Scilight

Pragmatic Cybersecurity
https://www.sciltp.com/journals/pc

Article

Check for updates

Decentralized Biometric Authentication via Threshold-Based Key Derivation †

Haoze Cheng *, Hui Cui and James Boorman

Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia
* Correspondence: Haoze.Cheng@monash.edu
† Extended from: Decentralized Biometric Authentication via Threshold-Based Key Derivation. In Proceedings of the 31st Australasian Conference on Information Security and Privacy, Perth, Australia, 6–9 July 2026.

How To Cite: Cheng, H.; Cui, H.; Boorman, J. Decentralized Biometric Authentication via Threshold-Based Key Derivation. *Pragmatic Cybersecurity* **2026**, *1*(2), 11. https://doi.org/10.53941/pc.2026.100011

Received: 1 May 2026
Revised: 20 July 2026
Accepted: 4 August 2026
Published: 11 August 2026

Abstract: Biometric authentication offers enhanced usability for digital payments, but traditional centralized architectures suffer from single points of failure. While blockchain integration promises decentralized trust, existing solutions often store biometric helper data (e.g., fuzzy commitments) directly on-chain. We demonstrate that such transparency, even in permissioned settings, exposes low-entropy biometric inputs to offline brute-force attacks and identity-linkage risks if the immutable ledger is accessed by malicious nodes. To resolve this transparency–privacy paradox, we propose a threshold-based decentralized authentication framework. Unlike prior works, our protocol keeps all sensitive Biometric-Enhanced Key Derivation (BEKD) tokens entirely off-chain, using the blockchain solely for freshness enforcement. We provide a game-based security analysis of brute-force resistance, unforgeability, and unlinkability. Our experimental results demonstrate that our scheme's gas cost is acceptable, offering a robust solution for self-sovereign biometric identity.

Keywords: biometric authentication; blockchain; smart contracts; decentralization; brute-force attacks

1. Introduction

Biometric authentication has established itself as the mainstream standard for modern digital payments, offering superior usability and strong user binding compared to traditional password authentication [1]. However, its current deployment is predominantly centralized, where sensitive feature vectors are aggregated in monolithic databases controlled by a single authority [2]. This centralization introduces critical vulnerabilities, including single points of failure and massive data breaches [3]. Beyond breach risks, centralization also breaks biometric portability: enrollment and verification are effectively bound to a single access point, creating an availability bottleneck and a single point of failure, while real deployments require multi-access-point, cross-device authentication [4, 5]. Accordingly, decentralized biometric authentication is being pursued to decouple verification from any single server or access point and enable cross-device authentication after one enrollment, even among mutually distrustful parties [4, 5]. This paradigm shift exposes a fundamental conflict between ledger immutability and biometric irrevocability: while decentralization effectively eliminates trusted third parties [6], its immutable and redundant storage inherently conflicts with the stringent privacy requirements of biometric templates [3]. Furthermore, the transparency requirement crystallizes into a storage dilemma: the immutable persistence of biometric data turns the ledger into a permanent offline oracle, as the public state is fully replicable and enables unbounded offline trial verification without rate limiting for insider brute-force attacks [7, 8], thereby creating an irreversible vulnerability [9]. By contrast, moving storage entirely off-chain often forces a retreat to centralized servers, eroding the very decentralization the system seeks to achieve. As we detail next, this challenge has not been adequately accounted for in the current literature, leaving existing frameworks exposed to latent vulnerabilities.

1.1. Challenges

A critical gap remains in decentralized biometric authentication on blockchains: current designs often fail to reconcile immutability with biometric revocability while preventing the ledger from becoming an offline brute-force oracle [7, 10]. Although recent biometric authentication studies have proposed decentralized architectures, they often fail to address two interrelated vulnerabilities:

- **Insecurity of On-Chain Helper Data:** Existing frameworks [4, 11] mostly rely on storing helper data, such as fuzzy commitments or offsets, on-chain to reconcile noisy biometric inputs with exact cryptographic keys. This design choice can fundamentally weaken security in the blockchain setting. Because the helper data is derived from low-entropy biometric inputs, it fails to provide adequate protection against adversaries with access to the ledger [9, 12].
- **The Offline Brute-Force Vector:** The transparent nature of distributed ledgers enables an offline brute-force capability that traditional authentication protocols cannot mitigate. An adversary, whether an external hacker on a public chain or a malicious insider on a consortium chain, can download the ledger history to obtain the helper data for every user. Because the biometric input space is significantly smaller than the cryptographic key space, biometric-derived secrets have lower effective entropy [12]. The adversary can execute an offline dictionary attack by iteratively testing candidate features against the helper data [12]. Crucially, unlike centralized servers, the ledger enables unbounded offline trials, unconstrained by rate limiting or lockout mechanisms.

Consequently, merely obfuscating biometric-derived artifacts on-chain is insufficient. A robust solution must eliminate the permanent storage of any biometric-derived artifacts on the immutable ledger. This raises a fundamental research question: how can biometric key derivation be securely integrated into blockchain systems without reintroducing centralized trust assumptions or enabling offline guess-and-check attacks?

1.2. Our Contributions

In light of the vulnerability of helper data to offline brute-force and unlimited guess-and-check attacks, we propose a decentralized biometric authentication framework built upon the Biometric-Enhanced Key Derivation (BEKD) scheme [13]. Our contributions are summarized as follows:

- **Decentralized Wallet-Anchored BEKD Architecture:** We design a blockchain-native biometric authentication framework that binds each authentication session to a single-use BEKD token, significantly reducing the offline guessing attack surface while preserving key irreversibility.
- **On-Chain Freshness Enforcement without Biometric Exposure:** We enforce token freshness and single-use on-chain via a minimal authorization contract interface, preventing replay without revealing biometric-derived artifacts.
- **Removal of Centralized Trust Assumptions:** We replace the traditional single certification authority with a threshold-secured certificate authority (CA) consortium deployed in a blockchain setting, eliminating the off-chain single point of failure while maintaining distributed trust.
- **Game-based Security Analysis:** We formalize adversarial views for ledger, network, threshold-CA, and client-snapshot attackers, and provide game-based proof sketches showing that the blockchain does not become an offline biometric-verification oracle.
- **Performance Evaluation:** We adapt the original BEKD workflow into a blockchain-compatible decentralized protocol and implement an Ethereum prototype that instantiates the on-chain authorization interface, benchmarking end-to-end latency and on-chain gas overhead.

2. Related Work

Biometric authentication is widely deployed for access control, yet decentralized biometric authentication remains comparatively underexplored. Blockchain-based architectures can contribute to tamper evidence and distributed trust, which are attractive for protecting biometric-derived artifacts [5]. In transparent-ledger deployments, however, integration introduces a critical constraint: any publicly visible helper artifact can enable unbounded offline guess-and-check, effectively turning the ledger into a biometric-verification oracle [8]. Accordingly, we compare prior systems using brute-force resiliency (BR), which reflects resistance to unlimited offline verification in ledger settings. We evaluate BR by (i) freshness/single-use enforcement, (ii) absence of permanent on-chain biometric artifacts, and (iii) removal of centralized querying authority.

Abo Alzahab et al. [4] propose a decentralized biometric-authentication system that integrates a fuzzy-commitment scheme with blockchain, storing hash–offset pairs on-chain. Sharma et al. [5] develop a multimodal framework (face + dorsal hand) using an improved fuzzy vault and hybrid decentralized storage, reporting 99.8% accuracy. Zhang et al. [11] introduce BAKA, a lightweight authentication and key-agreement protocol for wireless

body area networks (WBANs) that leverages a fuzzy extractor to derive stable keys from biometrics without storing raw templates, combining passwords, elliptic-curve cryptography, and blockchain with formal security analysis. Extending this line, Zhang et al. [14] present BCAE, a blockchain-based cross-domain authentication and key-agreement scheme for edge settings that employs blockchain digital certificates, elliptic-curve cryptography, and a Delegated Proof of Stake (DPoS) consensus to provide scalable, lightweight authentication across heterogeneous domains. Table 1 summarizes these schemes and their brute-force resilience (BR).

Despite adopting decentralized components, many designs still inherit limitations from ledger transparency. Protocols that record commitment artifacts (e.g., hash–offset pairs or vault encodings) directly on-chain expose helper data to all participants, enabling adversaries to mount unbounded offline guess-and-check attacks against biometric-derived secrets [9]. As Ferrag et al. note [15], transparent ledgers can exacerbate identity-based and cryptanalytic threats when sensitive artifacts are permanently recorded. Moreover, most implementations lack explicit contract-level mechanisms to enforce brute-force resistance (e.g., freshness and single use), instead relying chiefly on biometric entropy and code parameters. Cryptographic constructions such as polynomial encoding and vault assembly further incur storage and computation overheads, raising gas efficiency and scalability concerns. Finally, modality-specific assumptions can limit portability across use cases, while broader blockchain threats, including replay, double-spending, Sybil, and denial-of-service, are often insufficiently addressed [8].

Table 1. Summary of related decentralized biometric authentication schemes and their brute-force resiliency (BR).

Author(s)	Bio-Auth	De-Auth.	BR
Abo Alzahab et al. [4]	Yes	Yes	Low
Sharma et al. [5]	Yes	Yes	Medium
Zhang et al. [11]	Yes	Yes	Medium
Zhang et al. [14]	No	Yes	Medium
Cui et al. [13]	Yes	No	High
Proposed scheme	Yes	Yes	High

3. Preliminaries

In this section, we provide an overview of the definitions used in this paper.

3.1. Brute-Force Attacks on Centralized Authentication

Brute-force is the natural failure mode of biometric authentication once verification reduces to offline guessing. In this regime, security no longer derives from protocol-level rate limiting or lockout, but from the effective entropy of the biometric representation and the cost of validating candidate reconstructions. The BrutePrint/InfinityGauntlet [12] line of attacks illustrates this transition: once online throttling is bypassed, smartphone fingerprint authentication on Android can be exhaustively searched in approximately 40 min. The same structural risk appears in server-assisted biometric key-reconstruction schemes. The server stores public helper data h (typically with a verifier) to enable recovery of a key K from a fresh sample. If the server or its storage is compromised, the adversary obtains h and can perform offline enumeration over biometric guesses until a consistent K is found. Exposing reusable biometric-derived artifacts therefore instantiates an offline oracle, shifting security from access control mechanisms to the typically constrained biometric guess space [12].

3.2. Biometric-Enhanced Key Derivation (BEKD)

The framework builds upon the Biometric-Enhanced Key Derivation (BEKD) scheme [13]. BEKD realizes brute-force resistance via a token-gated retrieval interface rather than publishing reusable helper data: enrollment produces a finite set of one-time tokens $\mathcal{T} = \{T_1, \ldots, T_m\}$, and recovery requires presenting an unused token together with a fresh liveness-verified biometric sample W'. The server maintains a spent list of token identifiers to prevent repeated trials, thereby bounding online guessing attempts by the available token budget [13]. In a decentralized construction, the off-chain certificate authority (CA) consortium instantiates the BEKD server role by enforcing one-time retrieval (e.g., via an L_T-style list) and rejecting repeated requests for the same identifier. This preserves BEKD's rate-limited brute-force resistance and token-level unlinkability, as each token is generated with independent randomness.

4. System & Threat Model

Our system consists of an off-chain wallet client, an off-chain CA consortium, and a minimal on-chain authorization layer. The wallet captures live biometrics and performs biometric feature processing, token construction,

and key reconstruction locally. During retrieval, the CA consortium performs only the threshold-dependent masking evaluation and tag comparison required by the BEKD recovery interface. All BEKD tokens and biometric-derived artifacts remain off-chain. The blockchain is used only to enforce token freshness (single-use) and to verify application-layer authorization signatures, so the ledger never becomes a reusable offline verification oracle. The CA role is decentralized as a consortium providing threshold attestation (rather than a single server), reducing single points of failure while retaining public verifiability on-chain [16].

Our threat model adopts the biometric template-protection perspective of ISO/IEC 24745 [17] and the adversarial framework of Simoens et al. [18], but specializes them to the blockchain-BEKD setting. In our setting, the main template-protection risks are not conventional template-database leakage, but whether ledger-visible state, network transcripts, CA shares, or stolen tokens can be used for reference recovery, identity tracing, or repeated authentication attempts. We therefore formulate these concerns as three security goals: brute-force resistance against offline biometric guessing, unforgeability of wallet authorization, and unlinkability across independently generated token epochs. In addition to these biometric threats, we account for blockchain-specific risks where globally observable on-chain state can amplify guess-and-check attacks if reusable helper data is published [4,8,15]. Accordingly, we adopt token-based BEKD to bound brute-force attempts by an online budget rather than unlimited offline trials [13].

4.1. System Overview

We present a decentralized biometric authentication scheme for blockchain transactions. The system comprises three roles: (i) a user-side wallet module that performs local biometric capture and BEKD computation; (ii) an off-chain CA consortium that issues and services authentication tokens under distributed trust; and (iii) a minimal on-chain core that enforces authorization and one-time token use. All raw biometric capture and feature stabilization are confined to the wallet, while threshold-dependent retrieval operations are performed by the off-chain CA consortium. No biometric sample, reusable biometric template, BEKD helper data, or token content is written on-chain. To eliminate single-point trust, the CA functionality is distributed across n authorities whose signing capability is protected by a (t, n) threshold scheme with quorum $t{+}1$ [16,19].

- **Wallet (client-side module).** The wallet captures a live biometric sample and executes the client-side BEKD procedures locally to derive/restore the signing key material. For each authorization, it submits an owner-bound authorization signature to the on-chain verifier and triggers an atomic freshness update by marking the token identifier as spent via $\mathsf{used}[\rho]$. The wallet communicates with the CA consortium only to obtain threshold-attested token material and to receive the recovery response required for wallet-side key restoration. No biometric samples or BEKD artifacts are written on-chain.
- **CA Consortium (off-chain issuer/service).** The CA consortium is a set of n off-chain authorities that collectively act as the token issuer and retrieval service. A quorum of at least $t{+}1$ authorities is required to produce attested token material under the consortium signing policy [16]. The consortium also enforces one-time retrieval at the service interface (e.g., via an off-chain used-token list such as L_T), ensuring that token reuse cannot be amplified into an offline oracle. The scheme does not rely on any single CA for correctness or availability.
- **On-chain core (verifier + freshness state).** The on-chain functionality is limited to public verifiability and freshness enforcement. A `ParamRegistry` publishes public parameters (e.g., consortium public key material and the threshold configuration t, n), while `SpentSet` maintains the freshness map $\mathsf{used}[\rho]$ for single-use enforcement. The contracts verify only application-level authorization signatures and freshness; they do not process biometric data or act as a biometric-verification oracle.

Interaction Workflow Overview

Figure 1 shows the scheme workflow for enrollment and wallet initialization. (1) The wallet captures a fresh biometric sample and locally derives BEKD enrollment material (masked shares $\{A_i\}$ and commitment R_0). (2) The wallet submits $(\{A_i\}, R_0)$ to an off-chain CA consortium operating under a (t, n) threshold policy. (3) A quorum of at least $t{+}1$ CAs returns a threshold attestation σ over the token-binding message. (4) The wallet assembles and stores the resulting token T in local storage; no biometric data, helper material, or token contents are written on-chain. (5) The wallet registers the corresponding public verification information with the on-chain core; thereafter, each authentication uses (T, σ) (and, when needed, a CA-assisted retrieval response) to produce an application-level authorization signature, while the contract enforces single-use via $\mathsf{used}[\rho]$.

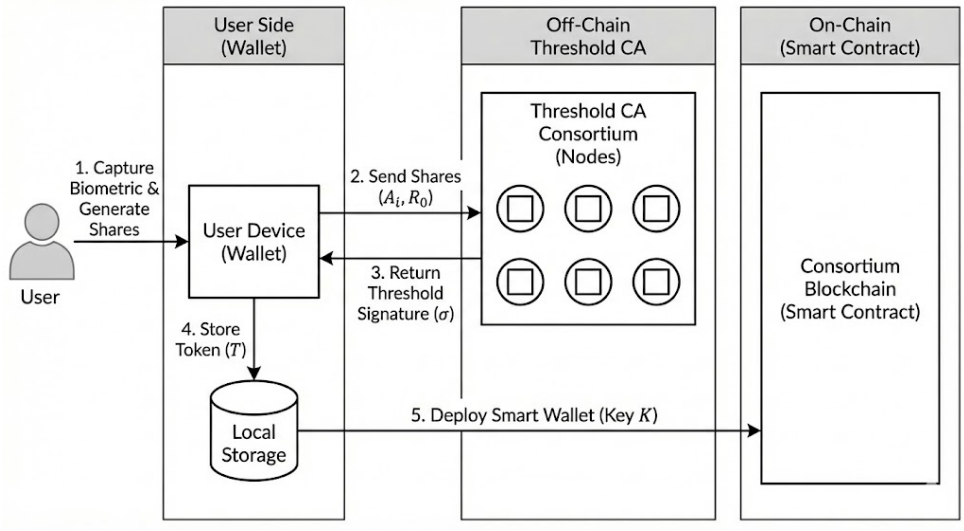

Figure 1. Decentralized BEKD authentication system overview.

4.2. Adversary Model and Assumptions

The following adversary classes specify which system components can be used to implement the objectives above.

4.2.1. Adversary Classification and Capabilities

Following the adversary classification of Simoens et al. [18] and the biometric template protection framework of ISO/IEC 24745 [17], we define four adversary classes distinguished by their access to system components. Each class is characterized by a formal view—the set of protocol artifacts observable by the adversary—which serves as the basis for the game-based security analysis in Section 5.2.

- **Ledger adversary** has unrestricted read access to the blockchain and can invoke any public contract function. *View*: $V_{\text{ledger}} = \{\rho,\ \mathsf{addr}(K),\ pk_{CA},\ \mathsf{used}[\rho],\ \text{public contract state}\}$, where $\mathsf{addr}(K)$ denotes the on-chain address derived from the public key $K = g^k$.
- **Network adversary** controls the transport between the client and the CA consortium, and can intercept, drop, delay, and replay messages. We assume the client–CA channel provides authenticated confidentiality (e.g., Transport Layer Security (TLS)), so the adversary observes only public metadata and ciphertext transcripts; any injected or modified ciphertexts will be rejected by channel authentication, and the adversary cannot learn plaintext biometric-derived scalars.
  View: $V_{\text{net}} = V_{\text{ledger}} \cup \{\mathsf{meta}, \mathsf{ct}\}$, where ct denotes the encrypted transcripts.
- **Threshold-CA adversary** statically corrupts up to t out of n consortium nodes during setup, learning their full internal state, including their key shares, local randomness, protocol state, and all messages received or generated by the corrupted nodes during enrollment and retrieval.
  View: $V_{\mathsf{tCA}} = V_{\mathsf{ledger}} \cup \left\{ sk_{CA}^{(i)}, \mathsf{state}_i, \mathsf{transcript}_i \right\}_{i \in S,\ |S| \le t}$.
- **Client snapshot adversary** performs a one-time compromise of a user device, exfiltrating all at-rest off-chain state (tokens and local logs). The adversary gains no access to live sensor inputs or runtime execution within the trusted execution environment (TEE). *View*: $V_{\mathsf{snap}} = V_{\mathsf{ledger}} \cup \{T\}$, where the complete token is $T = (T_U, T_{CA})$ with $T_U = (c, \rho)$ and $T_{CA} = (R_0, R_1, h_A, \sigma, \{A_i\}_{i\in[d]}, \{\mathsf{tag}_i\}_{i\in[d]})$ (Table 2).

These adversary classes are composable: a multi-capability adversary is modeled by taking the union of the corresponding capabilities and observable views. For example, the strongest non-live adversary used in Definition 2 combines ledger access, optional network control, client snapshot access, and corruption of up to t CA nodes, but still excludes access to a fresh biometric sample. The brute-force experiment further parameterizes this composed adversary by the number q_{CA} of online retrieval attempts allowed through the CA interface. In all classes, the adversary is computationally bounded and modeled as a probabilistic polynomial-time (PPT) algorithm.

4.2.2. Security Goals

We define three game-based security goals and two mechanism-level protection properties aligned with ISO/IEC 24745 [17], the adversarial framework of Simoens et al. [18], and the BEKD security model [13].

- **Brute-force resistance** (Definition 1): This goal captures reference recovery in the sense of Simoens et

al. [18] and irreversibility in the sense of ISO/IEC 24745 [17]: the adversary should not be able to recover the biometric-bound signing scalar or validate biometric guesses from on-chain data, protocol transcripts, stolen tokens, or sub-threshold CA shares. In the snapshot-only setting with an honest CA consortium, success is bounded by the online retrieval budget q_{CA} and biometric entropy, reflecting the rate-limited recovery interface of BEKD [13].

- **Unforgeability** (Definition 2): This goal captures authorization integrity and secure binding. In ISO/IEC 24745 terminology, the protected biometric reference should remain bound to the intended identity and application context [17]; in our setting, this means that an adversary who possesses stored tokens and corrupts up to t CA key shares, but lacks a live biometric sample and a successful retrieval, cannot produce a valid application-layer signature σ_U accepted by the smart contract.
- **Cross-epoch unlinkability** (Definition 3): This goal captures identity tracing in the adversarial framework of Simoens et al. [18] and unlinkability in ISO/IEC 24745 [17]: ledger, network, and threshold-CA adversaries should not determine whether two independently generated enrollment sessions originate from the same user with probability non-negligibly better than $1/2$.

Table 2. Key protocol notation.

Symbol	Description
Off-chain (CA + Token)	
sk_{CA}	CA master secret (Shamir-shared; never reconstructed).
$pk_{CA} = g^{sk_{CA}}$	CA master public key.
(t, n)	Threshold: $t{+}1$ shares required.
$sk_{CA}^{(i)}$	CA share at node i.
(R_0, R_1)	Envelope $(g^r,\ pk_{CA}^r \cdot g^k)$.
$M = pk_{CA}^r$	Masking base, computed via $(t{+}1, n)$-threshold evaluation.
W_i, W_i'	Stabilized biometric feature components from enrollment and retrieval captures.
w_i, w_i'	Feature-derived scalars: $w_i = H_0(W_i, c)$ and $w_i' = H_0(W_i', c)$.
Z_i	Mask $H_1(M, M^{w_i})$.
A_i	Masked share $f(i){+}Z_i \pmod q$, with $f(0) = k$.
tag_i	Tag $\mathsf{tag}_i = H_{\mathsf{tag}}(i \parallel \rho \parallel Z_i)$.
h_A	Commitment over $\{A_i\}, \{\mathsf{tag}_i\}$.
T_U, T_{CA}	$T_U = (c, \rho);\ T_{CA} = (R_0, R_1, h_A, \sigma, \{A_i\}, \{\mathsf{tag}_i\})$.
On-chain (freshness)	
ρ	Token identifier $\mathsf{Keccak256}(R_0)$.
$\mathsf{used}[\rho]$	Spent-set bit $(0 \to 1)$.
Authentication	
σ	CA threshold signature on $m = H_2(R_0, R_1, h_A)$.
K_{dec}	Derived public key $(= g^k)$.
$\mathsf{pk}_{\mathsf{owner}}$	Registered owner verification key/address bound to C_{auth}.
opHash	Application operation digest (request digest).
$\mathsf{authHash}$	Authorization digest $H_{\mathsf{auth}}(\rho \parallel \mathsf{opHash})$.
σ_U	User authorization signature on $\mathsf{authHash}$.
d, t_{bio}	Number of feature components and minimum matched components required for recovery.

Since our model instantiates ISO/IEC 24745 [17] and the Simoens et al. [18] framework in the blockchain-BEKD setting, freshness and renewability are included as mechanism-level protection properties rather than separate cryptographic games. Freshness is enforced by consuming token identifiers during CA-side retrieval and on-chain authorization, preventing a token from becoming a reusable online verification oracle. Renewability is provided by issuing a fresh token epoch with independent randomness (r, k, c, ρ) after a token is spent or revoked; by the unlinkability goal above, the new epoch is computationally unlinkable to prior epochs for the admissible adversaries in our model.

Accordingly, the threat objectives identified above are instantiated in the subsequent game-based security analysis as follows. Reference recovery and irreversibility are captured by the brute-force resistance experiment in Definition 1, which considers whether an adversary can validate biometric guesses or recover the biometric-bound secret from the admissible system views. Authorization integrity and secure binding are captured by the unforgeability experiment in Definition 2, which considers whether a valid on-chain authorization can be produced without a successful honest retrieval. Identity tracing and unlinkability are captured by the cross-epoch unlinkability

experiment in Definition 3, which measures an adversary's ability to link independently generated enrollment epochs. Freshness and renewability are addressed at the mechanism level through one-time token consumption and the issuance of newly randomized token epochs, respectively. The corresponding assumptions, security arguments, and proof sketches are presented in Section 5.2 and Appendix A.

5. Detailed Construction

We provide the detailed design of the decentralized biometric authentication framework in this section. Table 2 presents the notation employed throughout.

5.1. Decentralized Biometric Authentication Construction

Below, we describe the decentralized biometric authentication mechanism built upon the BEKD scheme in [13]. A biometric capture is represented as $W = (W_1, \ldots, W_d)$ after liveness checking and feature stabilization. Recovery requires at least t_{bio} reproduced components across enrollment and retrieval; tolerance to biometric variation is therefore captured by the stabilized feature representation and the t_{bio}-out-of-d recovery rule, while H_0 only maps each reproduced component to the scalar used by the BEKD masking procedure.

5.1.1. Setup

The CA consortium consists of n independent nodes operating under a (t, n)-threshold scheme, requiring cooperation from $t+1$ nodes for cryptographic operations [16]. To ensure Ethereum compatibility, we instantiate the `secp256k1` curve, i.e., a cyclic group $\mathbb{G}$ of prime order q with generator g [20]. We assume the standard hardness of the discrete logarithm (DLP) and computational Diffie–Hellman (CDH) problems in $\mathbb{G}$, and additionally rely on decisional Diffie–Hellman (DDH) when required for indistinguishability (e.g., unlinkability). A Distributed Key Generation (DKG) protocol [16,19] produces the master public key $pk_{CA} = g^{sk_{CA}}$, while private shares $sk_{CA}^{(i)}$ are held individually by nodes. The master secret sk_{CA} is never reconstructed; a valid consortium operation requires $t+1$ participants, ensuring security against any coalition of at most t nodes [16]. We employ domain-separated Keccak-256 [21] modeled as independent random oracles:

$$
\begin{aligned}
H_0 &: \mathcal{W} \times \{0,1\}^{256} \to \mathbb{Z}_q, & w_i &= H_0(W_i, c),\\
H_1 &: \mathbb{G}^2 \to \mathbb{Z}_q, & Z_i &= H_1(M, M^{w_i}),\\
H_2 &: \mathbb{G}^2 \times \mathbb{Z}_q \to \mathbb{Z}_q, & m &= H_2(R_0, R_1, h_A),\\
H_3 &: \{0,1\}^* \to \mathbb{Z}_q, & h_A &= H_3(A_1 \| \cdots \| A_d \| \mathrm{tag}_1 \| \cdots \| \mathrm{tag}_d),\\
H_{\mathsf{tag}} &: \{0,1\}^* \to \{0,1\}^{\lambda},\\
H_{\mathsf{auth}} &: \{0,1\}^* \to \{0,1\}^{256}.
\end{aligned}
$$

To minimize on-chain footprint, we deploy three contracts: `ParamRegistry` (public parameters including pk_{CA}), `SpentSet` (freshness mapping $\mathsf{used}[\rho]$), and an authorization contract C_{auth} for owner-bound signature validation. Tokens are identified by $\rho = \mathsf{Keccak256}(R_0)$. For evaluation, we instantiate a $(1, 3)$ threshold CA consortium, tolerating up to one malicious CA node. We model the on-chain component as a minimal authorization contract C_{auth} that (i) maintains a freshness map $\mathsf{used}[\rho] \in \{0, 1\}$ and (ii) verifies an owner-bound authorization signature over an operation digest. Let opHash denote an application operation digest, and define the authorization digest as

$$\mathsf{authHash} \leftarrow H_{\mathsf{auth}}(\rho \,\|\, \mathsf{opHash}),$$

where H_{auth} is a domain-separated Keccak-256 tag for authorization.

5.1.2. Enrollment

Enrollment corresponds to the BEKD Issue phase and is executed entirely off-chain between the user's wallet and the CA consortium, as detailed in Algorithm 1. A liveness-verified biometric capture is processed locally into the stabilized feature vector W and remains confined to the device throughout. The wallet generates a fresh secret k that determines the session key $K = g^k$. To bind this key to the user's biometric traits, the wallet derives feature-dependent scalars from W using a per-token blinding factor c. These values are used to construct masked Shamir shares of k, ensuring that recovery of the key later requires both sufficient biometric agreement and CA cooperation. The masking follows the BEKD paradigm: each Shamir share is protected using a value derived from an ElGamal-style envelope under the CA consortium's public key. This construction ensures that the masking base depends on the CA secret and per-token randomness, preventing offline recovery of masking values

from the token alone. To enable stateless verification during the retrieval phase, the wallet additionally generates per-feature verification tags and commits both masked shares and tags to the hash h_A. The tuple (R_0, R_1, h_A) is then authenticated by the CA consortium via the threshold Elliptic Curve Digital Signature Algorithm (ECDSA) [16,19], producing a signature σ without reconstructing the master secret key. Upon successful verification, the wallet forms the token $T = (T_U, T_{CA})$, where T_U contains the blinding factor and identifier, and T_{CA} contains the envelope, commitment, signature, and masked shares. No raw biometric samples, derived scalars, or secret keys are stored. Fresh randomness in each enrollment ensures unlinkability across tokens.

5.1.3. Retrieve (Off-Chain Key Recovery)

Algorithm 2 specifies the off-chain key-recovery procedure. When a user initiates an action requiring authorization, the wallet selects an unused token $T = (T_U, T_{CA})$ from secure storage and captures a fresh, liveness-verified biometric sample W'. Key recovery is performed entirely off-chain and requires cooperation from the CA consortium. The wallet first derives fresh biometric scalars from W' using the per-token blinding factor stored in T_U. It then transmits the signed envelope (R_0, R_1, h_A, σ) together with the derived scalars and verification tags to the CA consortium over a secure channel. A quorum of at least $t{+}1$ CA nodes validates the threshold signature under pk_{CA}. Upon successful verification, the consortium reconstructs the internal masking base corresponding to the enrollment randomness and derives a candidate session key from the ElGamal-style envelope. For each feature index, the consortium recomputes a masking value using the received biometric scalars and checks consistency against the stored verification tags. Indices that satisfy this check form a valid set S. If fewer than t_{bio} indices are valid, recovery fails. The CA consortium returns the valid index set together with the derived session key commitment to the wallet. The wallet then removes the masking values from the corresponding shares and interpolates the Shamir polynomial to recover the signing scalar k. A final consistency check ensures that the reconstructed key matches the session key derived by the consortium. If this verification succeeds, k is retained in volatile memory for immediate use in the authentication phase; otherwise, the procedure aborts without producing any on-chain effect. Throughout this process, no raw biometric samples are transmitted, and the masking base remains internal to the CA consortium. Consequently, successful recovery requires both biometric consistency and valid CA attestation. The retrieval-side spent list L_T is maintained as a consortium-wide service state rather than as an independent local state at each CA node. A token identifier ρ is accepted only once at the retrieval interface: after a quorum validates the token attestation and commits a replicated record $\rho \in L_T$, subsequent retrieval requests using the same identifier are rejected. This rule bounds biometric guessing attempts at the CA interface and prevents different CA subsets from repeatedly servicing the same token.

We model L_T as a quorum-replicated, linearizable set managed by the CA consortium. Each retrieval request invokes an atomic $\mathsf{CheckAndInsert}(\rho)$ operation across the consortium. The request is rejected if ρ is already in the set. Otherwise, the CA nodes must reach agreement and commit ρ to the set before performing the threshold-dependent retrieval computations. If the nodes fail to reach agreement, the retrieval request is aborted.

5.1.4. Authenticate (Application-Layer Authorization)

Algorithm 3 specifies the application-layer authorization procedure. After a successful retrieval, the client authorizes the pending operation at the application layer by signing a domain-separated authorization digest. Concretely, it binds the token identifier $\rho = \mathsf{Keccak256}(R_0)$ and an operation digest opHash into $\mathsf{authHash} \leftarrow H_{\mathsf{auth}}(\rho \,\|\, \mathsf{opHash})$. This binding prevents replay across execution contexts and ties each authorization to a specific token instance. The client signs $\mathsf{authHash}$ using the recovered ECDSA signing scalar k to obtain σ_U, and then immediately erases k from volatile memory to minimize post-signing exposure. It submits $(\rho, \mathsf{opHash}, \sigma_U)$ to the on-chain authorization contract C_{auth}. Notably, the transaction does not include any biometric material, the BEKD envelope $(R_0, R_1, \cdot)$, masked shares, or the CA attestation σ; these artifacts remain off-chain and are only involved in the retrieval stage.

On-chain, C_{auth} performs two checks atomically. First, it queries `SpentSet` to ensure $\texttt{used}[\rho] = 0$, enforcing one-time freshness and preventing replay of the same token identifier.

Second, it recomputes $\mathsf{authHash}$ and verifies σ_U under the registered owner verification key $\mathsf{pk}_{\mathsf{owner}}$ by checking $\mathsf{Verify}(\mathsf{pk}_{\mathsf{owner}}, \mathsf{authHash}, \sigma_U) = 1$, ensuring that the authorization originates from the key bound to C_{auth}. If any check fails, the transaction reverts with no state change; otherwise, $\texttt{used}[\rho]$ is set to 1 to permanently burn the token identifier, and the requested operation is executed. This design cleanly separates biometric-dependent processing (off-chain) from freshness enforcement and authorization (on-chain), providing strong replay resistance without exposing biometric-derived artifacts on the ledger. This separates freshness into two stages: the off-chain list L_T governs retrieval, while the on-chain map $\mathsf{used}[\rho]$ governs execution. The former limits key-recovery attempts

before a signing key is reconstructed, while the latter prevents replay of an already authorized blockchain operation. If retrieval succeeds but the later on-chain transaction fails, the identifier remains consumed in L_T; this fail-closed behavior preserves the BEKD-style bound on repeated biometric trials. The two freshness mechanisms are sequential rather than atomically synchronized across layers. Writing $\ell = 1$ when ρ is recorded in L_T and $u = 1$ when $\mathsf{used}[\rho]$ is set on-chain, the admissible state transitions are $(0,0) \to (1,0) \to (1,1)$. The intermediate state $(1,0)$ occurs when the retrieval attempt has been consumed but authorization has not yet completed. If the blockchain transaction subsequently fails, the token remains consumed and a new token must be issued. This fail-closed behavior prevents repeated biometric trials. The state $(0,1)$ is unreachable in an honest execution because an accepted on-chain authorization requires possession of the recovered signing key.

Algorithm 1 Enrollment Protocol (BEKD-Compatible)

Require: Stabilized biometric feature vector $W = (W_1, \ldots, W_d)$ from a liveness-verified capture
Ensure: Token $T = (T_U, T_{\mathsf{CA}})$ stored off-chain

Fetch pk_{CA} and parameters $(d, t_{\mathsf{bio}}, t, n)$ from $\mathsf{ParamRegistry}$
Sample $k \xleftarrow{\$} \mathbb{Z}_q$, set $K \leftarrow g^k$
Sample $c \xleftarrow{\$} \{0,1\}^{256}$
For $i \in [d]$: $w_i \leftarrow H_0(W_i, c)$
Sample $r \xleftarrow{\$} \mathbb{Z}_q$; set $R_0 \leftarrow g^r$, $R_1 \leftarrow pk_{CA}^r \cdot K$, $M \leftarrow pk_{CA}^r$, $\rho \leftarrow \mathsf{Keccak256}(R_0)$
Choose Shamir polynomial f of degree $(t_{\mathsf{bio}}-1)$ with $f(0) = k$
for $i \in [d]$ **do**
    $Z_i \leftarrow H_1(M, M^{w_i})$
    $A_i \leftarrow (f(i) + Z_i) \bmod q$
    $\mathrm{tag}_i \leftarrow H_{\mathrm{tag}}(i \parallel \rho \parallel Z_i)$
end for
$h_A \leftarrow H_3(A_1 \parallel \cdots \parallel A_d \parallel \mathrm{tag}_1 \parallel \cdots \parallel \mathrm{tag}_d)$
$m \leftarrow H_2(R_0, R_1, h_A)$
$\sigma \leftarrow \mathsf{ThresECDSA.Sign}(m)$
if $\mathsf{Verify}(pk_{CA}, m, \sigma) = 0$ then abort
$T_U \leftarrow (c, \rho)$
$T_{\mathsf{CA}} \leftarrow (R_0, R_1, h_A, \sigma, \{A_i\}_{i\in[d]}, \{\mathrm{tag}_i\}_{i\in[d]})$
Securely erase $k, \{w_i\}$; store T off-chain

Algorithm 2 Retrieve Protocol (Off-Chain Key Recovery)

Require: Fresh stabilized biometric feature vector $W' = (W'_1, \ldots, W'_d)$, token $T = (T_U, T_{CA})$
Ensure: Signing scalar $k \in \mathbb{Z}_q$ or $\perp$

Wallet $\to$ **CA:** parse $(c, \rho) \leftarrow T_U$, $(R_0, R_1, h_A, \sigma, \{A_i\}, \{\mathsf{tag}_i\}) \leftarrow T_{CA}$; compute $w'_i \leftarrow H_0(W'_i, c)\ \forall i \in [d]$; send $(R_0, R_1, h_A, \sigma, \rho, \{w'_i\}, \{\mathsf{tag}_i\})$.
CA consortium: $m \leftarrow H_2(R_0, R_1, h_A)$;
if $\mathsf{Verify}(pk_{CA}, m, \sigma) = 0 \vee \rho \in L_T$ **then**
    output $\perp$
end if
$L_T \leftarrow L_T \cup \{\rho\}$; select $J \subseteq [n]$, $|J| = t{+}1$, compute $M \leftarrow \prod_{j\in J} (R_0^{sk_{CA}^{(j)}})^{\lambda_j}$ (Lagrange at 0); $K_{\mathsf{dec}} \leftarrow R_1 / M$.
$S \leftarrow \{(i, Z'_i) \mid i \in [d],\ Z'_i \leftarrow H_1(M, M^{w'_i}),\ H_{\mathsf{tag}}(i \parallel \rho \parallel Z'_i) = \mathsf{tag}_i\}$;
if $|S| < t_{\mathsf{bio}}$ **then**
    output $\perp$
end if
send (S, K_{dec}).
Wallet: choose $S^\star \subseteq S$, $|S^\star| = t_{\mathsf{bio}}$; recover $k \leftarrow f(0)$ from $\{(i, (A_i - Z'_i) \bmod q)\}_{(i,Z'_i)\in S^\star}$;
if $g^k \neq K_{\mathsf{dec}}$ **then**
    output $\perp$
end if
output k.

Algorithm 3 Authenticate Protocol (Application-Layer Authorization)

Require: Signing scalar k, token $T = (T_U, T_{CA})$, operation digest opHash
Ensure: Executed on-chain operation or revert

Client (Off-Chain Signing):
Extract R_0 from T_{CA} and compute $\rho \leftarrow \mathsf{Keccak256}(R_0)$
$\mathsf{authHash} \leftarrow H_{\mathsf{auth}}(\rho \parallel \mathsf{opHash})$
Compute $\sigma_U \leftarrow \mathsf{ECDSA.Sign}(k, \mathsf{authHash})$
Securely erase k
Submit $(\rho, \mathsf{opHash}, \sigma_U)$ to C_{auth}
Authorization Contract (On-Chain Verification):
require $\mathsf{used}[\rho] = 0$
$\mathsf{authHash} \leftarrow H_{\mathsf{auth}}(\rho \parallel \mathsf{opHash})$
require $\mathsf{Verify}(pk_{\mathsf{owner}}, \mathsf{authHash}, \sigma_U) = 1$
$\mathsf{used}[\rho] \leftarrow 1$
Execute authorized operation

5.2. Security Analysis

We provide game-based security arguments for the decentralized BEKD construction under the adversary model of Section 4.2. The objective is not to introduce a new BEKD primitive, but to show that decentralizing the BEKD server role and restricting on-chain state to freshness checking do not reintroduce the offline verification oracle that BEKD is designed to avoid. The games below instantiate the biometric-template protection objectives discussed in Section 4.2.2: brute-force resistance captures reference recovery and irreversibility, unforgeability captures authorization integrity and secure binding, and cross-epoch unlinkability captures identity tracing. Residual guessing attacks are necessarily online and are bounded by a budget at the CA retrieval interface, following the token-gated recovery model of BEKD [13] and the budgeted, rate-limited treatment of brute-force-resistant biometrics [12].

Our security analysis treats BEKD as the underlying token-gated key-recovery primitive and focuses on whether the decentralized deployment preserves its security boundary. The hiding, binding, anonymity, and token-unforgeability properties of the BEKD token mechanism follow the original BEKD model [13]; we do not re-prove the primitive itself. Building on the BEKD recovery interface, our security analysis asks whether decentralizing the server role and exposing only freshness state on-chain introduces any new offline verification surface. The games below show that it does not: sub-threshold CA corruption does not reveal the masking base, ledger-visible state contains no biometric-derived matching artifact, and replay is handled by the on-chain freshness layer. Consequently, the construction preserves BEKD's token-gated brute-force boundary while removing the single-server trust assumption and making the mechanism suitable for decentralized authentication.

Each experiment below instantiates the adversary capability classes from Section 4.2 by explicitly specifying the adversary's view, permitted queries, challenge condition, and winning event.

5.2.1. Assumptions

We assume: (i) DLP and CDH hardness in $\mathbb{G}$ (secp256k1); (ii) domain-separated Keccak-256 in the random-oracle (RO) model, with fixed distinct tags $\mathsf{DS}_b \in \{\mathtt{H0}, \mathtt{H1}, \mathtt{H2}, \mathtt{H3}, \mathtt{HTAG}, \mathtt{HAUTH}\}$ and $H_b(x) = \mathsf{Keccak256}(\mathsf{DS}_b \parallel \mathsf{enc}(x))$; (iii) existential unforgeability under chosen-message attacks (EUF-CMA) of ECDSA and security of $(t{+}1, n)$-threshold ECDSA against up to t corruptions; (iv) liveness-verified biometric capture (TEE/sensor assumption) as in [13]. We use the capability classes from Section 4.2; composed adversaries are obtained by taking the union of the corresponding capabilities and views.

5.2.2. Token Binding

In Algorithm 1, the CA consortium signs

$$m = H_2(R_0, R_1, H_3(A_1 \| \cdots \| A_d \| \mathsf{tag}_1 \| \cdots \| \mathsf{tag}_d)) .$$

Any change to $(R_0, R_1, \{A_i\}, \{\mathsf{tag}_i\})$ changes m and invalidates σ, absent forging a threshold signature.

Definition 1 (Brute-Force Experiment)**.** $\mathsf{Exp}^{\mathsf{bf}}_{\mathcal{A},\Pi}(\lambda)$*: the challenger runs setup and one enrollment epoch to obtain* $T = (T_U, T_{CA})$ *and* $K = g^k$*. The adversary is instantiated as a composition of the capability classes in Section 4.2; in particular, it may obtain ledger and network views, optionally obtain a client snapshot, and optionally corrupt up*

to t CA nodes. The adversary may perform arbitrary offline computation and make at most q_{CA} online calls to the CA retrieval interface. It wins if it outputs a biometric guess W^* or scalar k^* such that the protocol accepts the guess and $g^{k^*} = K$. Define $\mathsf{Adv}^{\mathsf{bf}}(\mathcal{A}) = \Pr[\mathcal{A}\ wins]$.

Theorem 1 (Budgeted brute-force resistance)**.** *Assume CDH/DLP hardness, the RO model, and (t, n)-threshold (requiring $t+1$ parties) security. Let each feature component have min-entropy H_∞ under the standard limited-correlation model used in biometric brute-force analyses. Then:*

1. *(*No token snapshot*) If $\mathcal{A}$ does not obtain $T_U = (c, \rho)$, then $\mathsf{Adv}^{\mathsf{bf}}(\mathcal{A})$ is negligible.*
2. *(*Token snapshot, honest CA*) If $\mathcal{A}$ obtains T but fewer than $t+1$ CA nodes are corrupted, then $\mathsf{Adv}^{\mathsf{bf}}(\mathcal{A}) \le q_{\mathsf{CA}} \cdot \sum_{j=t_{\mathsf{bio}}}^{d} \binom{d}{j} 2^{-jH_\infty} + \mathsf{negl}(\lambda)$.*
3. *(*Token snapshot and CA threshold break*) If $\mathcal{A}$ obtains T and corrupts at least $t+1$ CA nodes, offline evaluation of $M = pk_{CA}^{r}$ becomes possible, and the bound degrades to the entropy-only term $\sum_{j=t_{\mathsf{bio}}}^{d} \binom{d}{j} 2^{-jH_\infty}$.*

Definition 2 (Forgery Experiment)**.** *In $\mathsf{Exp}^{\mathsf{forge}}_{\mathcal{A},\Pi}(\lambda)$, $\mathcal{A}$ receives the strongest non-live composition of capabilities from Section 4.2: ledger access, optional network control, client snapshot access, and corruption of up to t CA nodes. The adversary has no access to a fresh biometric sample or liveness oracle. It wins if it outputs an operation and signature accepted by the on-chain authorization contract under $\mathsf{addr}(K)$ without an honest successful retrieval.*

Theorem 2. *Assuming EUF-CMA security of ECDSA, security of the $(t+1, n)$-threshold ECDSA scheme against corruption of at most t consortium nodes, and the liveness-capture assumption, the advantage of any PPT adversary $\mathcal{A}$ in $\mathsf{Exp}^{\mathsf{forge}}$ satisfies*

$$\mathsf{Adv}^{\mathsf{forge}}_{\Pi}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{euf}}_{\mathsf{ECDSA}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{euf}}_{\mathsf{ThresECDSA}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{bf}}_{\Pi}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

Consequently, the forgery advantage is negligible whenever the budgeted brute-force advantage established in Theorem 1 is negligible under the selected biometric entropy, matching threshold, and online retrieval-query budget.

Definition 3 (Unlinkability Experiment)**.** *The experiment $\mathsf{Exp}^{\mathsf{unlink}}_{\mathcal{A},\Pi}(\lambda)$ generates two independently randomized enrollment epochs using fresh values (r, k, c) and provides $\mathcal{A}$ with the ledger view, the cryptographic network transcripts, and the threshold-CA views permitted under the adversary model in Section 4.2. The network view excludes stable transport-layer identifiers that would directly reveal the user's identity independently of the protocol execution. The adversary wins if it determines whether the two epochs were generated for the same user with probability non-negligibly greater than $1/2$. Client snapshot access is excluded because a snapshot exposes a concrete locally stored token instance and therefore falls outside the intended cross-epoch identity-tracing experiment.*

The unlinkability notion considered here is restricted to the cryptographic artifacts generated by the protocol, including ledger-visible state, encrypted protocol transcripts, and the internal views of sub-threshold corrupted CA nodes. Transport-layer side channels, such as persistent IP addresses, stable endpoint identifiers, timing correlation, or device fingerprinting, are outside the scope of this experiment unless the deployment additionally employs an anonymous or unlinkable communication layer. Accordingly, the experiment evaluates whether independently randomized token epochs can be linked from the protocol artifacts themselves, rather than whether the surrounding communication infrastructure provides network-level anonymity.

Theorem 3. *Assuming DDH in $\mathbb{G}$ and the RO model, the unlinkability advantage is negligible across distinct enrollment epochs.*

5.3. Security Discussion

Within the adversary scope defined in Section 4, the game-based security analysis in Section 5.2, together with the proof sketches in Appendix A, establishes the security objective that is meaningful in the blockchain setting: eliminating unbounded offline biometric guessing enabled by on-chain transparency. Concretely, ledger/network adversaries do not obtain an offline verification oracle because no biometric-derived helper data is published on-chain; without the CA-side masking base $M = pk_{CA}^{r}$, the masked shares $\{A_i\}$ cannot be validated against biometric guesses (Theorem 1). This is precisely the attack vector that motivates our design and differentiates it from prior on-chain helper-data approaches.

A snapshot adversary that steals the off-chain token can still mount guessing attempts, but only through the CA

retrieval interface; hence, the success probability is governed by an online budget and biometric entropy (Theorem 1, case ii). This residual surface is acceptable in our model for two reasons: (i) it restores the standard rate-limited online threat profile of practical authentication systems, rather than the blockchain-specific "download-and-guess forever" failure mode; and (ii) it is enforceable by operational controls at the CA interface (e.g., access control, throttling, and audit), whereas offline attempts against public ledger data are fundamentally uncontrollable. Finally, recovering an offline oracle requires violating the CA threshold assumption by corrupting at least $t+1$ nodes, in which case the scheme degrades to an entropy-only bound as expected for BEKD-style constructions. In summary, the analysis fully covers the intended adversary classes and shows that any successful brute-force attack must either remain online and budgeted or break the stated threshold-security assumption.

6. Performance Evaluation

In this section, we implement the proposed decentralized biometric authentication mechanism to assess its performance.

6.1. Experimental Setup

Within the laboratory-scale prototype, biometric inputs are represented by simulated stabilized feature vectors to evaluate the cryptographic and system workflow. Evaluation with sensor-specific biometric variability is reserved for future deployment studies. The off-chain wallet client and CA consortium were implemented in Python 3.10 using `py_ecc` over `secp256k1` [20], with hash functions H_0–H_3 and H_{tag} instantiated as domain-separated Keccak-256 [21]. The CA consortium comprises $n = 3$ Flask-based nodes under a $(t, n) = (1, 3)$ threshold configuration [16]. Biometric inputs were modeled as 128-dimensional Gaussian-distributed feature vectors [22] with $t_{\mathsf{bio}} = 4$. On-chain, the `ParamRegistry`, `SpentSet`, and ERC-4337 wallet contracts were implemented in Solidity 0.8.20 with the Hardhat framework (optimizer enabled, 200 runs), extending OpenZeppelin's `ERC4337Account` with EIP-1271 validation. All experiments were conducted on a local Hardhat network with 12-second block intervals.

6.2. Off-Chain Performance

We benchmark the off-chain cryptographic latency on a Google Cloud `e2-standard-4` instance (4 vCPU Intel Xeon @ 2.2 GHz, 16 GB RAM) running Ubuntu 22.04 LTS with Python 3.10 and `py_ecc` over secp256k1. Each measurement reports the median over 50 trials; final values are the median across three independent sessions. Three warm-up runs are excluded.

Table 3 presents the end-to-end latency breakdown. Two operations dominate: the wallet-side BEKD computation during enrollment (421 ms, 47.3%) and the CA-side threshold reconstruction during retrieval (446 ms, 50.1%), together accounting for 97.4% of the total 891 ms pipeline. Both are governed by $d = 128$ elliptic-curve scalar multiplications over secp256k1 in `py_ecc`: enrollment iterates over all feature indices to compute masked Shamir shares (Z_i, A_i, tag_i), while retrieval reconstructs the masking base M via Lagrange interpolation and re-evaluates all d indices for tag matching. By contrast, the threshold ECDSA signature (13.67 ms), wallet-side Shamir interpolation (3.21 ms), and final ECDSA signing (6.69 ms) are negligible.

Table 4 shows that increasing the CA consortium from $(1, 3)$ to $(5, 10)$ adds only 23 ms of latency (+5.2%), indicating that the dominant computational cost arises from the per-feature elliptic-curve operations rather than threshold coordination. In a production implementation, replacing `py_ecc` with an optimized native library such as `libsecp256k1` could substantially reduce these bottlenecks; quantifying the resulting end-to-end improvement is left for future evaluation.

Table 3. Off-chain latency breakdown ($d = 128$, $t_{\mathrm{bio}} = 4$, threshold $(1, 3)$).

Phase	Operation	Median (ms)	σ (ms)
Enrollment	Wallet (BEKD key + mask)	421.45	3.27
	CA (threshold sign)	13.67	0.29
	Subtotal	435.12	
Retrieval	CA (recon. M + match)	445.56	3.02
	Wallet (interpolate k)	3.21	0.14
	Subtotal	448.77	
Auth	ECDSA sign σ_U	6.69	0.10
End-to-end total		890.58	

Table 4. Threshold scalability ($d = 128$, $t_{\text{bio}} = 4$, 50 runs $\times$ 3 sessions).

(t, n)	Quorum	Median (ms)	σ (ms)	Overhead
$(1, 3)$	2	443.46	2.92	—
$(2, 5)$	3	448.07	2.81	+1.0%
$(3, 7)$	4	457.55	3.00	+3.2%
$(5, 10)$	6	466.55	3.83	+5.2%

6.3. On-Chain Gas Analysis

We benchmark on-chain costs for (i) one-time deployment and (ii) per-use operations, using a common-stack baseline that matches a vanilla ERC-4337-style wallet without freshness enforcement. Concretely, the baseline deploys ParamRegistry + Authorization + VanillaWallet, while our scheme adds SpentSet and a BiometricWallet that burns the token identifier ρ upon successful authentication. Table 5 reports the resulting gas usage and normalized ratios (baseline = 1.00). For the authentication line, we report the first-call cost ("real" = mock + 3000 in our harness) to avoid same-transaction warm-slot artifacts.

Table 5. Condensed on-chain costs (gas).

Category	Ours	Baseline	Ratio
Deploy (stack)	958,143	648,269	1.48×
Reads (total)	8595	5628	1.53×
Writes (total)	47,854	23,085	2.07×
Auth (first, real)	31,122	26,503	1.17×

6.3.1. Overhead Relative to a Vanilla Wallet

Using the first-call authentication cost, freshness enforcement increases on-chain authentication by only 17.4% over the common-stack baseline (31,122 vs. 26,503 gas), i.e., an extra 4619 gas per authentication. The added cost is dominated by the extra SpentSet lookup and the one-time burn of ρ, which together provide replay resistance with minimal changes to the wallet verification path. While write micro-benchmarks show a larger ratio (Table 5), this is expected: our scheme necessarily commits one additional state transition (burning ρ) to enforce one-time token semantics.

6.3.2. Practical Cost (USD Estimate)

To give an intuitive sense of cost, at 30 gwei and 3000 USD/ETH, the first-call authentication corresponds to approximately 0.00093 ETH (≈2.80 USD) for our scheme versus 0.00080 ETH (≈2.39 USD) for the baseline, i.e., about $0.42 extra per authentication. Deployment is a one-time cost: the full stack is about 0.0287 ETH (≈$86) versus 0.0194 ETH (≈$58) for the baseline under the same assumptions. These estimates suggest that the usability impact of on-chain freshness enforcement is modest relative to the security benefit of preventing token replay.

7. Conclusions

This paper addresses a fundamental vulnerability in decentralized biometric authentication: the exposure of low-entropy helper data on transparent ledgers, which enables unbounded offline brute-force attacks. We proposed a threshold-based framework that resolves this transparency–privacy paradox by keeping all BEKD tokens entirely off-chain while delegating only freshness enforcement and signature validation to a minimal on-chain authorization contract. A (t, n)-threshold CA consortium eliminates the single point of failure inherent in traditional certificate authorities, ensuring that no individual node can reconstruct the CA master signing key or serve as an offline verification oracle.

Our game-based security analysis establishes that, under standard hardness assumptions, ledger and network adversaries gain no offline guessing capability (Theorem 1), forging a valid authorization is infeasible without a live biometric sample (Theorem 2), and successive enrollment epochs remain computationally unlinkable (Theorem 3). The Ethereum prototype enables performance evaluation: the full off-chain pipeline completes in under 891 ms, the on-chain freshness check adds only 17.4% gas overhead (31,122 vs. 26,503 gas) over a vanilla ERC-4337 wallet, and threshold scalability from $(1, 3)$ to $(5, 10)$ incurs merely 5.2% additional latency. These results indicate that strong brute-force resistance and replay prevention can be achieved at modest computational and economic cost.

We acknowledge that our evaluation is conducted in a controlled experimental setting with simulated biometric inputs and a local Ethereum testnet; real-world deployment would introduce additional variability from network

latency, sensor noise, and heterogeneous hardware. Future work should explore tighter brute-force resistance bounds in the threshold setting, batched freshness proofs or rollup-based verification to reduce on-chain gas costs, and zero-knowledge attestations to enable privacy-preserving CA-side verification without revealing protocol metadata to the blockchain layer.

Author Contributions

H.C. (Haoze Cheng): Conceptualization, methodology, software, validation, investigation, writing—original draft preparation. H.C. (Hui Cui): Supervision, conceptualization, methodology, writing—reviewing and editing. J.B.: Supervision, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The source code and experimental data supporting the reported results are available at: https://github.com/sAeNrDER/Decentralised-BEKD-for-Biometric-Authentication.

Acknowledgments

The authors acknowledge the Faculty of Information Technology at Monash University for providing institutional access and research support for this work.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used AI-assisted tools to support document formatting and to assist with code generation, debugging, and software-environment configuration for experimental testing. These tools were not used to generate the manuscript's scientific arguments, research findings, or original written content. After using these tools, the authors reviewed and verified all outputs, made revisions where necessary, and take full responsibility for the content of the published article.

Appendix A

Proof. [Proof sketch of Theorem 1] Any offline verification of a biometric guess requires the masking base $M = pk_{CA}^{r}$ to recompute $Z_i = H_1(M, M^{w_i})$. Computing M from $(R_0 = g^r, pk_{CA})$ would solve CDH, and reconstructing sk_{CA} from at most t shares contradicts threshold security, establishing case (i). With a token snapshot, guesses can be tested only via the CA retrieval interface, which evaluates M using $t{+}1$ shares and returns success only if at least t_{bio} components match. Under the min-entropy model, this occurs with probability at most

$$\sum_{j=t_{\mathsf{bio}}}^{d} \binom{d}{j} 2^{-jH_\infty}$$

per attempt, giving case (ii) after q_{CA} trials. If $t{+}1$ CA shares are corrupted, M can be evaluated offline and only the entropy term remains, yielding case (iii). □

Remark 1 (Enforcing the online budget)**.** *SpentSet prevents on-chain replay. Off-chain, each retrieval attempt must present an identifier ρ and is recorded in L_T by the consortium; thus in deployment q_{CA} is bounded by the number of distinct tokens available in the adversary's snapshot view, as in BEKD [13]. In particular, if the consortium*

rejects repeated identifiers via the check $\rho \in L_T$, then

$$q_{\mathsf{CA}} \leq |\{\rho \mid \rho \text{ is contained in the snapshot}\}|.$$

Proof. [Proof sketch of Theorem 2] A successful forgery implies one of the following events: (1) an ECDSA forgery under k; (2) a validly attested but modified token transcript, which requires forging the threshold signature on $m = H_2(R_0, R_1, h_A)$; or (3) recovering k without liveness, which reduces to the budgeted online guessing captured by Theorem 1, case (ii). A union bound over these events yields negligible advantage. □

Proof. [Proof sketch of Theorem 3] Given a DDH challenge $(g, u = g^r, v = pk_{CA}, w)$, where w is either v^r or a uniformly random group element in $\mathbb{G}$, set $R_0 \leftarrow u$ and $R_1 \leftarrow w \cdot g^k$ for fresh k. If $w = v^r$, then (R_0, R_1) matches a real envelope; otherwise, R_1 is uniform given R_0. All remaining values are computed with fresh randomness and domain-separated random oracles, and hence are independent across epochs. Any epoch-linker yields a DDH distinguisher with the same advantage up to $\mathsf{negl}(\lambda)$.

Moreover, since $M = pk_{CA}^r$ is computationally hidden from any admissible view without threshold corruption, $\mathcal{A}$ can query H_1 at the exact points (M, M^{w_i}) only with negligible probability. Hence the induced $\{Z_i\}$, and therefore $\{A_i\}$ and $\{\mathsf{tag}_i\}$, are computationally independent across enrollment epochs in the random-oracle model. □

References

1. Ogbanufe, O.; Kim, D.J. Comparing fingerprint-based biometrics authentication versus traditional authentication methods for e-payment. *Decis. Support Syst.* **2018**, *106*, 1–14. https://doi.org/10.1016/j.dss.2017.11.003.
2. Nasution, M.I.P.; Nurbaiti, N.; Nurlaila, N.; et al. Face Recognition Login Authentication for Digital Payment Solution at COVID-19 Pandemic. In Proceedings of the 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 15–16 September 2020; pp. 48–51.
3. Ryu, R.; Yeom, S.; Kim, S.H.; et al. Continuous Multimodal Biometric Authentication Schemes: A Systematic Review. *IEEE Access* **2021**, *9*, 34541–34557. https://doi.org/10.1109/ACCESS.2021.3061589.
4. Abo Alzahab, N.; Rafaiani, G.; Battaglioni, M.; et al. BiometricIdentity dApp: Decentralized Biometric Authentication Based on Fuzzy Commitment and Blockchain. *SoftwareX* **2024**, *28*, 101932. https://doi.org/10.1016/j.softx.2024.101932.
5. Sharma, S.; Saini, A.; Chaudhury, S. Multimodal Biometric User Authentication Using Improved Decentralized Fuzzy Vault Scheme Based on Blockchain Network. *J. Inf. Secur. Appl.* **2024**, *82*, 103740. https://doi.org/10.1016/j.jisa.2024.103740.
6. Haleem, A.; Ben Jabra, S.; Meddeb, A. Survey on IoT Security Using Biometrics and Blockchain Technology. *Multimed. Tools Appl.* **2026**, *85*, 15. https://doi.org/10.1007/s11042-026-21279-6.
7. Bernabe, J.B.; Canovas, J.L.; Hernandez-Ramos, J.L.; et al. Privacy-preserving solutions for blockchain: Review and challenges. *IEEE Access* **2019**, *7*, 164908–164940.
8. Aggarwal, S.; Kumar, N. Chapter Twenty—Attacks on Blockchain. In *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*; Advances in Computers; Aggarwal, S., Kumar, N., Raj, P., Eds.; Elsevier: Amsterdam, The Netherlands, 2021; Vol. 121, pp. 399–410.
9. Rathgeb, C.; Uhl, A. Statistical attack against iris-biometric fuzzy commitment schemes. In Proceedings of the 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Colorado Springs, CO, USA, 20–25 June 2011; pp. 23–30.
10. Politou, E.; Casino, F.; Alepis, E.; et al. Blockchain mutability: Challenges and proposed solutions. *IEEE Trans. Emerg. Top. Comput.* **2019**, *9*, 1972–1986.
11. Zhang, S.; Yan, Z.; Liang, W.; et al. BAKA: Biometric Authentication and Key Agreement Scheme Based on Fuzzy Extractor for Wireless Body Area Networks. *IEEE Internet Things J.* **2024**, *11*, 5118–5128. https://doi.org/10.1109/JIOT.2023.3302620.
12. Boldyreva, A.; Mohan, D.I.; Tang, T. May the Force Not Be With You: Brute-Force Resistant Biometric Authentication and Key Reconstruction. In Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), Taipei, Taiwan, 13–17 October 2025; pp. 3620–3634.
13. Cui, H.; Yang, X.; Yang, W.; et al. Token-Based Biometric Enhanced Key Derivation for Authentication Over Wireless Networks. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2347–2357. https://doi.org/10.1109/TNSE.2023.3246439.
14. Zhang, S.; Yan, Z.; Liang, W.; et al. BCAE: A Blockchain-Based cross Domain Authentication Scheme for Edge Computing. *IEEE Internet Things J.* **2024**, *11*, 24035–24048. https://doi.org/10.1109/JIOT.2024.3387934.
15. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; et al. Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet Things J.* **2019**, *6*, 2188–2204. https://doi.org/10.1109/JIOT.2018.2882794.
16. Gennaro, R.; Goldfeder, S. Fast Multiparty Threshold ECDSA with Fast Trustless Setup. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), Toronto, ON, Canada, 15–19 October 2018; pp. 1179–1194.
17. *ISO/IEC 24745:2022*; Information Security, Cybersecurity and Privacy Protection—Biometric Information Protection, Ed.

2. ISO/IEC JTC 1/SC 27. ISO: Geneva, Switzerland, 2022; 63 pages.
18. Simoens, K.; Bringer, J.; Chabanne, H.; et al. A Framework for Analyzing Template Security and Privacy in Biometric Authentication Systems. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 833–841. https://doi.org/10.1109/TIFS.2012.2184092.
19. Wang, M.; Rui, L.; Yang, Y.; et al. A Blockchain-Based Multi-CA Cross-Domain Authentication Scheme in Decentralized Autonomous Network. *IEEE Trans. Netw. Serv. Manage.* **2022**, *19*, 2664–2676. https://doi.org/10.1109/TNSM.2022.3180357.
20. Mehrabi, M.A.; Doche, C.; Jolfaei, A. Elliptic Curve Cryptography Point Multiplication Core for Hardware Security Module. *IEEE Trans. Comput.* **2020**, *69*, 1707–1718. https://doi.org/10.1109/TC.2020.3013266.
21. Kanth, V.; Hale, B. Blockchain-Based Authenticated Stego-Channels and Application to Ethereum. *IEEE Trans. Dependable Secure Comput.* **2025**, *22*, 373–387. https://doi.org/10.1109/TDSC.2024.3399696.
22. Wang, C.; Chen, Y.; Liu, F.; et al. Mixture of Gaussian-Distributed Prototypes with Generative Modelling for Interpretable and Trustworthy Image Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 6974–6989. https://doi.org/10.1109/TPAMI.2025.3566425.