mathematics

MDPI

Article

A Linkable Ring Signature Based on Polar Codes with Reusable Authorization

Zhiqiang Lin [1], Jingang Liu [2,*] and Qinrui Lv [1]

[1] School of Information Network Security, Xinjiang Institute of Political Science and Law, Tumxuk 843900, China; linzhiqiang@gzhu.edu.cn (Z.L.); 2026078@xjzfu.edu.cn (Q.L.)
[2] School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China
* Correspondence: liujingang@gpnu.edu.cn

Abstract

Post-quantum digital signatures are needed to protect authentication against quantum attacks. Code-based techniques offer a promising foundation for this goal. Linkable ring signatures additionally provide signer anonymity and public linkage, but repeated membership proofs can incur substantial communication costs. We propose a linkable ring signature based on polar codes with reusable anonymous authorization. The scheme supports independently verifiable messages, public linking across rings, and repeated signing. Under the stated primitive assumptions and honest registration, we prove unforgeability, restricted anonymity, linkability, and non-frameability in the classical random oracle model. For parameters selected against a 128-bit classical generic-attack target, the authorization proof is 98.52% shorter than the selected-matrix implementation of the same relation for a 32-member ring, and each cached online packet occupies 807 bytes. Compared with a direct-proof baseline using the same parameters, certificate reuse reduces total communication from the second message onward for all evaluated ring sizes. A prototype executes key generation, authorization, certificate installation, and online signing and verification with actual polar-code keys. For a 1024-member ring, the authorization certificate occupies 189.5 MB. For a single verifier with pre-distributed rings, total communication is lower than that of a reference linkable ring signature scheme for workloads of at least 6561 messages.

Keywords: post-quantum cryptography; polar codes; linkable ring signatures; anonymous authentication; MPC-in-the-head

MSC: 94A60; 94B05; 94B35; 68P25

Check for updates

Academic Editor: Mariana I. Durcheva

Received: 25 September 2026
Revised: 19 October 2026
Accepted: 20 October 2026
Published: 26 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Large-scale quantum computers threaten public-key authentication based on factoring and discrete logarithms, motivating the development of post-quantum digital signatures. The standardization of ML-DSA and SLH-DSA provides concrete alternatives [1,2], while NIST's additional-signature evaluation examines complementary constructions, including code-based and symmetric-primitive-based approaches [3]. Deploying these signatures also requires attention to public-key distribution, signature size, and recurring communication costs [4]. Applications involving anonymous reporting or repeated group authentication additionally require a verifier to authenticate a message without identifying its sender.

Ring signatures address this requirement by allowing a member to sign on behalf of a set of public keys while concealing the signing member [5]. Linkable ring signatures further

allow a verifier to recognize signatures associated with the same signer without revealing that signer's identity [6]. This functionality is useful when anonymous submissions must remain individually verifiable while repeated participation is detectable. Post-quantum constructions include the code-based scheme of Liu and Wang [7] and proof-based approaches. Goel et al. developed MPC-in-the-head set-membership proofs with applications to ring signatures [8]. Xue et al. proposed a framework for linkable ring signatures with post-quantum instantiations and reusable offline signing and verification work [9]. These results provide the context for reducing the recurring cost of anonymous authentication.

Code-based signatures offer several ways to build the underlying authentication primitive. CFS uses a secret Goppa-code decoding structure [10], whereas Wave uses preimage sampling over generalized ternary codes [11]. Wave parameter selection illustrates the trade-off between short signatures and large public keys [12]. Syndrome Decoding in the Head instead proves knowledge of a low-weight solution to a public decoding relation [13]. For ring signatures, the additional cost of hiding the signing key makes both the verification relation and the proof representation important design choices.

Repeated use must also be considered within an explicit anonymity model. Bultel and Olivier-Anclin studied anonymity definitions for linkable ring signatures [14], and Tran et al. introduced threshold-based many-time anonymity with a lattice-based construction in the quantum random oracle model [15]. We consider anonymous authentication with public linkage under honestly registered keys. Messages remain independently verifiable, and signatures from the same registered key remain linkable across different rings and online authorizations.

Polar codes are attractive for this setting because they combine a recursive encoding structure with efficient decoding algorithms [16]. Standard polar encoding and successive-cancellation decoding have $O(N \log N)$ complexity for block length N, and list decoding offers further decoding trade-offs [17]. Forghani et al. adapted modified polar codes to digital signatures in PolarSig [18]. We instantiate its modified-polar key family with a longer mother code and successive-cancellation randomized rounding for member authorization and online message signing. Its short basic signature limits recurring message expansion, while its binary parity-check verification relation provides the structure used in our authorization-circuit optimization.

The MPC-in-the-head paradigm supplies a general method for proving the authorization relation without revealing the member [19]. ZKBoo and ZKB++ provide practical Boolean-circuit instantiations [20,21]. Banquet reduces proof costs through specialized AES checks [22], and recent FAEST optimizations combine witness compression with improved VOLE-in-the-head analysis [23]. These approaches motivate adapting the proof representation to the computation being verified. In our binary verification circuit, selecting the public linear-map outputs instead of a hidden matrix reduces the number of nonlinear gates.

Our objective is to reduce communication when members repeatedly authenticate messages over persistent rings. We propose a linkable ring signature based on polar codes with reusable anonymous authorization. A member establishes an authorization before future messages are known. An accepting verifier can then reuse it for subsequent messages while preserving public linkage. The design combines a smaller authorization circuit with short online signatures. Its evaluation accounts for the initial certificate and public-ring distribution, so the reported savings reflect the complete repeated-authentication workload.

The contributions are as follows.

1. **Polar-code-based construction and security.** We design a linkable ring signature based on polar codes with reusable anonymous authorization. Under the stated primitive assumptions and honest registration, we prove unforgeability, restricted anonymity, linkability, and non-frameability in the classical random oracle

model. The scheme supports independently verifiable messages and public linking across rings.

2. **Smaller anonymous-authorization proofs.** We specialize standard linearity and one-hot selection to the authorization relation by evaluating public binary linear maps before selecting the hidden member's output. A hybrid circuit uses $d \min(L-1, n)$ syndrome AND gates, where L is the ring size and $d \times n$ is the public matrix dimension. For a 32-member ring, the fixed-length FS-ZKB++ proof is 98.52% shorter than the selected-matrix implementation of the same relation.
3. **Lower communication through authorization reuse.** We bind an online signing key and a persistent linking tag to a reusable anonymous certificate, allowing cached online packets of 807 bytes. Relative to a direct-proof baseline with the same parameters, certificate reuse reduces total communication from the second message onward for all four evaluated ring sizes.

Section 2 introduces the notation and primitives. Section 3 defines the security model and presents the construction. Sections 4 and 5 give its security analysis and circuit instantiation. Section 6 evaluates communication and the integrated prototype, and Section 7 concludes.

2. Preliminaries

2.1. Notation and Primitives

Let λ denote the security parameter, $[L] = \{0, \ldots, L-1\}$, and $\mathrm{wt}(e)$ the Hamming weight of a binary vector. Matrix operations are over $\mathbb{F}_2$. The notation $\langle \cdot \rangle$ denotes an injective, length-delimited encoding, and $\|$ denotes concatenation.

Polar codes and signatures. Polar codes are binary linear codes obtained through channel polarization. Their recursive structure supports encoding and successive-cancellation decoding in $O(N \log N)$ operations for block length N [16]. Successive-cancellation list decoding maintains multiple candidate paths to improve decoding performance [17]. PolarSig adapts modified polar codes to digital signatures [18]. Write $\mathsf{PS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ for its algorithms. A public key is a binary matrix $H \in \mathbb{F}_2^{d \times n}$, and a signature contains an error vector e and a random string a. Verification checks

$$He = h_{\mathrm{PS}}(M \| a) \tag{1}$$

and $W_\theta(e) = \mathbf{1}[\mathrm{wt}(e) \leq t]$, where t is the prescribed weight upper bound. PolarSig's EUF-CMA theorem is a parameterized reduction under the CSD and MPD hardness assumptions in the classical random oracle model [18] [Theorem 3].

Noninteractive zero-knowledge arguments. For a relation $\mathcal{R}$, an argument $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Vfy})$ allows a prover to establish that a public statement has a valid witness without revealing it. Completeness ensures acceptance of valid proofs, adaptive multi-theorem zero knowledge hides the witness, and simulation-sound extractability permits witness extraction from fresh accepting statements even in the presence of simulated proofs [24] [Definitions 4–5]. The extraction setup preserves the simulator's public-parameter and trapdoor distribution. Such arguments support anonymous membership and credential verification [8,24].

MPC-in-the-head and Fiat–Shamir. MPC-in-the-head constructs zero-knowledge proofs by simulating a multiparty computation and opening selected execution views [19]. ZKBoo and ZKB++ instantiate this approach for Boolean circuits [20,21]. The Fiat–Shamir transformation derives verifier challenges by hashing the public statement and complete first protocol message [25].

Further analyses address strong unforgeability and binding properties of MPC-in-the-head signatures [26], and Fiat–Shamir and commit-and-open protocols in the quantum random oracle model [27,28].

Hash commitments and domain separation. A randomized hash commitment has the form $C = H(\langle x, \rho \rangle)$, where ρ is a secret random salt. In the random oracle model, the salt hides the committed value, while collision resistance provides computational binding. Domain separation assigns distinct prefixes to different hash functions so that commitments, identifiers, and protocol challenges use separate hash domains.

2.2. Linkable Ring Signatures

A ring signature authenticates a message under a set of public keys without identifying the signing member [5]. A linkable ring signature additionally provides a public test for whether two valid signatures originate from the same signing key [6]. With common public parameters implicit, its algorithmic interface is

$$\begin{aligned}(pk, sk) &\leftarrow \mathsf{LRS.KeyGen}(1^\lambda),\\ \sigma &\leftarrow \mathsf{LRS.Sign}(sk_i, R, m),\\ v &\leftarrow \mathsf{LRS.Verify}(R, m, \sigma) \in \{0,1\},\\ b &\leftarrow \mathsf{LRS.Link}(X_0, X_1) \in \{0,1,\perp\},\end{aligned}$$

where $R = (pk_0, \ldots, pk_{L-1})$ is a ring containing pk_i, and $X_b = (R_b, m_b, \sigma_b)$ is a ring–message–signature tuple. Verification returns 1 for an accepted signature. Linking returns 1 for linked signatures, 0 for unlinked signatures, and $\perp$ if either signature is invalid. The linking scope specifies which pairs of rings or application contexts are eligible for comparison.

Verification correctness. For honestly generated keys, every message m, and every ring R containing pk_i,

$$\Pr\big[\mathsf{LRS.Verify}\big(R, m, \mathsf{LRS.Sign}(sk_i, R, m)\big) = 1\big] \geq 1 - \mathrm{negl}(\lambda).$$

The probability is over key generation and signing randomness.

Linking correctness. Let $\sigma_b \leftarrow \mathsf{LRS.Sign}(sk_{i_b}, R_b, m_b)$ for $b \in \{0,1\}$, with both tuples in the same linking scope. For distinct, honestly generated member keys, valid signatures satisfy

$$\Pr[\mathsf{LRS.Link}(X_0, X_1) = \mathbf{1}[i_0 = i_1]] \geq 1 - \mathrm{negl}(\lambda),$$

where i_0, i_1 identify member keys, and $\mathbf{1}[\cdot]$ is the indicator function.

Anonymity hides the signing member subject to the information disclosed by public linking [14]. Unforgeability prevents unauthorized message signatures. Linkability prevents a signer from evading the linking test. Non-frameability prevents an adversary from producing a fresh signature attributed by linking to an honest member.

3. Proposed Construction

This section specifies the security model, authorization relation, and algorithms of the polar-code-based linkable ring signature.

3.1. Security Model

We consider a managed group with honestly provisioned, immutable registered keys. A ring is a canonical ordered set of distinct registered keys. θ fixes the parameters and encoding. The certificate cache contains accepted records bound to their ring and parameter identifier. Linking is global per registered key.

Classical probabilistic polynomial-time adversaries access the registration, authorization, signing, and corruption oracles in Appendix A. Let $\mathcal{C}$, $\mathcal{Q}$, and $\mathcal{T}_{\text{issued}}$ denote corrupted member indices, answered ring-message pairs, and issued authorization tags, respectively.

Unforgeability. A winning forgery is an accepted signature on (R^*, m^*) with $\text{ids}(R^*) \cap \mathcal{C} = \varnothing$ and $(R^*, m^*) \notin \mathcal{Q}$, where $\text{ids}(R)$ denotes its registered member indices.

Anonymity. A uniform bit b selects one of two uncorrupted candidates $i_0, i_1 \in R$ for the challenge authorization and its messages. The adversary may query further challenge messages and other members, but may neither corrupt nor obtain identity-labeled authorizations or signatures from either candidate before or after the challenge. Its advantage is $|\Pr[b' = b] - 1/2|$. Tag and certificate equality remain visible. This model covers managed anonymous reporting in which credentials are provisioned privately and submissions are exposed through pseudonymous tags. An identity-labeled sample from either candidate would identify its persistent tag and all linked submissions.

Linkability. Signatures from the same registered key link. An adversary wins by outputting accepted signatures with more than $|\mathcal{C}|$ distinct tags outside $\mathcal{T}_{\text{issued}}$.

Non-frameability. Given an uncorrupted target's tag and signatures, the adversary wins by producing an accepted signature with that tag on a fresh ring-message pair. The target's long-term and online state remains unexposed.

3.2. Overall Framework

The construction anonymously certifies an online key and reuses its certificate for message authentication (Figure 1). Its interface is

$$\begin{aligned}
(PK, SK) &\leftarrow \mathsf{KeyGen}(1^\lambda; \theta),\\
(\mathsf{Cert}, \mathsf{st}) &\leftarrow \mathsf{Authorize}(SK, R; \theta),\\
D \text{ or } \perp &\leftarrow \mathsf{Install}(R, \mathsf{Cert}, \mathsf{Cache}; \theta),\\
\omega &\leftarrow \mathsf{SignOnline}(\mathsf{st}, m; \theta),\\
v &= \mathbf{1}[\mathsf{VerifyTag}(R, m, \Sigma; \theta) \neq \perp],\\
b &\leftarrow \mathsf{Link}(X_0, X_1).
\end{aligned}$$

For a standalone signature $\Sigma = (\mathsf{Cert}, \tau)$, VerifyTag installs the certificate in an empty cache and invokes VerifyOnline. An installed certificate supports the compact packet $\omega = (D, \tau)$. Boolean verification returns v. Linking uses the accepted tag.

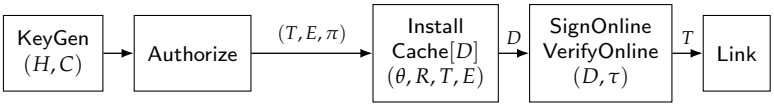


Figure 1. Scheme workflow.

During registration, Algorithm 1 associates a polar-code verification key with a commitment to a linking secret. Algorithm 2 proves that one hidden registered member both possesses this secret and authorizes an independently generated online key. The proof uses the same hidden member index for both checks. This binds the online key to an authorized member while hiding that member's identity.

Algorithm 3 verifies the authorization and installs its public record. Algorithm 4 uses the authorized polar-code key to sign and verify each message. Its signed encoding includes the certificate identifier and protocol context, preventing a message signature from being transferred to a different authorization. The short polar-code signature is the recurring communication cost, while the membership proof is paid for when the certificate is installed.

The public transcript exposes the ring, messages, online public key, certificate identifier, and linking tag. Equal tags reveal use of the same registered member key. Equal certificate identifiers additionally reveal reuse of an online authorization. The linking tag is derived from the registered member's secret, independently of the ring and online key. Commitment binding prevents one registered member from obtaining different valid tags, and the joint proof binds every accepted tag to its authorized member. After message verification, Equation (5) compares the tags to implement public linking.

3.3. Authorization Relation

The domain-separated ℓ-bit hashes H_C, H_T, H_A, H_M implement commitments, tags, authorization identifiers, and direct-baseline message digests. Let Π satisfy the properties in Section 2.1.

Member i registers $PK_i = (H_i, C_i)$ and keeps $SK_i = (sk_i, x_i, \rho_i)$, where

$$C_i = H_C(\langle x_i, \rho_i \rangle), \qquad T_i = H_T(\langle x_i \rangle). \tag{2}$$

Let E be a freshly generated online PolarSig public key. Define

$$D = H_A(\langle \theta, R, T, E \rangle), \qquad B_\theta(D) = \langle \texttt{member}, \theta, D \rangle. \tag{3}$$

For the statement $S = (\theta, R, T, D)$ and witness $w = (j, x, \rho, \sigma)$, the authorization relation is

$$\mathcal{R}_{\text{auth}}(S, w) = 1 \iff \begin{cases} j \in [L], \quad R_j = (H_j, C_j), \\ H_C(\langle x, \rho \rangle) = C_j, \\ H_T(\langle x \rangle) = T, \\ \mathsf{PS.Verify}(H_j, B_\theta(D), \sigma) = 1. \end{cases} \tag{4}$$

The same index j selects both the commitment and verification matrix.

3.4. Key Generation and Authorization

Algorithms 1 and 2 generate registered keys and anonymous certificates.

Algorithm 1 Member key generation

Require: $1^\lambda, \theta$
Ensure: $PK = (H, C)$, $SK = (sk, x, \rho)$
1: $(H, sk) \leftarrow \mathsf{PS.Gen}(1^\lambda; \theta)$
2: $x, \rho \xleftarrow{\$} \{0,1\}^\ell$
3: $C \leftarrow H_C(\langle x, \rho \rangle)$
4: **return** $((H, C), (sk, x, \rho))$

Algorithm 2 Anonymous authorization

Require: $SK_j = (sk_j, x_j, \rho_j), R, \theta, pp$
Ensure: $\mathsf{Cert} = (T, E, \pi)$, $\mathsf{st} = (D, esk)$
1: $(E, esk) \leftarrow \mathsf{PS.Gen}(1^\lambda; \theta)$
2: $T \leftarrow H_T(\langle x_j \rangle)$
3: $D \leftarrow H_A(\langle \theta, R, T, E \rangle)$
4: $\sigma \leftarrow \mathsf{PS.Sign}(sk_j, B_\theta(D))$
5: $S \leftarrow (\theta, R, T, D), \quad w \leftarrow (j, x_j, \rho_j, \sigma)$
6: $\pi \leftarrow \Pi.\mathsf{Prove}(pp, S, w)$
7: **return** $((T, E, \pi), (D, esk))$

Setup generates $pp \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$ and θ. Authorization precedes application messages.

3.5. Certificate Installation and Online Signing

Algorithm 3 validates and caches certificates. Algorithm 4 authenticates messages under the certified online key.

Algorithm 3 Certificate installation

Require: R, Cert, Cache, θ, pp
Ensure: D or $\perp$; updated Cache
1: $(T, E, \pi) \leftarrow$ Cert
2: **if** $\mathsf{ValidEnc}_\theta(R, T, E, \pi) = 0$ **then**
3: **return** $\perp$
4: **end if**
5: $D \leftarrow H_A(\langle \theta, R, T, E \rangle)$
6: **if** $\Pi.\mathsf{Vfy}(pp, (\theta, R, T, D), \pi) = 0$ **then**
7: **return** $\perp$
8: **end if**
9: $\mathsf{Cache}[D] \leftarrow (\theta, R, T, E)$
10: **return** D

Algorithm 4 Online signing and verification

procedure SIGNONLINE($(D, esk), m; \theta$)
$M \leftarrow \langle \texttt{online}, \theta, D, m \rangle$
$\tau \leftarrow \mathsf{PS.Sign}(esk, M)$
return (D, τ)
end procedure

procedure VERIFYONLINE($R, m, (D, \tau), \mathsf{Cache}; \theta$)
if $D \notin \mathrm{dom}(\mathsf{Cache})$ **then**
return $\perp$
end if
$(\theta', R', T, E) \leftarrow \mathsf{Cache}[D]$
if $(\theta', R') \neq (\theta, R)$ **then**
return $\perp$
end if
$M \leftarrow \langle \texttt{online}, \theta, D, m \rangle$
if $\mathsf{PS.Verify}(E, M, \tau) = 0$ **then**
return $\perp$
end if
return T
end procedure

$\mathsf{ValidEnc}_\theta$ checks the header, parameter identifier, ring-size bounds, and exact field lengths; canonical distinct registered keys; registered matrix dimensions and full row rank; online-key dimensions; ℓ-bit commitments and tags; and prescribed proof seeds, commitments, views, ternary challenges, and zero padding. Extra bytes are rejected.

3.6. Public Linking

For two ring-message-signature tuples X_0, X_1, the linking procedure resolves their certificates and verifies each message:

$$\mathsf{Link}(X_0, X_1) = \begin{cases} \perp, & \mathsf{VerifyTag}(X_0) = \perp \ \vee \ \mathsf{VerifyTag}(X_1) = \perp, \\ [T_0 = T_1], & \text{otherwise.} \end{cases} \tag{5}$$

Equation (2) preserves the tag across rings and online keys.

4. Scheme Analysis

We establish correctness and the four security properties defined in Section 3.1.

4.1. Correctness

Proposition 1. *If the underlying signing algorithms return valid signatures and* Π *is complete, the construction is correct and signatures from the same member link.*

Proof. Algorithm 2 satisfies all conditions in Equation (4). Completeness and base-signature correctness give acceptance, while Equation (2) fixes each member's tag. For U registered members, repeated linking secrets and distinct-input tag-hash collisions each have probability at most $U(U-1)/2^{\ell+1}$. Their union bounds accidental cross-member tag equality by $U(U-1)/2^{\ell}$. The construction inherits the base signing failure probability. □

4.2. Unforgeability

Theorem 1. *Assume that the polar-code signature is EUF-CMA secure,* Π *satisfies Section 2.1, and* H_A *is collision resistant. Then, the construction is unforgeable in the classical random oracle model under Section 3.1.*

Proof. Simulate the challenger's authorization proofs, at cost Δ_{ZK}, and charge H_A collisions to δ_A. An unissued payload gives a fresh statement. Apply the extractor to a wrapper returning this authorization alone. Embedding a base-signature challenge key at a uniform registered index requires no member-signing query because authorization proofs are simulated. If the guess matches the extracted index, σ forges $B_\theta(D)$. The winning ring excludes corruption of this member.

For an issued payload, guess its online key among the issued authorizations and answer its signing queries through the base signing oracle. Freshness of (R, m) makes $\langle \texttt{online}, \theta, D, m \rangle$ fresh under that key, yielding the forgery τ.

Let $\epsilon^{\Pi}_{\mathrm{SE}}(\lambda, T_W, \mathbf{Q}_W, N_{\mathrm{sim}}; T_E, \mathbf{Q}_E)$ bound the joint event of a fresh accepting wrapper output and invalid extraction in the simulated experiment. Here, N_{sim} bounds simulated proofs and $\mathbf{Q}$ separates hash domains. Let $\epsilon_{\mathrm{PS}}(T, Q, q_s)$ denote the maximum base-signature EUF-CMA advantage within time T, Q base-hash queries, and q_s signing queries. For at most U members and $A \geq 1$ authorizations,

$$\begin{aligned} \mathrm{Adv}_{\mathrm{UF}} \leq \Delta_{\mathrm{ZK}} + \delta_A + \epsilon^{\Pi}_{\mathrm{SE}} \\ + U\epsilon_{\mathrm{PS}}(T_1, Q_1, 0) + A\epsilon_{\mathrm{PS}}(T_2, Q_2, q_{\mathrm{sign}}). \end{aligned} \tag{6}$$

The two guesses cost factors U and A, respectively, with budgets

$$\begin{aligned} T_1 &= T_{E,W} + T_{\mathrm{post}}, & Q_1 &= Q^{\mathrm{PS}}_{E,W} + Q^{\mathrm{PS}}_{\mathrm{post}}, \\ T_2 &= T_{\mathcal{B}} + T^{(2)}_{\mathrm{env}} + T_{\mathrm{post}}, & Q_2 &= Q^{\mathrm{PS}}_{\mathcal{B}} + Q^{(2),\mathrm{PS}}_{\mathrm{env}} + Q^{\mathrm{PS}}_{\mathrm{post}}. \end{aligned}$$

$T_{E,W}, \mathbf{Q}_{E,W}$ include the extractor, wrapper, registration, simulation, and oracle environment over all rewinding executions. The online environment budgets include registration, other key generation, simulation, verification, and state maintenance. Postprocessing covers witness checking and output. Superscript PS selects base-hash queries, and q_{sign} counts signing queries under the guessed online key. □

4.3. Anonymity

Theorem 2. *Under adaptive zero knowledge of* Π *and the domain-separated random-oracle assumptions for* H_C *and* H_T*, the construction satisfies the anonymity game of Section 3.1.*

Proof. Simulate all authorization proofs needed by the game. The independent commitment and tag domains hide the candidate-to-tag correspondence until an unexposed linking secret is queried. Replacing the challenge tag by a uniform value and switching candidates

preserves the independent online-key and message distributions, including their public equality information. Thus

$$\mathrm{Adv}_{\mathrm{ANON}} \le \Delta_{\mathrm{ZK}}^{\mathrm{anon}} + \delta_{\mathrm{secret}}. \tag{7}$$

Let $q = q_C + q_T$ include pre-registration queries. The secret-query event matches any unexposed registered x, or a later registration secret matches an earlier query. An incorrect-salt commitment query counts as an x guess. Conditional on distinct secrets, no public-hash collision, and no earlier hit, each unexposed secret is uniform over at least $2^\ell - q - U$ possibilities. A union bound over registration and query steps yields

$$\delta_{\mathrm{secret}} \le \frac{Uq}{2^\ell - q - U} + \frac{U(U-1)}{2^{\ell+1}} + \delta_{\mathrm{col}}, \qquad q + U < 2^\ell.$$

For $q + U \le 2^{\ell-1}$ the first term is at most $2Uq/2^\ell$. With N_h distinct inputs in domain $h \in \{C, T, A\}$, $\delta_{\mathrm{col}} \le \sum_h N_h(N_h - 1)/2^{\ell+1}$. Query budgets include challenger and verifier evaluations. Proof-programming losses enter $\Delta_{\mathrm{ZK}}^{\mathrm{anon}}$. □

4.4. Linkability and Non-Frameability

Theorem 3. *Under the assumptions of Theorem 1, commitment binding and tag collision resistance, the construction satisfies linkability and non-frameability as defined in Section 3.1.*

Proof. The challenger knows each registered opening and canonical tag, including those of subsequently corrupted members. A winning linkability output contains an accepted tag outside both $\mathcal{T}_{\mathrm{issued}}$ and $\{T_j : j \in \mathcal{C}\}$. A wrapper verifies the outputs and selects the first such authorization. Its statement is fresh because its tag was not issued. Extract once. An uncorrupted extracted member yields the member-level forgery of Theorem 1. A corrupted member has $x \ne x_j$, so its extracted and registered openings collide under H_C. The joint extraction-failure term is $\epsilon_{\mathrm{SE}}^{\Pi}$. For N outputs, one wrapper run costs $T_{\mathcal{B}} + NT_{\mathrm{VerifyTag}} + T_{\mathrm{lookup}}$. Verification queries enter $\mathbf{Q}_W$ and all extractor rewinding enters $T_{E,W}, \mathbf{Q}_{E,W}$.

For non-frameability, consider a fresh accepted message with the target tag. Outside registered-secret and tag collisions, an issued payload carrying that tag belongs to the target and yields an online forgery. A new payload admits fresh-statement extraction: the target member yields a member-level forgery, while another member yields commitment or tag collision. Repeated registered secrets contribute at most $U(U-1)/2^{\ell+1}$. □

5. Authorization Circuit and Instantiation

This section realizes the authorization relation as a Boolean circuit and instantiates its proof with Fiat–Shamir ZKB++.

5.1. Selecting Public Linear-Map Outputs

Let $L = 2^s$ and encode the hidden index by s witness bits. A binary decoder produces $b_i = [j = i]$. At each internal node, the two children can be computed as vz and $v \oplus vz$. Excluding the first split, the decoder requires $L - 2$ AND gates. The selected commitment is

$$C_j = \bigoplus_{i=0}^{L-1} b_i C_i. \tag{8}$$

Since every C_i is public, Equation (8) is linear in the witness bits. For the syndrome, use

$$H_j e = H_0 e \oplus \bigoplus_{i=1}^{L-1} b_i \big((H_i \oplus H_0)e\big). \tag{9}$$

The products $(H_i \oplus H_0)e$ are public linear maps and use only XORs. Each of their d output coordinates then requires one AND with b_i. Algorithm 5 evaluates the hidden-member syndrome.

Algorithm 5 Hidden-member syndrome evaluation

Require: $R = ((H_i, C_i))_{i\in[L]}, j, e$
Ensure: $y = H_j e$
1: $(b_0, \ldots, b_{L-1}) \leftarrow \mathsf{Decode}(j)$
2: $y \leftarrow H_0 e$
3: **for** $i = 1, \ldots, L-1$ **do**
4: $\quad z_i \leftarrow (H_i \oplus H_0)e$
5: $\quad y \leftarrow y \oplus (b_i z_i)$
6: **end for**
7: **return** y

Proposition 2. *Equation (9) computes $H_j e$ with $d(L-1)$ AND gates after index decoding, compared with dn for selecting a secret matrix and multiplying it by e.*

Proof. For $j = 0$, all terms indexed by $i \geq 1$ vanish. Otherwise, only term $i = j$ remains, giving $H_0 e \oplus (H_j \oplus H_0)e = H_j e$. The stated gate count follows from $L-1$ coordinatewise products of length d. In the selected-matrix baseline, each of the dn entries multiplies a secret coordinate of e. □

The reduction is strict when $L < n+1$. The decoder cost $L-2$ is common to both circuits. Selecting the smaller syndrome circuit gives $d\min(L-1, n)$ AND gates, with $O(Ldn)$ public XOR work. This specializes standard linearity and one-hot membership selection [8].

5.2. Hash and Weight Constraints

The witness is (j, x, ρ, e, a) with length

$$I(L) = \log_2 L + 2\ell + n + |a|. \tag{10}$$

The circuit outputs commitment, tag, and syndrome residuals, together with the failure bit of the prescribed weight test. Its public target is the all-zero vector of length $d + 2\ell + 1$. Using this vector avoids a final conjunction over all comparison bits.

We select SHAKE256 [29] for the concrete hash interfaces. The commitment and tag inputs occupy 65 and 33 bytes. The authorization message has a four-byte prefix and a 32-byte digest. Its 20-bit random string occupies three bytes with four zero padding bits. The hash h_{PS} returns the first 4179 bits of SHAKE256, requiring four 1088-bit output blocks. Each private-input hash fits one absorption block. The commitment and tag each use one Keccak-f[1600] permutation, and h_{PS} uses four, including the additional squeeze permutations. The six permutations contribute 230,400 AND gates, or 61.22% of the $L = 32$ budget. Public authorization hashing and Fiat–Shamir transcript hashing are outside the witness circuit.

For weight evaluation, pad e to $B = 2^{\lceil \log_2 n \rceil}$ bits and use a balanced addition tree. A full-adder carry is

$$c' = u \oplus \big((u \oplus v) \wedge (u \oplus c)\big). \tag{11}$$

The tree uses $2B - \log_2 B - 2$ AND gates. A constant comparison uses at most $\log_2 B + 1$ additional gates. With $n = 6144$, the budget before constant-dependent gate elimination is 16,383 gates.

5.3. Parameters and Proof Encoding

Table 1 specifies the modified-polar instance used for member and online keys. We select it against a 128-bit classical generic-attack target.

Table 1. Reference parameters.

Parameter	Value
Mother-code length N_0	8192
Base information/additional unfrozen positions	1720/245
Punctured positions	2048
Code length n/dimension k/syndrome length d	6144/1965/4179
Weight upper bound t	1600
Design channel/Bhattacharyya samples	BSC(0.24)/300
PolarSig random string a/signature	20 bits/771 bytes
x, ρ, C, T, D	256 bits each
Proof field/parallel repetitions r	$\mathbb{F}_2$/329
View seed/commitment	256/256 bits
Packed online public key K	3,209,472 bytes
Packet header b	4 bytes

Finite-length key generation and signing. In $\mathsf{Gen}_{\mathrm{MC}}$, the information set consists of the 1720 smallest estimated Bhattacharyya parameters for BSC(0.24), obtained from 300 Monte Carlo synthetic-channel samples. Starting from the polar transform $F^{\otimes 13}$, key generation uniformly unfreezes 245 other positions and punctures 2048 coordinates, resampling punctures until the generator has rank 1965. For a full-rank parity-check matrix H', the public key is $H = SH'P$, where S is uniform in $\mathrm{GL}(4179, 2)$ and P is a uniform coordinate permutation. These are the frozen-set randomization, puncturing, and masking operations of the modified-polar family [18].

Signing first inverts S on the hashed syndrome and forms a systematic preimage under H'. It randomly fills punctured coordinates and applies successive-cancellation randomized rounding for BSC(0.24) [30]. The original frozen positions remain zero, except that the 245 additionally unfrozen positions receive independent random bits. The punctured quantization error is permuted back to public coordinates. A fresh 20-bit salt and fresh rounding randomness are sampled until its weight is at most 1600.

Concrete generic-attack estimates. An invertible d-column submatrix H_J gives the public forgery trial

$$e_J = H_J^{-1} h_{\mathrm{PS}}(M \| a), \qquad e_{[n]\setminus J} = 0.$$

Its acceptance probability for a fresh hash input is

$$p(d, t) = 2^{-d} \sum_{w=0}^{t} \binom{d}{w}.$$

The parameters of Table 1 give $-\log_2 p = 171.89$. General decoding estimates must account for the many solutions: at weight t, their expected number is $\binom{n}{t}/2^d \approx 2^{897.96}$. Table 2 reports the evaluated classical attacks. The fixed-syndrome values use CryptographicEstimators 2.1.1 with this solution count [31]. The DS-DOOM value uses the authors' multi-target estimator at 2^{128} targets [32]. The collision-DOOM model [33] is also evaluated with unlimited free targets and zero target-conversion cost. Within its searched

auxiliary-length range $\ell_D \leq 512$, summing all weights up to t improves the relevant collision probability by at most a factor 4.42. The table conservatively subtracts 2.15 bits from that model's 145.20-bit estimate.

Table 2. Classical generic-attack estimates.

Attack Model	$\log_2$ Work	Unit
Public invertible-submatrix trial	171.89	Hash queries
Prange/Stern/Dumer	207.50/172.96/172.38	Bit operations
May–Ozerov/BJMM	169.31/165.80	Bit operations
DS-DOOM, 2^{128} targets	167.48	Bit operations
Collision-DOOM, free unlimited targets, wt $\leq t$	143.05	Model column operations

Instantiation assumptions and structural attacks. Theorems 1–3 hold under the CSD and MPD assumptions for the key distribution specified above and the stated proof-system and hash assumptions. Bardet et al. recover permuted polar codes using minimum-weight codewords and recursive code equivalence [34]. Random unfreezing of 245 positions and puncturing of 2048 coordinates of the $N_0 = 8192$ mother code modify the information set and full-length structure used by that attack [18]. The masks S and P preserve permutation-invariant code properties. The published attack does not directly establish a key-recovery cost for this modified distribution. Structural security remains conditional on its MPD assumption. Table 2 evaluates classical generic attacks. Its 128-bit target is not an established concrete security level against structural or distribution-specific attacks.

FS-ZKB++ instantiation. The public circuit target is $y_{\text{pub}} = 0^{d+2\ell+1}$. Each repetition uses independent uniform 256-bit seeds. A view commitment binds the seed and canonical view to the parameter identifier, statement, repetition, and party indices. Seed expansion, commitments, and challenges have disjoint domains, separate from $H_C, H_T, H_A, H_M, h_{\text{PS}}$. Proof simulation programs only proof-system domains. The Fiat–Shamir hash binds the statement and complete first message, including all output shares and view commitments. Uniform ternary challenges are sampled by rejecting two-bit value 11. The verifier reconstructs the opened views, obtains the unopened output share by XORing the public target with the two opened shares, and checks the recomputed challenge against the encoded vector. This uses ZKB++ output-share compression [21] and the classical-ROM adaptive zero-knowledge and simulation-extraction results of [24] [Lemmas 3–4 and Corollary 1].

For $r = 329$, the knowledge-error term $(Q+1)(2/3)^r$ is below 2^{-128} when $Q \leq 2^{64}$ [35]. Seed-and-view commitments give quasi-unique-response error at most $Q_{\text{com}}^2/2^{256}$ [24] [Lemma 4]. The extraction resources are represented separately in Equation (6).

A repetition contains one unopened-view commitment, two opened-view seeds, one AND transcript, and a fixed input-share field. The challenge vector uses two bits per repetition. Unused input-share fields and trailing padding bits are checked zeros. Using the smaller syndrome circuit at each ring size,

$$G(L) = 230400 + (L-2) + 4179\min(L-1, 6144) + 16383. \tag{12}$$

Applying the ZKB++ view encoding [21] with byte-aligned, fixed-length fields gives

$$P(L) = \left\lceil \frac{2r}{8} \right\rceil + r\left(96 + \left\lceil \frac{G(L)}{8} \right\rceil + \left\lceil \frac{I(L)}{8} \right\rceil\right). \tag{13}$$

The four-byte packet header identifies the version, parameter set, ring-size exponent, and record type. Matrices are packed continuously by rows. Equation (13) fixes the proof length.

6. Performance Analysis and Comparison

This section quantifies the communication cost of authorization and online authentication. We compare the optimized circuit with a selected-matrix baseline and derive the savings from certificate reuse. We then evaluate the complete protocol and compare repeated-authentication structures with Liu and Wang [7] and Xue et al. [9], using Xue et al.'s published reference for the numerical communication comparison.

6.1. Circuit Comparison

Table 3 compares Equation (9) with the selected-matrix baseline. Both use the same relation, hashes, repetition count, and proof format. The baseline replaces the syndrome cost $d(L-1)$ by dn.

Table 3. Authorization proof sizes.

Ring Size L	Proposed AND Gates	Baseline Proof (Bytes)	Proposed Proof (Bytes)	Reduction
8	276,042	1,066,371,991	11,658,856	98.91%
32	376,362	1,066,373,307	15,784,845	98.52%
128	777,642	1,066,377,255	32,287,485	96.97%
1024	4,522,922	1,066,414,103	186,312,125	82.53%

6.2. Certificate Reuse

Let s_T, c, s_σ be the byte lengths of the tag, certificate identifier, and basic signature. A certificate occupies

$$C_{\text{cert}} = b + s_T + K + P, \tag{14}$$

and an online packet occupies $b + c + s_\sigma$. In the reference instance, these are $P + 3,209,508$ and 807 bytes. A standalone signature carrying the certificate occupies $C_{\text{cert}} + s_\sigma$.

The direct-proof baseline signs each public digest $H_M(\langle \theta, R, T, m \rangle)$ under the hidden member key and proves Equation (4) for that digest. Its circuit and proof length are identical to the authorization circuit. Both schemes share the public ring, and the baseline sends its tag only once. For q messages,

$$\begin{aligned} L_{\text{direct}}(q) &= b + s_T + q(b + P), \\ L_{\text{cached}}(q) &= b + s_T + K + P + q(b + c + s_\sigma). \end{aligned} \tag{15}$$

Proposition 3. *Cached authorization reduces total communication relative to the direct-proof baseline exactly when*

$$(q-1)P > K + q(c + s_\sigma). \tag{16}$$

Proof. Subtract the second expression in Equation (15) from the first and cancel the common header and tag terms. □

When $P > c + s_\sigma$, the minimum integer satisfying Equation (16) is

$$q_{\min} = \left\lfloor \frac{K+P}{P-c-s_\sigma} \right\rfloor + 1. \tag{17}$$

It equals two for all four instances in Table 3. For $L = 32$ and $q = 2$, the direct-proof and cached constructions require 31,569,734 and 18,995,967 bytes, respectively, a 39.83% reduction. Table 4 includes the initial certificate in the per-message averages.

Table 4. Certificate and amortized communication (bytes).

L	Certificate	Standalone Signature	Average, $q = 128$	Average, $q = 1024$
8	14,868,364	14,869,135	116,966.09	15,326.89
32	18,994,353	18,995,124	149,200.38	19,356.17
128	35,496,993	35,497,764	278,127.26	35,472.03
1024	189,521,633	189,522,404	1,481,444.76	185,886.72

Public-ring distribution excluded.

For h certificate transmissions, total communication is $hC_{\text{cert}} + q(b + c + s_\sigma)$. Table 5 summarizes the cryptographic operations.

Table 5. Primitive calls.

Operation	Cached Authorization Construction
Authorize	1PS.Gen + 1PS.Sign + 1Π.Prove
Install	1Π.Vfy
Sign each message	1PS.Sign
Verify each message	1PS.Verify and a cache lookup

6.3. Experimental Evaluation

We evaluate signing performance and complete protocol execution using the parameters in Table 1. The prototype implements polar-code key generation and signing, authorization, certificate installation, and cached online signing and verification.

Experimental setup. The prototype uses Python 3.12.14 and NumPy 2.3.5 on Windows and a C++17 ZKB++ prover and verifier on Ubuntu 26.04 (WSL2), compiled with GCC 15.2.0 (`-O2 -std=c++17`). Measurements use one worker pinned to logical CPU 0 of an Intel Xeon Gold 5218R at 2.10 GHz. Protocol timings include serialization, process launch, and file I/O.

Test inputs. Eight independently generated, full-rank masked member keys form a canonical ring. One member creates three authorizations, each with an independently generated online key, and signs 32 distinct messages per authorization. Each authorization digest is computed from the actual parameter identifier, ring, tag, and online key as in Equation (3). Installation verifies the proof using only the public statement and stores the resulting online key and tag.

Results. All three authorizations installed successfully, and all 96 online signatures verified and linked to the same member tag. Every proof contained 11,658,856 bytes, matching Equation (13), and every online packet contained 807 bytes. Table 6 reports the measurements. The independent base-signing test produced 140 valid signatures on three masked keys in 187 attempts, averaging 1.34 attempts with a maximum of 4. Additional correctness and tampering checks passed. Peak RSS of the C++ proof processes was 98.81 MiB for proving and 66.94 MiB for verification.

Table 6. Integrated execution times.

Operation	Median (ms)	Min–Max (ms)	Samples
Registered key generation	27,866.67	20,537.12–53,416.39	8
Authorize	49,604.94	46,184.72–67,599.10	3
Install	7363.98	7293.43–7449.19	3
SignOnline	77.65	69.56–308.12	96
VerifyOnline	4.09	3.24–6.50	96

The results demonstrate complete protocol execution and certificate reuse for repeated, publicly linkable message authentication at $L = 8$. For fixed code parameters, proving and proof-verification work scales as $O(r[Ldn + G(L)])$, which is linear in L at fixed r. Larger-ring proof sizes and communication costs in Tables 3 and 4 are derived analytically from the circuit and serialization formulas.

6.4. Published-Scheme Comparison

Liu and Wang [7] bind each Stern-type proof transcript to its message. Xue et al. [9] reuse offline signing and verification work, while each signature retains a message-bound signature of knowledge. Our authorization certificate binds an independent online key before subsequent messages are known. Table 7 summarizes these published formats. Liu and Wang report numerical parameters at 80-bit security, so we use their scheme for this structural comparison.

Table 7. Repeated-authentication structure.

Scheme	Membership Evidence Transmitted	Reused Across Messages
Liu and Wang [7]	Message-bound Stern-type proof	Public keys and parameters
Xue et al. [9]	Message-bound signature of knowledge	Offline signing and verification work
Proposed	Certificate once, then identifier and basic signature	Installed certificate and online key

For the numerical comparison, our instance uses the 128-bit classical generic-attack target and CSD/MPD assumptions of Section 5.3, and Xue et al. use their published 128-bit configuration. For $L = 1024$, we use their approximate 29 KiB reference as $S_X = 29{,}696$ bytes [9,36]. With the hybrid circuit, Equations (12) and (13) give $G = 4{,}522{,}922$, a 186,312,125-byte proof, a 189,521,633-byte certificate, and a 189,522,404-byte standalone signature. The public ring occupies

$$C_R = 4 + L(K + 32) = 3{,}286{,}532{,}100 \text{ bytes}. \tag{18}$$

Table 8 gives analytical communication costs with the full initial certificate and matching public-key-list distribution: both lists are available, or each is sent once. All messages reuse one certificate at one recipient. Xue et al.'s reported 32-byte public keys give $C_{R,X} = 32L$ bytes.

Table 8. Communication comparison.

	Both Lists Already Available		Each List Distributed Once	
Messages q	**Proposed**	**Xue et al. [9]**	**Proposed**	**Xue et al. [9]**
128	1,481,444.76	29,696.00	27,157,476.79	29,952.00
1024	185,886.72	29,696.00	3,395,390.72	29,728.00
8192	23,941.96	29,696.00	425,129.97	29,700.00
65,536	3698.87	29,696.00	53,847.37	29,696.50
131,072	2252.94	29,696.00	27,327.19	29,696.25
262,144	1529.97	29,696.00	14,067.09	29,696.13

Bytes per message. $L = 1024$, one recipient, one certificate installation.

Let s count list distributions under the same convention for both schemes, and h count certificate installations. Then

$$\begin{aligned} L_P(q; s, h) &= sC_R + hC_{\text{cert}} + 807q, \\ L_X(q; s) &= sC_{R,X} + qS_X. \end{aligned} \tag{19}$$

For $S_X > 807$ and $h \geq 1$, the strict crossover is

$$q_{\min} = \left\lfloor \frac{hC_{\text{cert}} + s(C_R - C_{R,X})}{S_X - 807} \right\rfloor + 1.$$

With $h = 1$, $S_X = 29{,}696$, and $C_{R,X} = 32{,}768$, the thresholds at one recipient are 6561 messages for $s = 0$ and 120,324 for $s = 1$. At $q = 131{,}072$ with each ring distributed once, the calculated communication is 27,327.19 versus 29,696.25 bytes per message, approximately 7.98% lower under this reference workload.

The 3.29 GB public ring and 189.5 MB certificate favor long-lived rings, persistent verifier caches, and many messages per certificate. A candidate workload is periodic pseudonymous monitoring by privately provisioned devices reporting to one persistent collector. At one report per minute under an unchanged certificate, 120,324 messages accumulate in about 84 days. A replacement certificate starts a new amortization period.

6.5. Lifecycle and Deployment Conditions

A certificate binds an exact canonical ring: membership changes require a new authorization, while old certificates refer to the old ring unless that context is rejected. There is no built-in expiration or cryptographic revocation. An authenticated external rejection list may invalidate identifiers or ring versions and remove cache records. Storage eviction requires certificate re-verification before accepting compact packets. Online-key compromise permits further signing under that certificate. Rejection of one identifier applies only to that authorization. Long-term compromise exposes the linking secret and permits future authorizations.

Global tags link a registered key across rings and time. If a side channel identifies one tag, its entire recorded history becomes attributable. Global linking supports cross-context duplicate detection. A scoped extension using $H_T(\langle x, \text{scope} \rangle)$ would require binding the scope in the statement, identifier, cache, and linking rule. MPC-friendly hashes, preprocessing-based MPC-in-the-head, Banquet, and VOLE-in-the-head provide alternative instantiation directions [22,23,37].

Analytical variants. A systematic $[I_d \mid A]$ key would use dk bits (1,026,467 bytes), subject to compatible syndrome-basis and trapdoor transformations and security analysis. An opening-only membership relation could omit the member-level PolarSig verification. Under the same layout, its 32-member proof would occupy 3,212,768 bytes. The member-level signature check establishes authorization under a registered verification key in addition to possession of the linking-secret opening.

7. Conclusions

We presented a linkable ring signature based on polar codes with cached anonymous authorization. A certificate binds a hidden member's authorization signature, a global tag, and an independent online key. Public linear-map evaluation before hidden selection reduces the authorization circuit's nonlinear cost, while certificate reuse removes the membership proof from subsequent online packets. For parameters selected against a 128-bit classical generic-attack target, the FS-ZKB++ instance provides a 98.52% proof-size reduction against the selected-matrix baseline for a 32-member ring and 807-byte cached online packets. The integrated prototype executes actual key generation, authorization,

installation, and online signing and verification. Under the stated primitive assumptions and honest registration, the construction supports repeated anonymous authentication with independently verifiable messages in the classical random oracle model.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math14213908/s1, Supplementary File S1: `S1_Integrated_Polar_LRS_Prototype.zip`. Source code, reproduction instructions, finite-length design parameters, generic-attack calculations, execution environment, raw measurements, and validation records for the complete authorization and online-authentication experiments.

Author Contributions: Conceptualization, Z.L.; methodology, Z.L. and J.L.; formal analysis, J.L.; validation, Q.L.; writing—original draft preparation, Z.L. and J.L.; writing—review and editing, Z.L., J.L. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (U25A20543).

Data Availability Statement: All analytical expressions and parameters supporting the results are included in the article. The source code, finite-length design, attack-estimation scripts, and raw measurements supporting the integrated experiments are provided in Supplementary File S1.

Acknowledgments: During preparation and revision of this manuscript, the authors used OpenAI Codex (OpenAI; version 26.1007.21434; accessed September–October 2026) to assist with literature searches, checking analytical calculations, prototype implementation and testing, benchmark processing, editing English, and LaTeX. The authors reviewed and revised the assisted content, remain responsible for independently checking the assisted claims, references, calculations, and code, and accept full responsibility for the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

LRS	Linkable ring signature
ROM	Random oracle model
MPC	Secure multiparty computation
FS	Fiat–Shamir
ZK	Zero knowledge
SC	Successive cancellation

Appendix A. Security Experiments

Initialize $\mathsf{Reg}, \mathsf{Cache}, \mathcal{A}, \mathcal{C}, \mathcal{Q}, \mathcal{I}, \mathcal{T}_{\text{issued}}$ as empty. $\mathcal{I}$ records identity-labeled authorization recipients. All games use Algorithm A1 and independent classical random oracles. VerifyTag uses a complete certificate or the validated cache.

ANON enforces candidate corruption and identity-labeled authorization/signing restrictions through eligibility and oracle checks before and after the challenge. $\mathcal{O}_*$ provides challenge messages. Its advantage is $|\Pr[b' = b] - 1/2|$. Other advantages are winning probabilities in Algorithms A2 and A3.

Algorithm A1 Common stateful game oracles

procedure $\mathcal{O}_{\mathrm{Reg}}$
  $i \leftarrow |\mathsf{Reg}|$; $(PK_i, SK_i) \leftarrow \mathsf{KeyGen}(1^\lambda; \theta)$
  $\mathsf{Reg}[i] \leftarrow PK_i$; retain SK_i
  return (i, PK_i)
end procedure
procedure $\mathcal{O}_{\mathrm{Auth}}(i, R)$
  Reject unless R is canonical and contains $\mathsf{Reg}[i]$
  $(\mathsf{Cert}, \mathsf{st}) \leftarrow \mathsf{Authorize}(SK_i, R; \theta)$
  Parse $\mathsf{Cert} = (T, E, \pi)$ and $\mathsf{st} = (D, esk)$
  Reject unless $\mathsf{Install}(R, \mathsf{Cert}, \mathsf{Cache}; \theta) = D$
  $\mathcal{A}[D] \leftarrow (i, R, \mathsf{Cert}, \mathsf{st})$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$; $\mathcal{T}_{\mathrm{issued}} \leftarrow \mathcal{T}_{\mathrm{issued}} \cup \{T\}$
  return Cert
end procedure
procedure $\mathcal{O}_{\mathrm{Sign}}(D, m)$
  Reject unless $D \in \mathrm{dom}(\mathcal{A})$
  $(i, R, \mathsf{Cert}, \mathsf{st}) \leftarrow \mathcal{A}[D]$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(R, m)\}$
  return $\mathsf{SignOnline}(\mathsf{st}, m; \theta)$
end procedure
procedure $\mathcal{O}_{\mathrm{Corrupt}}(i)$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$
  return SK_i and all retained online states belonging to i
end procedure

Algorithm A2 Unforgeability and linkability experiments

procedure UF($\mathcal{B}$)
  $(R, m, \Sigma) \leftarrow \mathcal{B}^{\mathcal{O}}(\theta, pp)$
  $T \leftarrow \mathsf{VerifyTag}(R, m, \Sigma; \theta)$
  return $\mathbf{1}[T \neq \perp \wedge (R, m) \notin \mathcal{Q} \wedge \mathrm{ids}(R) \cap \mathcal{C} = \varnothing]$
end procedure

procedure LINK($\mathcal{B}$)
  $(X_1, \ldots, X_N) \leftarrow \mathcal{B}^{\mathcal{O}}(\theta, pp)$
  $\mathcal{T} \leftarrow \varnothing$
  for $a = 1, \ldots, N$ **do**
    $T_a \leftarrow \mathsf{VerifyTag}(X_a; \theta)$
    if $T_a \neq \perp$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_a\}$
    end if
  end for
  return $\mathbf{1}[|\mathcal{T} \setminus \mathcal{T}_{\mathrm{issued}}| > |\mathcal{C}|]$
end procedure

$\mathcal{O}$ comprises the common and random oracles. Malformed or unregistered inputs are rejected.

https://doi.org/10.3390/math14213908

Algorithm A3 Anonymity and non-frameability experiments

procedure ANON($\mathcal{B}$)
  $(i_0, i_1, R, z) \leftarrow \mathcal{B}_1^{\mathcal{O}}(\theta, pp)$
  Reject unless $i_0 \neq i_1$, both are in R, and $\{i_0, i_1\} \cap (\mathcal{C} \cup \mathcal{I}) = \varnothing$
  $b \xleftarrow{\$} \{0,1\}$
  $(\mathsf{Cert}^*, \mathsf{st}^*) \leftarrow \mathsf{Authorize}(SK_{i_b}, R; \theta)$
  Define $\mathcal{O}_*(m) = \mathsf{SignOnline}(\mathsf{st}^*, m; \theta)$
  $b' \leftarrow \mathcal{B}_2^{\mathcal{O}, \mathcal{O}_*}(z, \mathsf{Cert}^*)$
  return $\mathbf{1}[b' = b]$
end procedure
procedure FRAME($\mathcal{B}$)
  $(i^*, z) \leftarrow \mathcal{B}_1^{\mathcal{O}}(\theta, pp)$
  Reject unless $i^* \notin \mathcal{C}$ is registered
  $T^* \leftarrow H_T(\langle x_{i^*} \rangle)$
  $(R, m, \Sigma) \leftarrow \mathcal{B}_2^{\mathcal{O}}(z, T^*)$
  $T \leftarrow \mathsf{VerifyTag}(R, m, \Sigma; \theta)$
  return $\mathbf{1}[T = T^* \neq \perp \wedge (R, m) \notin \mathcal{Q} \wedge i^* \notin \mathcal{C}]$
end procedure

References

1. *FIPS 204*; Module-Lattice-Based Digital Signature Standard. National Institute of Standards and Technology: Gaithersburg, MD, USA , 2024. [CrossRef]
2. *FIPS 205*; Stateless Hash-Based Digital Signature Standard. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024. [CrossRef]
3. Alagic, G.; Bros, M.; Ciadoux, P.; Dang, Q.; Dang, T.H.; Kelsey, J.; Lichtinger, J.; Liu, Y.-K.; Miller, C.; Moody, D.; et al. *Status Report on the Second Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process*; NIST IR 8610; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2026. [CrossRef]
4. Chang, S.-Y.; Khan, Q. Post-Quantum Cryptography in Networking Protocols: Challenges, Solutions, and Future Directions. *Cryptography* **2026**, *10*, 12. [CrossRef]
5. Rivest, R.L.; Shamir, A.; Tauman, Y. How to Leak a Secret. In *Advances in Cryptology—ASIACRYPT 2001*; LNCS 2248; Springer: Berlin/Heidelberg, Germany, 2001; pp. 552–565. [CrossRef] [PubMed]
6. Liu, J.K.; Wei, V.K.; Wong, D.S. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). In *Information Security and Privacy—ACISP 2004*; LNCS 3108; Springer: Berlin/Heidelberg, Germany, 2004; pp. 325–335. [CrossRef]
7. Liu, X.; Wang, L.-P. A Postquantum Linkable Ring Signature Scheme from Coding Theory. *Secur. Commun. Netw.* **2023**, *2023*, 1794053 . [CrossRef]
8. Goel, A.; Green, M.; Hall-Andersen, M.; Kaptchuk, G. Efficient Set Membership Proofs using MPC-in-the-Head. *Proc. Priv. Enhancing Technol.* **2022**, *2022*, 304–324. [CrossRef]
9. Xue, Y.; Lu, X.; Au, M.H.; Zhang, C. Efficient Linkable Ring Signatures: New Framework and Post-quantum Instantiations. In *Computer Security—ESORICS 2024*; LNCS 14985; Springer: Cham, Switzerland, 2024; pp. 435–456. [CrossRef]
10. Courtois, N.; Finiasz, M.; Sendrier, N. How to Achieve a McEliece-Based Digital Signature Scheme. In *Advances in Cryptology—ASIACRYPT 2001*; LNCS 2248; Springer: Berlin/Heidelberg, Germany, 2001; pp. 157–174. [CrossRef] [PubMed]
11. Debris-Alazard, T.; Sendrier, N.; Tillich, J.-P. Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes. In *Advances in Cryptology—ASIACRYPT 2019*; LNCS 11921; Springer: Cham, Switzerland, 2019; pp. 21–51. [CrossRef]
12. Sendrier, N. Wave Parameter Selection. In *Cryptology ePrint Archive*; Paper 2023/588; 2023. Available online: https://eprint.iacr.org/2023/588 (accessed on 19 October 2026) .
13. Feneuil, T.; Joux, A.; Rivain, M. Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs. In *Advances in Cryptology—CRYPTO 2022*; LNCS 13508; Springer: Cham, Switzerland, 2022; pp. 541–572. [CrossRef]
14. Bultel, X.; Olivier-Anclin, C. On the Anonymity of Linkable Ring Signatures. In *Cryptology and Network Security—CANS 2024*; Springer: Singapore, 2024; pp. 212–235. [CrossRef]
15. Tran, N.; Nguyen, K.; Liu, D.; Pieprzyk, J.; Susilo, W. Many-Time Linkable Ring Signatures. In *Provable and Practical Security—ProvSec 2025*; LNCS 16172; Springer: Singapore, 2025; pp. 3–22. [CrossRef]
16. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
17. Tal, I.; Vardy, A. List Decoding of Polar Codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 2213–2226. [CrossRef]
18. Forghani, P.; Koochak Shooshtari, M.; Aref, M.R. PolarSig: An Efficient Digital Signature Based on Polar Codes. *IET Commun.* **2020**, *14*, 2889–2897. [CrossRef]

19. Ishai, Y.; Kushilevitz, E.; Ostrovsky, R.; Sahai, A. Zero-Knowledge from Secure Multiparty Computation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*; ACM: New York, NY, USA, 2007; pp. 21–30. [CrossRef]
20. Giacomelli, I.; Madsen, J.; Orlandi, C. ZKBoo: Faster Zero-Knowledge for Boolean Circuits. In *25th USENIX Security Symposium*; USENIX Association: Berkeley, CA, USA, 2016; pp. 1069–1083. Available online: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/giacomelli (accessed on 19 October 2026).
21. Chase, M.; Derler, D.; Goldfeder, S.; Orlandi, C.; Ramacher, S.; Rechberger, C.; Slamanig, D.; Zaverucha, G. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In *ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2017; pp. 1825–1842. [CrossRef]
22. Baum, C.; Delpech de Saint Guilhem, C.; Kales, D.; Orsini, E.; Scholl, P.; Zaverucha, G. Banquet: Short and Fast Signatures from AES. In *Public-Key Cryptography—PKC 2021*; LNCS 12710; Springer: Cham, Switzerland, 2021; pp. 266–297. [CrossRef] [PubMed]
23. Baum, C.; Beullens, W.; Braun, L.; Delpech de Saint Guilhem, C.; Klooß, M.; Majenz, C.; Mukherjee, S.; Orsini, E.; Ramacher, S.; Rechberger, C.; et al. Shorter, Tighter, FAESTer: Optimizations and Improved (QROM) Analysis for VOLE-in-the-Head Signatures. In *Advances in Cryptology—CRYPTO 2025*; LNCS 16005; Springer: Cham, Switzerland, 2025; pp. 124–156. [CrossRef]
24. Derler, D.; Ramacher, S.; Slamanig, D. Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives. In *Post-Quantum Cryptography—PQCrypto 2018*; Springer: Cham, Switzerland, 2018; pp. 419–440. [CrossRef]
25. Fiat, A.; Shamir, A. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology—CRYPTO '86*; LNCS 263; Springer: Berlin/Heidelberg, Germany, 1987; pp. 186–194. [CrossRef] [PubMed]
26. Kulkarni, M.; Xagawa, K. Strong Existential Unforgeability and BUFF Securities of MPC-in-the-Head Signatures. In *Cryptology ePrint Archive*, Paper 2024/1069; 2024; revised 2025. Available online: https://eprint.iacr.org/2024/1069 (accessed on 19 October 2026).
27. Don, J.; Fehr, S.; Majenz, C.; Schaffner, C. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. In *Advances in Cryptology—CRYPTO 2019*; LNCS 11693; Springer: Cham, Switzerland, 2019; pp. 356–383. [CrossRef]
28. Don, J.; Fehr, S.; Majenz, C.; Schaffner, C. Efficient NIZKs and Signatures from Commit-and-Open Protocols in the QROM. In *Advances in Cryptology—CRYPTO 2022*; LNCS 13508; Springer: Cham, Switzerland, 2022; pp. 729–757. [CrossRef]
29. *FIPS 202*; SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015. [CrossRef]
30. Korada, S.B.; Urbanke, R. Polar Codes are Optimal for Lossy Source Coding. *IEEE Trans. Inf. Theory* **2010**, *56*, 1751–1768. [CrossRef]
31. Crypto-TII. CryptographicEstimators, Version 2.1.1. Available online: https://github.com/Crypto-TII/CryptographicEstimators/releases/tag/v2.1.1 (accessed on 12 October 2026).
32. May, A.; Sá Diogo, G. Multi-Instance Security Degradation of Code-Based KEMs. In *Cryptology ePrint Archive*; Paper 2026/517; 2026. Available online: https://eprint.iacr.org/2026/517 (accessed on 12 October 2026).
33. Sendrier, N. Decoding One Out of Many. In *Cryptology ePrint Archive*; Paper 2011/367; 2011. Available online: https://eprint.iacr.org/2011/367 (accessed on 19 October 2026).
34. Bardet, M.; Chaulet, J.; Dragoi, V.; Otmani, A.; Tillich, J.-P. Cryptanalysis of the McEliece Public Key Cryptosystem Based on Polar Codes. In *Post-Quantum Cryptography—PQCrypto 2016*; LNCS 9606; Springer: Cham, Switzerland, 2016; pp. 118–143. [CrossRef]
35. Attema, T.; Fehr, S.; Klooß, M.; Resch, N. The Fiat-Shamir Transformation of $(\Gamma_1, \ldots, \Gamma_\mu)$-Special-Sound Interactive Proofs. *J. Cryptol.* **2026**, *39*, 19. [CrossRef]
36. Xue, Y. Post-Quantum-Linkable-Ring-Signature: Author Implementation. Proof-Size Reporting in examples/src/main.rs. Available online: https://github.com/yuxi16/Post-Quantum-Linkable-Ring-Signature (accessed on 6 October 2026).
37. Katz, J.; Kolesnikov, V.; Wang, X. Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures. In *ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2018; pp. 525–537. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.