and
- matching binding and composition analyses, including concrete bounds and counterexamples for omitted domain fields.

For n commitments and an adversary making q_H random-oracle queries, our central bound has the form

$$\mathrm{Adv}^{\mathsf{aso}}(\mathcal{A}) \le q_H n\, 2^{-(k+\rho)} + \frac{n(n-1)}{2^{\rho+1}} + \varepsilon_{\mathsf{enc}}, \tag{2}$$

where k is the residual conditional min-entropy of each unopened message and $\varepsilon_{\mathsf{enc}}$ captures any failure of the implementation's canonical encoder. For the specified encoder this last term is zero. The bound is conservative, easy to instantiate, and highlights that salt entropy and message entropy contribute additively against pre-query attacks.

Scope. We work in the classical random-oracle model (ROM), and our positive result assumes residual message entropy. We do not claim security for arbitrary correlated distributions or against quantum oracle queries. Those settings need stronger notions and different proof tools [4,7]. Our construction should therefore be read as a narrow, useful component rather than a universal selective-opening compiler.

2 Preliminaries

For $n \in \mathbb{N}$, write $[n] = \{1, \ldots, n\}$. Sampling x uniformly from a finite set S is denoted $x \xleftarrow{\$} S$. All algorithms receive the unary security parameter 1^λ. A

function is negligible if it eventually falls below the inverse of every positive polynomial. Probabilities are over the coins of all algorithms and the random oracle.

2.1 Min-entropy with side information

The min-entropy of a discrete random variable X is

$$H_\infty(X) = -\log_2 \max_x \Pr[X = x].$$

For jointly distributed (X, Z), we use the worst-case conditional quantity

$$H_\infty(X \mid Z) = -\log_2 \max_{z \in \mathrm{Supp}(Z), x} \Pr[X = x \mid Z = z]. \tag{3}$$

This strong form is convenient for adaptive openings: after every admissible history, no particular unopened message has probability greater than 2^{-k}.

Definition 1 (Residual k-entropy). *Let $M = (M_1, \ldots, M_n) \leftarrow \mathcal{D}$. We say that $\mathcal{D}$ has residual k-entropy if, for every $i \in [n]$ and every $I \subseteq [n] \setminus \{i\}$,*

$$H_\infty(M_i \mid M_I, I) \geq k,$$

where $M_I = (M_j)_{j \in I}$. Public auxiliary information is included as an additional conditioned variable when present.

The definition excludes, for example, a batch in which all messages are copies of one low-entropy secret. It permits arbitrary correlations so long as every still-hidden coordinate retains the stated pointwise uncertainty.

2.2 Random oracles and programming

A random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\tau$ is represented by a lazy-sampled table T. A fresh input X receives an independent uniform value $T[X] \xleftarrow{\$} \{0,1\}^\tau$; repeated queries return the stored value. A simulator may program $T[X] := y$ without changing the distribution provided that X has not previously been queried. We record a bad event whenever programming is attempted at a point already in T.

The following standard lemma is the engine of our proof.

Lemma 1 (Identical until bad). *Let games G_0 and G_1 proceed identically until an event bad, and let b_j be the output of G_j. Then*

$$\big|\Pr[b_0 = 1] - \Pr[b_1 = 1]\big| \leq \Pr[\mathsf{bad}].$$

Proof. Couple the games with identical randomness up to their first divergent step. If bad does not occur, their complete transcripts and outputs agree. Taking a union bound over the remaining executions gives the claim.

2.3 Canonical encodings

For a byte string x, define $L(x)$ as a fixed-width encoding of $|x|$ followed by x. Integers are represented in unsigned big-endian form using a public fixed width. We use

$$\begin{aligned}\mathsf{Enc}(\mathsf{Tag}, sid, i, m; r) = \texttt{TCC\_COM\_V1} \,\|\, L(\mathsf{Tag}) \,\|\, L(sid) \\ \|\, \mathsf{I2OSP}(i) \,\|\, L(m) \,\|\, L(r). \end{aligned} \tag{4}$$

The literal prefix is an ASCII byte string, not human-readable metadata that may be normalized by a transport layer. Every accepted opening is parsed and then re-encoded canonically before hashing.

Lemma 2 (Context separation). *If two strings produced by Equation* (4) *are equal, then their protocol tags, session identifiers, positions, messages, and salts are respectively equal.*

Proof. The version prefix and integer width are fixed. Each variable-length field is length-delimited, so a left-to-right parser has exactly one valid parse. Equality of encoded strings therefore implies equality field by field.

3 Syntax and Security Experiments

A non-interactive commitment scheme consists of three algorithms: Setup, Com, and Vfy. Setup selects public parameters $pp \leftarrow \mathsf{Setup}(1^\lambda)$. The commitment algorithm returns $(C, d) \leftarrow \mathsf{Com}(pp, sid, i, m)$, where d is the opening. The verifier outputs $\mathsf{Vfy}(pp, sid, i, C, m, d) \in \{0, 1\}$. Perfect correctness requires honest openings always to verify.

3.1 Adaptive selective-opening experiment

We compare two experiments shown in Figure 1. Both sample a message vector $M \leftarrow \mathcal{D}$ and give the adversary a vector of commitments. The adversary may make random-oracle queries and adaptively ask to open positions. Its opening set I may depend on commitments and prior answers. Finally it outputs a bit based on its entire view.

In Real, each commitment is produced from M_i. In Ideal, a simulator first publishes independent uniform strings of the required commitment length. Upon an opening query i, it returns (M_i, r_i) and programs the real oracle point to equal the already published C_i. If that point was queried earlier, the ideal game sets $\mathsf{bad}_{\mathsf{pre}}$ and aborts. Message length is regarded as public leakage; applications requiring length hiding must pad before invoking the scheme.

Definition 2 (Adaptive selective-opening advantage). *For an adversary $\mathcal{A}$, distribution $\mathcal{D}$, and simulator Sim, define*

$$\mathrm{Adv}^{\mathsf{aso}}_{\Pi,\mathcal{D}}(\mathcal{A}, \mathsf{Sim}) = \left| \Pr[\mathsf{Real}^{\mathcal{A}}_{\Pi,\mathcal{D}} = 1] - \Pr[\mathsf{Ideal}^{\mathcal{A},\mathsf{Sim}}_{\Pi,\mathcal{D}} = 1] \right|.$$

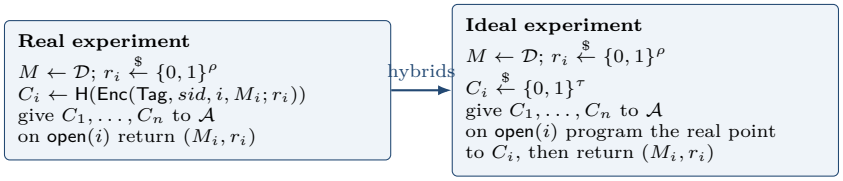

Fig. 1. The real and simulated views. Oracle queries are answered by a shared lazy-sampling interface. The ideal experiment aborts only if a real opening point was queried before its position was opened.

The adversary is limited to q_H distinct oracle queries and at most n adaptive opening queries.

This indistinguishability formulation captures the claim needed here: before opening a position, its commitment can be simulated without its message. Stronger simulation definitions may additionally demand recovery of the adversary's state or support arbitrary leakage functions; we make no such claim.

3.2 Binding experiment

An adversary wins binding if it outputs $(sid, i, C, m, r, m', r')$ such that $(m, r) \neq (m', r')$ and both openings verify for the same tag, session, position, and commitment. Write $\mathrm{Adv}^{\mathsf{bind}}_{\Pi}(\mathcal{A})$ for its success probability. We count all oracle queries, including those implicit in verification, in q_H.

4 Construction

Let ρ be the salt length and τ the commitment length. The public parameters contain $(\rho, \tau, \mathsf{Tag}, \mathsf{Enc})$ and a random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\tau$.

$\mathsf{Setup}(1^\lambda)$**.** Choose parameters $\rho, \tau \geq \lambda$, fix a public protocol tag and the canonical encoder from Equation (4), and return $pp = (\rho, \tau, \mathsf{Tag}, \mathsf{Enc}, \mathsf{H})$.

$\mathsf{Com}(pp, sid, i, m)$**.** Sample $r \xleftarrow{\$} \{0,1\}^\rho$, compute

$$X = \mathsf{Enc}(\mathsf{Tag}, sid, i, m; r), \qquad C = \mathsf{H}(X),$$

and return (C, d) for $d = (m, r)$. If the API supplies m separately at opening time, storing only r is equivalent.

$\mathsf{Vfy}(pp, sid, i, C, m, r)$**.** Reject if any field is malformed or if $|r| \neq \rho$. Canonically encode $X = \mathsf{Enc}(\mathsf{Tag}, sid, i, m; r)$ and accept if and only if $C = \mathsf{H}(X)$.

Correctness is immediate. The construction is deliberately stateless, but an implementation should still reject duplicate positions within a session: this catches application errors and prevents confusing two semantic records that intentionally share the same hash domain.

5 Selective-Opening Analysis

We first isolate the guessing event. Fix a history in which position i has not been opened. By residual entropy, the conditional probability of every candidate m_i is at most 2^{-k}. Its salt is independent and uniform, so every pair (m_i, r_i) has conditional probability at most $2^{-(k+\rho)}$. Injectivity then transfers the same bound to the real oracle input.

Lemma 3 (Pre-query bound). *Suppose $\mathcal{D}$ has residual k-entropy. For any unopened position i and any adversary making q_H distinct oracle queries,*

$$\Pr[\mathsf{bad}_{\mathsf{pre},i}] \le q_H 2^{-(k+\rho)}.$$

Consequently, for a set U of at most n unopened positions,

$$\Pr\left[\bigvee_{i\in U} \mathsf{bad}_{\mathsf{pre},i}\right] \le q_H n 2^{-(k+\rho)}.$$

Proof. Condition on any transcript immediately before an oracle query and on all messages revealed so far. For a fixed unopened i, residual entropy bounds the chance of guessing M_i by 2^{-k}; independence of r_i contributes $2^{-\rho}$. By Lemma 2, no differently typed tuple encodes to the same point. Union bound over q_H queries and then over U. The argument remains valid under adaptive opening because it conditions on each realized history.

The simulator also needs published commitments to be independent. Random-oracle outputs at distinct inputs are independent, but inputs could coincide if two equal messages receive equal salts. Positions already prevent such a collision. We nevertheless retain a salt-collision term in the main theorem because many optimized implementations omit i when a separate per-position oracle is used. For the encoder specified here, that term can be deleted.

Lemma 4 (Salt collisions). *For n independent ρ-bit salts,*

$$\Pr[\exists i < j : r_i = r_j] \le \binom{n}{2} 2^{-\rho}.$$

With the position-aware encoder of Equation (4), equal salts do not produce equal oracle inputs for distinct positions.

Proof. Each pair collides with probability $2^{-\rho}$; apply a union bound. The second statement follows from Lemma 2.

Theorem 1 (Adaptive selective opening). *Let $\mathcal{D}$ have residual k-entropy, and let $\mathcal{A}$ create or receive at most n commitments and make at most q_H distinct random-oracle queries. There exists a straight-line simulator Sim such that*

$$\mathrm{Adv}^{\mathsf{aso}}_{\Pi,\mathcal{D}}(\mathcal{A}, \mathsf{Sim}) \le q_H n 2^{-(k+\rho)} + \varepsilon_{\mathsf{enc}}. \tag{5}$$

For a variant whose encoder omits the position but uses unique session-local records, add at most $\binom{n}{2}2^{-\rho}$ to the right-hand side.

Proof. Order positions by the time at which the adversary first opens them, placing never-opened positions last. Consider hybrids $G_0, \ldots, G_n$ that replace real commitments with uniform strings one position at a time. In a replaced position, the simulator samples $C_i \xleftarrow{\$} \{0,1\}^\tau$. If i is later opened, it programs

$$\mathsf{H}(\mathsf{Enc}(\mathsf{Tag}, sid, i, M_i; r_i)) := C_i$$

and returns (M_i, r_i). Before the first query to that input, a lazy-sampled random-oracle value is itself uniform, so adjacent hybrids are identically distributed unless the input was pre-queried or the encoder is non-canonical.

By Lemmas 1 and 3, the total distinguishing gap for all simulated positions is at most $q_H n 2^{-(k+\rho)}$. Canonical encoding makes $\varepsilon_{\mathsf{enc}} = 0$. Opened positions are handled at their opening time, and programming preserves their exact joint distribution. This yields Equation (5). If position is absent from the input, condition additionally on distinct salts and apply Lemma 4.

Corollary 1 (Computational message recovery). *Before opening position i, an adversary making q_H oracle queries recovers M_i with probability at most*

$$2^{-k} + q_H 2^{-(k+\rho)} + 2^{-\tau},$$

apart from public leakage and encoder failures.

Proof. In the ideal game the commitment is independent of M_i, so the best direct guess succeeds with probability at most 2^{-k}. Transfer to the real game using Theorem 1; the $2^{-\tau}$ term accounts for guessing an output in an interface that treats equality with a target digest as evidence of recovery.

6 Binding and Composition

Binding reduces to finding two distinct random-oracle inputs with the same output. The reduction is unusually direct because verification recomputes the full canonical encoding.

Theorem 2 (Binding). *For an adversary making at most q_H distinct oracle queries,*

$$\mathrm{Adv}^{\mathsf{bind}}_{\Pi}(\mathcal{A}) \le \binom{q_H + 2}{2} 2^{-\tau} + \varepsilon_{\mathsf{enc}}.$$

Proof. Two accepted, unequal openings for the same context yield distinct canonical inputs $X \neq X'$ by Lemma 2, but verification requires $\mathsf{H}(X) = C = \mathsf{H}(X')$. Across at most $q_H + 2$ evaluated inputs, the birthday bound for independent τ-bit outputs gives the stated probability. A non-canonical parser is represented by $\varepsilon_{\mathsf{enc}}$.

Table 1. Proof obligation associated with each encoded field.

Field	Binds	Failure if omitted
Version/tag	scheme and format	cross-protocol reinterpretation
Session *sid*	execution instance	replay across executions
Position i	semantic coordinate	vector permutation or substitution
Message m	committed payload	no message binding
Salt r	fresh entropy	efficient dictionary attack
Lengths/types	parse boundaries	tuple aliasing

6.1 Why every context field matters

The encoding is not decorative. Removing fields changes the theorem that can be claimed.

No protocol tag. Suppose a ballot protocol and a credential protocol both hash (sid, i, m, r) with the same oracle. A commitment produced for one is byte-for-byte valid in the other. Neither local binding nor local hiding is violated, yet cross-protocol replay is possible.

No session identifier. If positions restart at one in every session, a commitment and its later opening can be transplanted to another session. An application-level freshness check may stop the replay, but that check then becomes a missing hypothesis of the cryptographic composition theorem.

No position. Swapping commitments inside a vector preserves validity. This may be acceptable for an unordered set, but not for a ballot indexed by contest or a proof whose coordinates have different meanings. Including i also makes the oracle inputs distinct even when messages and salts repeat.

Ambiguous concatenation. Raw concatenation permits tuples such as $(\mathtt{a}, \mathtt{bc})$ and $(\mathtt{ab}, \mathtt{c})$ to share an input. Length prefixes or a canonical serialization format are necessary for the injectivity used in both main proofs.

7 Concrete Parameters and Implementation

The main theorem makes parameter selection mechanical. If an application wants the pre-query term below 2^{-s}, it suffices to choose

$$\rho \geq s + \lceil \log_2 q_H \rceil + \lceil \log_2 n \rceil - k. \tag{6}$$

Salt length should still have a conservative floor (typically 128 bits), since message entropy estimates are easy to overstate. Commitment length τ is chosen independently from the binding bound; achieving approximately s-bit binding against q_H queries requires

$$\tau \geq s + 2\lceil \log_2(q_H + 2) \rceil - 1. \tag{7}$$

Table 2. Illustrative bounds for $\rho = 128$, $\tau = 256$, $q_H \le 2^{40}$, and $n \le 2^{20}$. Exponents are conservative.

Residual k	Pre-query bound	Binding bound	Interpretation
0	2^{-68}	$< 2^{-175}$	salt carries hiding loss
32	2^{-100}	$< 2^{-175}$	moderate-entropy data
64	2^{-132}	$< 2^{-175}$	high-entropy secret
128	2^{-196}	$< 2^{-175}$	binding dominates

The numbers in Table 2 are examples, not universal deployment targets. The query budget must include parallel attackers and the lifetime of the chosen domain. If a standardized hash is used as a ROM heuristic, use an approved output length and retain the full digest unless a protocol analysis justifies truncation.

7.1 Implementation checklist

1. Generate each salt with a cryptographically secure random-number generator. Never derive it from a timestamp or resettable counter.
2. Allocate a stable, documented protocol tag and version. Do not reuse a tag for a changed serialization.
3. Encode lengths and integers canonically. Reject overlong integers, duplicate map keys, alternative Unicode normalizations, and trailing bytes before hashing.
4. Make *sid* globally unique for the required replay horizon, and define exactly which application fields it authenticates.
5. Compare digests in constant time when the surrounding protocol makes timing observable, and erase retained openings when no longer needed.
6. Test negative cases: altered tag, session, position, message, and salt must each cause verification to fail.

8 Related Work and Limitations

Commitment schemes originate in foundational work on cryptographic protocol design, including Blum's coin-flipping methodology [3]. The random-oracle model provides a widely used abstraction for hash-based proofs [2], while Canetti, Goldreich, and Halevi emphasize that secure ROM proofs do not automatically instantiate securely with every concrete hash function [5]. Our theorem should be read with that methodological limitation in mind.

Selective opening was isolated in the context of concurrent and composable protocols by Dwork, Naor, Reingold, and Stockmeyer [6]. Subsequent work developed simulation-based definitions and constructions under standard assumptions [1]. Those results address more general message distributions and definitions than ours. We instead make a ROM entropy argument explicit and connect it to domain-separated serialization.

Our analysis has four principal limitations. First, residual entropy is a strong, application-dependent hypothesis; opening one coordinate of a secret- sharing or replicated vector may destroy it. Second, message length is public. Third, classical lazy sampling does not cover superposition queries to a quantum random oracle [7]. Fourth, the construction provides no equivocation against a party that already queried the eventual opening point. Applications needing universally composable commitments, extraction, or arbitrary correlated-message selective opening should use a construction proved for that stronger target.

9 Conclusion

We presented a self-contained domain-separated commitment and an adaptive selective-opening analysis for message distributions with residual min-entropy. The proof reduces security to a single operational question: could the adversary have guessed an unopened message–salt pair before the simulator programs it? Explicit typing turns that question into the bound $q_H n 2^{-(k+\rho)}$, while a standard birthday argument supplies binding. The same syntax also blocks cross-protocol, cross-session, and cross-position reinterpretation. The broader lesson is that domain separation belongs in the statement of a compositional theorem, not merely in an implementation note.

References

1. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer (2009)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73. ACM (1993)
3. Blum, M.: Coin flipping by telephone: A protocol for solving impossible problems. SIGACT News **15**(1), 23–27 (1983)
4. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 639–648. ACM (1996)
5. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 209–218. ACM (1998)
6. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions. Journal of the ACM **50**(6), 852–921 (2003)
7. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 755–784. Springer (2015)

A Expanded Hybrid Argument

For completeness, we spell out the hybrids implicit in Theorem 1. This appendix also clarifies what the simulator stores.

State. For each position i, the simulator stores $(sid, i, C_i, r_i, \mathsf{status}_i)$. Status is initially closed and changes once to opened. The random-oracle table maps canonical byte strings to τ-bit outputs. Repeated opening requests return the identical stored pair and do not reprogram the oracle.

Hybrid G_j. Positions $1, \ldots, j$ use uniform commitments and are programmed on opening. Positions $j+1, \ldots, n$ use honest commitments. Adaptive order is handled by relabeling positions according to their first opening time in the analysis; this relabeling is conceptual and does not change the bytes containing i.

Transition. To pass from G_{j-1} to G_j, replace

$$\mathsf{H}(\mathsf{Enc}(\mathsf{Tag}, sid, j, M_j; r_j))$$

with a uniform C_j. These variables have identical distributions unless the encoded point already occurs in the oracle table. At the eventual opening the game installs the same value, after which all future answers coincide. Thus the transition cost is precisely the probability of a prior query to that point. Lemma 3 bounds this probability after conditioning on every earlier adaptive decision.

No hidden independence assumption. The proof never assumes that the messages M_i are mutually independent. It uses only the pointwise bound in Definition 1 after conditioning on the values already opened. The salts, by contrast, must be fresh and mutually independent. If a salt is derived deterministically from the message, the additive entropy calculation fails.

B Minimal Test Vectors

An implementation can exercise the semantic structure without fixing a particular hash function. Let the tag be `ballot-v1`, the session be `election-2030`, the position be the integer 7, the message be the UTF-8 bytes for `yes`, and the salt be a 16-byte test value. A test harness should establish the following properties:

- serializing the tuple twice yields identical bytes;
- parsing and re-encoding yields the original byte string;
- changing any one field changes the encoded byte string;
- non-minimal integer encodings and truncated length fields are rejected;
- the same opening fails under position 8 and under a different session identifier; and
- a bit flip in the message or salt causes verification to reject.

These tests do not prove security, but they directly exercise the injectivity hypothesis used by Lemma 2 and Theorem 2.