Proof-of-Uniqueness: Sybil-Resistant Privacy-Preserving Decentralized Identity through Threshold-OPRF and zk-SNARK Registry

Adam Vozda[1] , Martin Perešíni[1,3] , Juraj Mariani[1] , and Ivan Homoliak[1,2]

[1] Brno University of Technology, Faculty of Information Technology, Božetěchova 1/2, 612 00 Brno, Czechia
[2] Slovak University of Technology in Bratislava, Faculty of Informatics and Information Technologies, Ilkovičova 2, 842 16 Bratislava, Slovakia
[3] Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH, Zürich, Switzerland
xvozdaa00@stud.fit.vut.cz; {iperesini,mariani,homoliak}@fit.vut.cz

Abstract. Several decentralized applications and blockchains, such as blockchain-based voting systems or Proof-of-Social-Capital, require a strict one-account-per-person policy, yet public identity records often expose sensitive attributes or enable offline attribution. This work presents a Proof-of-Uniqueness blockchain-based registry that composes an issuer-signed verifiable credential, two zero-knowledge proofs, a threshold verifiable oblivious pseudorandom function (vOPRF), and a smart contract. The first proof authorizes a blinded evaluation of an issuer's certified canonical identifier, and the second validates the OPRF transcript and binds the resulting global nullifier to a wallet. The design further binds issuer authorization, credential status, validity bounds, canonical encoding, and replay state. Its person-level guarantee is conditional on a stable injective identifier within a coordinated issuance namespace and a stable, valid OPRF key. Confidentiality holds against public observers and against fewer than the OPRF threshold of nodes, but not against an issuer that actively probes its own identifiers. Our prototype couples Noir and UltraPlonk circuits, three local threshold OPRF nodes, and a Solidity contract. For one fixed-schema credential input, it derives a single wallet-bound deterministic nullifier, verifies a real 2,144-byte proof on-chain, and rejects a repeated nullifier with $O(1)$ lookups. One real enrollment consumes 615k gas, and proof construction takes about 65 seconds on one desktop software thread.

Keywords: Sybil resistance · Proof of personhood · Verifiable credentials · zk-SNARK · OPRF · Nullifier · Decentralized identity

1 Introduction

A core prerequisite for many decentralized protocols is enforcing that an individual cannot instantiate multiple identities to gain disproportionate control

or resources. Personhood weighted consensus protocols [4,29], community basic income systems [5], quadratic funding mechanisms [7], and blockchain-based voting systems [19,39,24] all depend on a single entity uniqueness invariant. Achieving that invariant without a central identity authority or specialized hardware remains hard, as the Ghost Trilemma shows for the joint decentralized verification of *sentience, location, and uniqueness* [32]. Biometric deduplication needs dedicated hardware operators, social graph analysis gives probabilistic guarantees and exposes connections, and document schemes need complex legacy signature checks inside zero-knowledge circuits (Section 8), reflecting broader evaluations of decentralized identity architectures where a perfect balance of privacy, decentralization, and Sybil-resistance remains elusive [13].

Credentials alone do not solve the public deduplication problem, because conventional signatures expose the claims to a verifier [38]. A zk-SNARK conceals those claims [34], yet public tags such as $H(\text{identity})$ stay open to offline dictionary attacks, because human attributes carry limited entropy. A public uniqueness tag must therefore be deterministic, unavailable for unrestricted offline evaluation, and bound to the credential and destination wallet.

Our Work. This work proposes a standalone Proof-of-Uniqueness (PoU) registry that addresses this composition requirement. An issuer certifies an opaque canonical identifier within a W3C model credential, a first zero-knowledge proof (π_1) authorizes a blinded request to a threshold vOPRF committee, and a second proof (π_2) verifies the issuer signature and the OPRF transcript, derives a deterministic global nullifier, and binds it to a public wallet. An EVM smart contract verifies π_2, rejects duplicate nullifiers, and manages record lifecycle events. Membership nullifiers, issuer-based pseudonyms, and threshold OPRFs have been explored separately [36,10,22], and World ID 4.0 likewise combines proof stages, authenticated threshold vOPRFs, and public registries [40]. Our contribution occupies a distinct systems point: it turns a stable credential identifier into one application-independent registry record that serves as a cross-service uniqueness oracle, at the cost of a persistent public pseudonym.

Contributions. The core contributions of this work are the following.

- We specify two proof relations with explicit bindings among signed credential fields, issuer authorization, request identifiers, validity and status data, holder keys, OPRF transcripts, wallets, and nullifiers.
- The lifecycle design uses current issuer and credential status, validity intervals, generations, and bounded cleanup. The Solidity prototype implements constant lookup, duplicate rejection, wallet binding, expiration, deletion, and issuer administration, with the stated gaps.
- We evaluate our approach on a local 2-of-3 threshold OPRF deployment, 2 UltraPlonk circuits, real proof fixtures, and Foundry contract tests.
- We identify the conditions that lift record equality to person-level uniqueness, and we analyze issuer probing, linkability, threshold compromise, replay, recovery, governance, and unresolved master key migration.

2 Background

Verifiable credentials and zk-SNARKs. The W3C Verifiable Credentials Data Model defines three roles [38]: an issuer signs a credential, a holder stores it, and a verifier checks it. A plain signature check discloses every signed claim to the verifier, and a zk-SNARK removes that disclosure, because the holder proves in zero knowledge that an authorized issuer signed a credential and that a public statement follows from its contents [34]. This capability has been successfully adapted for various privacy-preserving applications, such as the verification of NFT-issued skill certificates [25]. The cost of such a circuit depends on the primitives inside it. Credential circuits therefore use arithmetization-friendly primitives: an elliptic curve whose base field equals the scalar field of the proof system, which makes signature verification native in the circuit [41], and an algebraic hash family that supplies the hash and the Merkle tree at a low constraint cost [16,17]. A proving toolchain compiles the circuit to an intermediate representation, and a backend of the PLONK family then produces a constant-size proof under a universal setup [15,23].

Nullifiers and pseudonym scope. A nullifier is a public tag that a verifier stores and rejects on a second use, and its scope decides what an observer can link [31]. A context-specific tag is a separate pseudonym in each application, so it limits cross-service tracking, but it cannot deduplicate across applications. A global tag is one value per subject, so it supports shared deduplication and it also becomes a persistent public pseudonym. A tag that the holder computes alone is only as unpredictable as its private input, so a low-entropy input admits an offline dictionary attack.

Oblivious pseudorandom functions. An OPRF lets a client evaluate $F_k(x)$ without revealing x to the server and without learning the key k, and a verifiable OPRF (vOPRF) adds a proof that the server used the committed key, which a Discrete Logarithm Equivalence (DLEQ) proof supplies over a prime-order group. RFC 9497 defines the OPRF and VOPRF modes and registers their suites [11]. A threshold OPRF secret-shares k over n nodes with reconstruction threshold t, so fewer than t nodes cannot evaluate F_k and no single node holds k [21]. Proactive secret sharing refreshes the shares within an epoch without changing k, so every earlier output stays valid [18].

3 System Model and Requirements

Actors. The system model comprises four distinct trust domains, shown in Figure 1. The **issuer** verifies identity evidence and signs a credential for the holder, and the **holder and local prover** stores the credentials and private keys, contacts OPRF nodes, and generates proofs. The **threshold OPRF committee** has n nodes that hold shares k_i, of which at least t are required for evaluation, and the **registry contract** checks proofs and current policy data, then maintains active nullifier records.

Identity and Issuance Assumptions. Let $\mathsf{uid}_I(c)$ denote the canonical subject identifier certified within credential c under issuance domain ρ, and let $P(c)$ denote its subject. Person-level deduplication requires two explicit conditions.

$$\begin{aligned} P(c) = P(c') &\Longrightarrow \mathsf{uid}_I(c) = \mathsf{uid}_I(c') \quad \text{(Renewal invariance)}, \\ P(c) \neq P(c') &\Longrightarrow \mathsf{uid}_I(c) \neq \mathsf{uid}_I(c') \quad \text{(Subject injectivity)}. \end{aligned} \tag{1}$$

These assumptions separate cryptographic deduplication from issuance policy, so credential renewal and holder key updates preserve deduplication while uid_I stays invariant. Multiple issuer signing keys within one domain are authorized rotations over the same namespace, and independent issuers that assign unrelated identifiers give no cross-issuer person uniqueness.

Design Requirements. We evaluate nine explicit requirements. **(R1) Deduplication** rejects a second active registration for one canonical person identifier, and **(R2) holder control** authorizes enrollment for one exact wallet. **(R3) Lost key recovery** rebinds a record after holder or wallet key loss, and **(R4) durable revocation** prevents replay of earlier enrollment or revocation authorizations. **(R5) Credential lifecycle** enforces validity bounds, current issuer trust, and credential status, and **(R6) ordinary client equipment** requires no dedicated biometric sensor. **(R7a) Attribute confidentiality** reveals no private claim beyond the public statement and traffic metadata, and **(R7b) protocol decentralization** distributes OPRF evaluation and permits public registry execution. **(R8) Accountable governance** manages issuers, epochs, and keys under an auditable policy.

Threat Model and Observation Boundaries. Aligning with security reference architectures for blockchains [20], we define observation boundaries: the issuer observes identity evidence during issuance and knows the canonical identifier uid_I. The holder observes all private claims, the derived identity secret H_{id}, and the blinding scalar β that randomizes the OPRF request and hides H_{id} from the committee. OPRF nodes observe the target statement $\mathbf{x}_1$, the authorization proof π_1, and network and timing metadata, while public blockchain observers view $\mathbf{x}_2$, the enrollment proof π_2, and transaction metadata. The target statement therefore reveals the policy roots, the issuer key, the relevant times, the epoch, the public OPRF key K, the wallet w, and the global nullifier nf. The evaluated π_1 instead exposes only the request identifier rid and the coordinates of the blinded point A, so it omits issuer, policy, and time bindings. The identifier rid is a fresh committee challenge that ties one authorized request to one evaluation. We assume that fewer than t OPRF nodes collude, that the configured K is a valid prime subgroup point, and that the setup material and implementations are correct. The person-level property additionally assumes honest canonical issuance, and privacy from an issuer requires that it issues no shadow credential and runs no surveillance evaluation, because otherwise it reproduces a subject's nullifier and searches the registry.

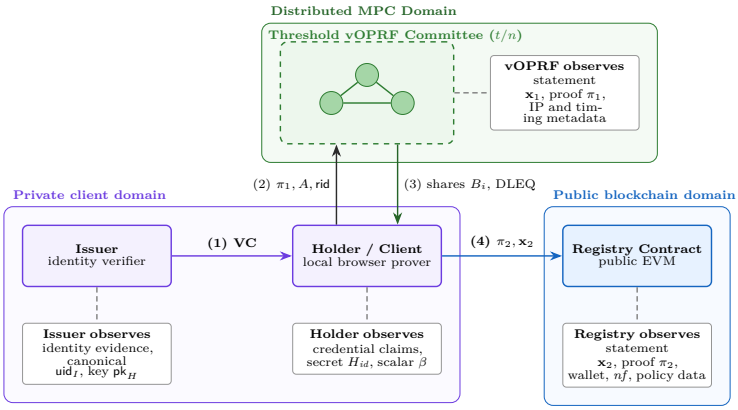

Figure 1. System architecture and observation boundaries. Each domain observes only the data in its card: the client holds the credential and the identity secret H_{id}, the committee sees π_1 and the blinded point A but not H_{id}, and the blockchain sees $\mathbf{x}_2$ and the nullifier nf. Network metadata stays observable in every domain.

4 Proof-of-Uniqueness Design

This section specifies the target protocol, which includes safeguards that the evaluated prototype omits (Section 5).

Overview. The registry is a smart contract on a public blockchain that holds one record per registered person and no credential claim. Each record is keyed by a *nullifier*, a public tag that names one person independently of any wallet or application. The contract accepts a record only when its tag is absent from the current state, so a second registration of the same person fails, and any application reads the resulting state as a shared uniqueness oracle. Figure 1 shows the four trust domains and the data that each domain observes.

Figure 2 shows the enrollment sequence, which has four steps. First, the holder derives one private identity input H_{id} from the issued credential, and it blinds the curve point that encodes H_{id} with a fresh scalar β. Second, the holder sends the blinded point to the threshold OPRF committee together with the authorization proof π_1, and the committee returns partial evaluations with proofs of correct key use. Third, the holder combines and unblinds those shares, and it derives the nullifier nf from the result. Fourth, the holder submits the enrollment proof π_2 to the contract, which verifies it, applies the current policy, and stores nf against the wallet w.

Strawman: a public registry of credential hashes. Consider the simplest construction that satisfies deduplication. The issuer certifies a canonical subject identifier, the holder discloses the credential to a verifier, and the verifier writes $H(\rho, \mathsf{uid}_I)$ into the registry. Duplicate detection works, because one person always yields one tag, but privacy fails in two ways: the verifier learns every claim, and the public tag admits an offline dictionary attack, because a certified identifier carries

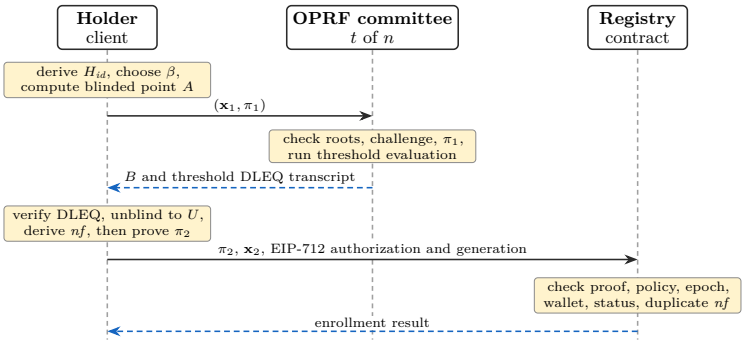

Figure 2. Target enrollment sequence, in the four steps of the Overview. The prototype omits the issuer root, credential status, time challenge, and epoch generation.

little entropy and has a public format. Any observer can therefore enumerate candidates, recompute the tag, and deanonymize a record.

1st refinement: ZK proofs. A zk-SNARK removes the disclosure to the verifier: the holder proves that an authorized issuer signed a credential and that the published tag is the correct function of the certified identifier inside it. No claim leaves the client, and the contract needs no trusted verifier. The offline attack survives, because the tag remains a public function of a low-entropy input. Holder-chosen randomness in the tag defeats that attack, but it also defeats deduplication: the holder picks fresh randomness for every attempt and mints unlimited distinct tags from one credential. A sound tag therefore depends on a secret that no holder controls and that is identical for every holder.

2nd refinement: a threshold vOPRF. A single service that holds one PRF key k meets that requirement, but it learns every queried identifier and can enumerate the population offline. An oblivious PRF removes this exposure, since the service evaluates F_k on the blinded point and learns neither the input nor the output. A *verifiable* OPRF forces the service to use the committed key, so a deviating node cannot silently change the output and break determinism. Threshold evaluation over n nodes with reconstruction threshold t then removes the single point of trust, because no node alone holds k. One gap remains: an outsider can query the committee on guessed identifiers and use it as an online enumeration oracle. Proof π_1 closes that gap by restricting queries to holders of a valid credential, which makes guessing a rate-limited online attack.

Composition and canonical encoding. The design binds four checks: π_1 ties a fresh OPRF request to a credential signed by an authorized issuer, the threshold evaluation fixes the output as a function of H_{id} and the committee key alone, π_2 rechecks the credential and the evaluation while binding the nullifier to one wallet, and the registry checks current policy, generation, and duplicate state. The second check stops a holder from deriving a second tag from one credential. The credential profile uses the signed vector

$$\mathbf{v} = (\mathsf{pk}_{H,x}, \mathsf{pk}_{H,y}, \mathsf{uid}_I, \mathsf{dob}, \rho, \mathsf{name}, \mathsf{nationality}, \mathsf{birthplace}, \mathsf{sex}, t_{\mathrm{from}}, t_{\mathrm{until}}, \mathsf{vcid}).$$

Here $\mathsf{pk}_H = (\mathsf{pk}_{H,x}, \mathsf{pk}_{H,y})$ is the holder public key, uid_I is the canonical subject identifier that the issuer assigns, and the tag ρ names the issuance domain that scopes that identifier. The field vcid is the credential identifier used for status lookup, dob is the date of birth, t_{from} and t_{until} are the validity bounds, and the remaining fields carry ordinary subject attributes. Each value is encoded as a typed, length-prefixed sequence of field chunks: text is UTF-8 in Unicode NFC form, dates are UTC integers with fixed units, and every schema and domain tag is fixed. The symbol t denotes the verification time that a relation compares against the validity interval, and its two instantiations are the OPRF query time t_q and the enrollment time t_e. Times satisfy $0 \le t_{\text{from}}, t, t_q, t_e, t_{\text{until}} < 2^{64} < p$, while epochs e and record generations g satisfy $0 \le e, g < 2^{64}$, with range constrained integer decompositions for the circuit comparisons. For the bounded profile, the tagged representation is injective before hashing, and security after hashing rests on collision resistance. The signed leaves are $L_j = \text{Poseidon}(\ell_j, \text{Enc}(v_j))$ for $0 \le j < 12$, followed by four fixed zero leaves, which removes leading zero, type, and concatenation ambiguities. Let R_I be the current issuer anchor root and R_S the current credential status root, where authorized governance updates R_I and each issuer or an appointed status authority authenticates its R_S update to the registry. Signature verification includes scalar-range checks and prime subgroup checks on public keys and signature points, and cryptographic hashes use Domain Separation Tags (DST) for independent outputs across contexts. For membership witnesses μ_I, μ_S and 16 leaf root function MR_{16}, the credential predicate holds exactly when

$$\begin{aligned} L_j &= \text{Poseidon}(\ell_j, \text{Enc}(v_j)) \quad (j < 12), \\ L_j &= 0 \quad (12 \le j < 16), \qquad R = \mathsf{MR}_{16}(\mathbf{L}), \\ m_I &= \text{Poseidon}(DST_{VC}, \mathsf{schemaId}, R), \\ \text{Vfy}_{\text{EdDSA}}(\mathsf{pk}_I, m_I, \sigma_I) &= 1, \\ \text{Member}(R_I, H(\mathsf{pk}_I), \mu_I) &= 1, \\ \text{Active}(R_S, \rho, \mathsf{vcid}, \mu_S) &= 1, \quad t_{\text{from}} \le t < t_{\text{until}}. \end{aligned} \tag{2}$$

We denote their conjunction by $\mathsf{CV}(\mathbf{v}, \mathbf{L}, \sigma_I, \mu_I, \mu_S; \mathsf{pk}_I, R_I, R_S, t)$. In particular, $\mathsf{uid}_I = v_2$, $\rho = v_4$, $\mathsf{pk}_H = (v_0, v_1)$, and both validity bounds are issuer signed. The stable private input is $H_{id} = \text{Poseidon}(DST_{id}, \rho, \mathsf{uid}_I)$, which no volatile claim enters.

Authorization relation $\mathcal{R}_1$. Let $\mathbb{G}$ be the prime order Baby Jubjub subgroup with order q, distinct from the BN254 circuit field modulus p, and let $\mathbb{G}^* = \mathbb{G} \backslash \{\mathcal{O}\}$. The map EncodeToCurve is deterministic, domain separated, and produces a point in $\mathbb{G}^*$ [12]. The holder samples a fresh blinding scalar $\beta \in \mathbb{Z}_q^*$, which conceals the encoded identifier from the committee. The public statement is $\mathbf{x}_1 = (\mathsf{rid}, A, \mathsf{pk}_I, R_I, R_S, t_q)$, where rid is a fresh committee challenge and t_q is its issue time. The committee issues an authenticated unpredictable challenge with at least 128 bits of entropy, binds it to the current roots and expiry, and atomically reserves it in a shared replay set before returning shares. With the

signed vector and credential witnesses kept private, relation $\mathcal{R}_1$ enforces

$$\begin{aligned}
\mathsf{CV}(\mathbf{v}, \mathbf{L}, \sigma_I, \mu_I, \mu_S; \mathsf{pk}_I, R_I, R_S, t_q) &= 1, \\
H_{id} &= \text{Poseidon}(DST_{id}, \rho, \mathsf{uid}_I), \\
M &= \text{EncodeToCurve}(H_{id}), \quad M, A \in \mathbb{G}^*, \\
A &= \beta M, \quad \beta \in \mathbb{Z}_q^*, \\
m_H &= \text{Poseidon}(DST_{auth}, \mathsf{rid}, A_x, A_y), \\
\text{VerifyEdDSA}(\mathsf{pk}_H, m_H, \sigma_H) &= 1.
\end{aligned} \tag{3}$$

Threshold vOPRF evaluation. Informally, this step turns the private identity input into the public uniqueness tag: the committee applies a keyed function that no participant can evaluate alone, and it learns neither the identifier nor the result. The holder learns the tag but never the key, so it cannot produce a second tag, and an observer who holds the tag cannot invert it. The nonzero share indices α_i are distinct in $\mathbb{Z}_q$, and $k \in \mathbb{Z}_q^*$. Node i publishes $K_i = k_i G$, returns $B_i = k_i A$, and proves $\text{DLEQ.Verify}(G, K_i, A, B_i) = 1$. For transcript $T_i = (G, K_i, A, B_i)$, the target defines $e_D = \text{HashToScalar}(DST_{DLEQ}, T_i) \in \mathbb{Z}_q$ and range constrains every response scalar in $\mathbb{Z}_q$. For a set S of at least t valid responses, the client uses

$$\lambda_i = \prod_{\substack{j \in S \\ j \neq i}} \frac{-\alpha_j}{\alpha_i - \alpha_j} \bmod q, \qquad B = \sum_{i \in S} \lambda_i B_i = kA, \qquad K = \sum_{i \in S} \lambda_i K_i = kG.$$

The client rejects identity, off curve, or non-subgroup points, obtains $U = \beta^{-1} B$, and derives

$$nf = \text{Poseidon2}(DST_{nf}, H_{id}, U_x, U_y). \tag{4}$$

Enrollment relation $\mathcal{R}_2$. For OPRF epoch e, the target public statement is $\mathbf{x}_2 = (e, K, t_e, t_{\text{from}}, t_{\text{until}}, \mathsf{pk}_I, R_I, R_S, w, nf)$, over the same signed fields as $\mathcal{R}_1$. For an evaluation transcript τ and private A, B, U, β, relation $\mathcal{R}_2$ enforces

$$\begin{aligned}
\mathsf{CV}(\mathbf{v}, \mathbf{L}, \sigma_I, \mu_I, \mu_S; \mathsf{pk}_I, R_I, R_S, t_e) &= 1, \\
H_{id} &= \text{Poseidon}(DST_{id}, \rho, \mathsf{uid}_I), \\
m_w &= \text{Poseidon}(DST_{enr}, e, H_{id}, w), \\
\text{VerifyEdDSA}(\mathsf{pk}_H, m_w, \sigma_w) &= 1, \\
A &= \beta\, \text{EncodeToCurve}(H_{id}), \quad \beta \in \mathbb{Z}_q^*, \\
\text{DLEQ.Verify}(G, K, A, B; \tau) &= 1, \quad \beta U = B, \\
K, A, B, U &\in \mathbb{G}^*, \quad 0 \leq w < 2^{160}, \\
nf &= \text{Poseidon2}(DST_{nf}, H_{id}, U_x, U_y).
\end{aligned} \tag{5}$$

The registry pins e, K, R_I, and R_S, checks that t_e is recent and the validity interval is current, stores the policy roots used by the proof, and rejects a duplicate nf. The wallet authorizes the exact proof, signals, contract, blockchain, deadline, and record generation using EIP-712 [3].

Lifecycle and recovery. Each record carries a monotonic generation g, a lifecycle state, and its proven policy roots. Authorizations include g and a deadline, and revocation increments g while retaining a tombstone, which defeats archived replay. Validity requires a current interval and record roots equal to the current R_I, R_S, so an authenticated root change invalidates predecessor records at once, and unaffected holders renew by reusing $\mathcal{R}_2$ for the same nf, current roots, and g. This conservative rule sacrifices availability, and a selective accumulator or public status handle requires separate privacy analysis. Bounded pagination reclaims storage without determining validity.

Recovery after loss of pk_H requires issuer reissuance with the same uid_I and a new holder key. The issuer must revoke the old credential in an authenticated R_S update before or atomically with recovery. Statement $(e, K, nf, w', g, R_I, R_S, t)$ then rechecks the replacement credential and the OPRF relation as in $\mathcal{R}_2$, deriving the existing nf and binding w'. Only an active record with the current epoch, roots, and generation can transition, and the registry atomically supersedes the old binding and increments g, after which stale credential proofs fail. The prototype does not implement this target recovery protocol.

5 Implementation

Software stack. The prototype comprises a TypeScript and React client, two Noir circuits, a local 2-of-3 TACEO OPRF service, and a Solidity registry. Noir 0.36 compiles each circuit to the Abstract Circuit Intermediate Representation (ACIR), and the Barretenberg backend proves it with UltraPlonk [33,2]. The OPRF service instantiates the TACEO threshold nullifier protocol over Baby Jubjub with Poseidon2 [22], which is not among the suites that RFC 9497 registers [11]. Its credential is a fixed 12-field profile in a 16-leaf Poseidon tree with a custom Baby Jubjub EdDSA signature, so it is a W3C-shaped demonstration credential, not one under a standardized Data Integrity cryptosuite [27]. All 30 Foundry tests pass, including real proof verification and duplicate rejection, and the source, artifacts, fixtures, and measurement records are in the artifact repository.

Compiled circuit size. The authorization artifact contains 59,221 ACIR opcodes at witness index 76,288, and the enrollment artifact contains 120,980 opcodes at witness index 160,138, where both figures are decoded Noir 0.36 opcode vector lengths, not backend gate counts. Both proofs are 2,144 bytes.

Security basis. Acceptance of π_1 and π_2 is an argument about the exact compiled relations and verification keys. The claim assumes knowledge soundness and zero knowledge of the deployed UltraPlonk system, binding KZG commitments, a correctly generated universal structured reference string, Fiat Shamir in the random oracle model [15,23], unforgeable Baby Jubjub EdDSA, collision-resistant Poseidon and Poseidon2 [16,17], correct subgroup checks, and correct implementations. Under these assumptions, an accepted proof has a witness satisfying the stated relation, and zero knowledge reveals no more than the public statement, so

it does not hide the issuer, validity data, wallet, global equality tag, or network metadata.

Differences from the target design. The prototype establishes deterministic duplicate rejection for one identical concrete input, and Table 2 reports its coverage of the nine requirements, so this paragraph states only the gaps that the requirement view cannot express. Circuit π_1 keeps pk_I private and proves only that some key signed the credential, so a self-signed credential passes the authorization check and the evaluated service has no effective rate control boundary. The field encoder omits type tags, length tags, and Unicode normalization, the four padding leaves are signed but not constrained to zero, and the issuer signs the tree root alone. Mock issuance draws a fresh random subject identifier, so reissuance changes the nullifier nf and breaks renewal invariance. The contract range checks the public OPRF key K but does not validate its prime subgroup membership, which the upstream circuit audit requires of an integrating verifier [26], and replacing K leaves earlier records active. The compiled DLEQ gadget constrains the response scalar below q, but it interprets a BN254 Poseidon2 challenge modulo the group order instead of the target HashToScalar value. The contract further truncates the wallet field to 160 bits and grants each owner unilateral administrative authority. Recovery and key migration have no circuit, contract transition, or test, and the measurements use a local committee rather than a live distributed exchange.

6 Evaluation

Experimental setup. Client proving performance was measured on an Intel Core i5-10400 CPU at 2.90 GHz with six physical cores, twelve hardware threads, and 15 GiB RAM under Linux, with Barretenberg limited to one software thread. Runs used Node.js 18.20.2 with WebCrypto enabled, Noir 0.36.0, the Barretenberg wrapper 0.36.0, and `bb.js` 0.58.0, and gas consumption was measured with Foundry Forge 0.2.0 and solc 0.8.34 at 200 optimizer runs under the EVM Paris ruleset.

Client proving. Authorization proof π_1 requires a median of 1.554 s for witness generation and 21.603 s for proving, over 3 runs, and enrollment proof π_2 requires 3.348 s and 43.389 s in a single run, so the two proof construction stages take

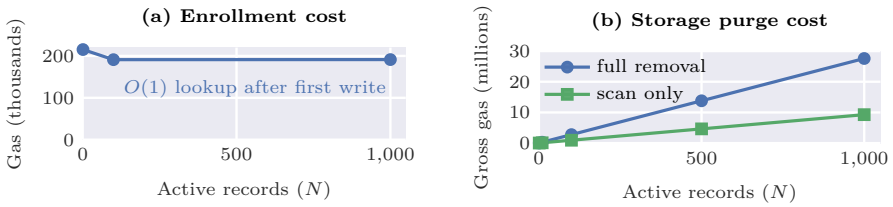

Figure 3. Registry scaling measurements. (a) shows the first write initialization effect and size-independent later mock enrollments. (b) shows the scan and total removal slopes of Table 1.

Table 1. Foundry gas profile using solc 0.8.34 and EVM Paris.

Transaction or item	Gas or size	Transaction or item	Gas or size
Verifier deployment	2,526,248 gas	Proof verification	376,631 gas
Registry deployment	1,562,760 gas	Real enrollment at $N = 0$	614,701 gas
Verifier creation / runtime	14,059 / 11,237 B	Mock enrollment at $N = 100$	191,111 gas
Registry creation / runtime	7,173 / 6,613 B	Wallet revocation	52,297 gas
Purge scan slope	9,251 gas/record	Purge total removal slope	27,551 gas/record

about 65 s, or about 70 s with witness generation. The single π_2 observation does not support a variance estimate, and the benchmark synthesizes the OPRF transcript locally, so it excludes committee and network latency.

Contract gas and $O(1)$ ***scaling.*** Table 1 lists deployment and transaction execution costs, and one complete real enrollment into an initially empty registry consumes 614,701 gas, of which proof verification accounts for 376,631 gas. Mock verifier tests at $N \in \{1, 100, 1000\}$ consume 215,272, 191,111, and 191,309 gas, where the first insertion is 12.64% higher because it initializes the collection state. The flat later values support $O(1)$ registry lookup, but a real proof was not retested at every N.

Storage maintenance scaling. Figure 3(b) plots linear maintenance costs, and linear regression over storage purges gives $G_{\text{scan}}(N) = 24{,}810.8 + 9{,}251.1N$ and $G_{\text{remove}}(N) = 21{,}470.7 + 27{,}550.5N$, both with $R^2 > 0.999999$. Here G_{remove} is the complete scan and deletion cost, not deletion overhead alone.

EVM verifier profile. The generated UltraPlonk verifier makes 39 `ecMul` calls, 39 `ecAdd` calls, and one `ecPairing` call over two pairs, which the applicable gas schedule prices at 6,000 gas per `ecMul` and 45,000 gas plus 34,000 gas per pair [8], so the verifier and the mapping lookup have a fixed execution structure independent of registry size.

7 Discussion and Security Analysis

Conditional uniqueness and issuer trust. Identical H_{id} values under key k produce identical nf values, but the person-level implication also requires Equation 1 and a coordinated namespace. An adversary that holds two credentials with distinct canonical identifiers for one person derives two distinct nullifiers,

Table 2. Requirement coverage in the evaluated prototype.

Requirement	Status	Evidence and boundary
(R1) Deduplication	Record only	Repeated nf is rejected. Reissuance is untested.
(R2) Holder binding	Yes	Holder EdDSA, signals, and EIP-712 bind the wallet.
(R3) Recovery	Design only	No recovery circuit, transition, or test exists.
(R4) Durable revocation	No	No generation or tombstone stops archived replay.
(R5) Lifecycle	Partial	`validUntil` is checked, start time and status are not.
(R6) Client equipment	Desktop only	No dedicated sensor is needed. Mobile is untested.
(R7a)Confidentiality	Partial	Proofs hide private fields. Metadata and probing remain.
(R7b)Decentralization	Partial	Node code exists. No live committee run was measured.
(R8) Governance	Partial	Any owner can change policy or K without a timelock.

and no check in the protocol rejects the second record, because the registry observes nullifiers alone. The issuer also sits inside the privacy boundary: it knows (ρ, uid_I), so it can issue a shadow credential, evaluate it, and search the registry. Issuer anchoring blocks outsider self-signing but not issuer probing, and avoiding that probe requires a stable secret that no single issuer holds.

Transaction relabelling protection. Relation $\mathcal{R}_2$ and the EIP-712 signature both bind the wallet, so replacing w with w' invalidates them and copying the complete transaction can only enroll w, while durable replay protection after deletion still needs the target generation mechanism.

Threshold failure and key compromise. Any t colluding active nodes can evaluate candidate inputs as an online oracle, and an adversary that extracts and combines t same-epoch shares can reconstruct k. The key does not invert Poseidon2, but it enables offline guessing by computing each full candidate nullifier from Equation 4. Proactive share refresh protects against a mobile adversary that never obtains t shares within one epoch [18], but it does not repair a reconstructed master key.

Linkability scope. The global nullifier in Equation 4 is a persistent pseudonym associated with wallet w. This linkability is intentional for unified uniqueness ecosystems, such as Proof-of-Social-Capital consensus [29,30], global basic income [5], or ecosystem-wide quadratic voting [7], which share the eligibility state on purpose. Otherwise, a context tag $nf_{\mathsf{ctx}} = \text{Poseidon2}(DST_{nf}, \mathsf{ctx}, H_{id}, U_x, U_y)$ reduces tracking, at the cost of global duplicate detection [31].

OPRF key stability and migration. Master key rotation $k \mapsto k'$ alters all derived nullifiers, which lets individuals enroll again and bypass duplicate checks. Proactive secret sharing refreshes the node shares k_i without modifying k, so it preserves nullifier stability, but the prototype setter changes K without invalidating old records or storing an epoch, so it must not serve as a production rotation procedure. A secure migration protocol would need to prove that the old and new nullifiers ($nf_{\text{old}} = F_k(H_{id})$ and $nf_{\text{new}} = F_{k'}(H_{id})$) belong to the same credential. Since achieving this without linking the two records on a public ledger remains an open challenge, our current design does not support master key migration.

Production requirements. A production deployment requires six additions: issuer public keys anchored inside circuit π_1, attribute canonicalization across credential renewals, persistent generation counters in enrollment signatures, paginated storage purges, threshold OPRF nodes on independent infrastructure, and a formal audit of the circuit constraint systems.

8 Related Work

Personhood systems. World ID 4.0 targets proof of personhood at population scale. Its identity root is an iris credential from a dedicated Orb device or a

Table 3. Comparison of uniqueness architectures. Endpoint equipment is the dedicated biometric or document hardware required of a participant. RP and TEE denote relying party and trusted execution environment.

System	Identity root	Public scope	Threshold primitive	Endpoint equipment	Available form
World ID 4.0 [40]	Orb or document credential	RP	vOPRF	Orb or NFC	Public proposal
Self [35]	ePassport chip	Application proof	None	NFC and TEE	Open framework
CanDID [28]	Provider records	Application identifier	Committee MPC	Ordinary device	Research prototype
SyRA [10]	Issuer credential	Context	Threshold VRF	Ordinary device	Formal protocol
Semaphore [36]	Group membership secret	Group and scope	None	Ordinary device	Open library
Anon Aadhaar [1]	Signed Aadhaar QR and secret	Application proof	None	Ordinary device	Open library
This work	**Stable credential identifier**	**Global registry**	**Threshold vOPRF**	**Ordinary device**	**Design and prototype**

document credential from a registered issuer, and an authenticated threshold vOPRF across two proof stages produces its uniqueness tag. Version 4 scopes that tag to one relying party, which blocks correlation across services but removes any single shared uniqueness state [40]. BrightID instead derives uniqueness from a social graph of video-verified meetings, so its guarantee is probabilistic and its edges expose social contacts [6]. Pseudonym parties issue one token per attending body at a physical event [14], and a recent framework places these constructions under one cryptographic definition [9].

Credential-based identity. Self converts a government document into an on-chain attestation: its application reads the NFC chip of an ePassport or a chip identity card, proves the document signature chain on the device, and discloses one attribute or one application-scoped tag [35]. Anon Aadhaar proves the issuer signature over the Indian Aadhaar QR code inside a circuit and reveals selected fields only, so it needs no external service, and any party that obtains the same document bytes also computes its tag [1]. CanDID imports attributes from unmodified web providers through oracles, and its committee runs MPC over a secret-shared table of deduplication attributes, such as a national number, which rejects a second identity, screens sanctions, and supports key recovery [28]. Coconut adds threshold issuance, so no single issuer mints a credential alone [37].

Nullifier constructions. Semaphore proves membership in a Merkle tree of identity commitments and emits one signal under a scope tag, so a member signals at most once per scope and stays anonymous inside its group, but it carries no personhood root, because group admission sits outside the protocol [36]. SyRA turns a legacy identifier of low entropy into a key of high pseudoentropy: a distributed issuer evaluates a verifiable random function on that identifier, so a user derives at most one unlinkable pseudonym per context and the issuer keeps no per-user state [10]. The TACEO nullifier protocol specifies a verifiable threshold OPRF over Baby Jubjub with Poseidon2, with blinded evaluation and per-node DLEQ proofs [22].

Architectural comparison. Table 3 compares these architectures. World ID 4.0 shares the two proof stages and the authenticated threshold vOPRF, but its per-relying-party scope excludes the single global record that this work publishes, and Self and Anon Aadhaar root uniqueness in a state document, so they need a reader device or one national scheme. CanDID and SyRA also distribute the deduplication authority, but they bind the result to an application or a context instead of one public registry. A production W3C Data Integrity integration

further requires a documented cryptosuite with a canonical transformation, hashing, and verification procedure [27].

9 Conclusion

We presented a privacy-preserving Proof-of-Uniqueness registry that unites verifiable credentials, two zero-knowledge proofs, a threshold vOPRF, and a smart contract. The architecture establishes a conditional deduplication guarantee: identical issuer-canonical inputs under a stable master key yield a single wallet-bound nullifier, while credential claims stay off-chain. The evaluation shows constant-cost enrollment at 614,701 gas and single-thread desktop proving in 65 seconds. The work gives a testable foundation for decentralized identity deduplication, and it states the operational trade-off between global cross-service uniqueness and context-bound anonymity. A production deployment still requires the six additions listed above, and the source code is available in an open GitHub repository.

Acknowledgement. This work was supported by the Science Grant Agency VEGA under project 1/0771/26, and by the Brno University of Technology internal project FIT-S-23-8151.

Declarations. The authors declare no competing interests. The authors used generative AI (OpenAI ChatGPT-5 and Google Gemini 2.5 Pro) to assist with language polishing, rephrasing, and LaTeX formatting. The authors take full responsibility for the manuscript.

References

1. Anon Aadhaar: Anon Aadhaar. https://github.com/privacy-scaling-explorations/anon-aadhaar (2024)
2. Aztec Labs: Barretenberg.js v0.58.0. Official GitHub release (2024)
3. Bloemen, R., Logvinov, L., Evans, J.: EIP-712: Typed Structured Data Hashing and Signing. EIP 712, Ethereum Improvement Proposals (Sep 2017)
4. Borge, M., Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Ford, B.: Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies. In: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 23–26 (2017). https://doi.org/10.1109/EuroSPW.2017.46
5. Brenzikofer, A.: encointer – Local Community Cryptocurrencies with Universal Basic Income (2019). https://doi.org/10.48550/arXiv.1912.12141
6. BrightID: BrightID: Universal Proof of Uniqueness. White paper, https://www.brightid.org/whitepaper (2022)
7. Buterin, V., Hitzig, Z., Weyl, E.G.: A Flexible Design for Funding Public Goods. Management Science **65**(11), 5171–5187 (2019). https://doi.org/10.1287/mnsc.2019.3337
8. Cardozo, A.S., Williamson, Z.: EIP-1108: Reduce alt_bn128 Precompile Gas Costs. EIP 1108, Ethereum Improvement Proposals (May 2018)

9. Choudhuri, A.R., Garg, S., Lee, K., Montgomery, H., Policharla, G.V., Sinha, R.: A Cryptographic Framework for Proof of Personhood. Cryptology ePrint Archive, Paper 2026/333 (2026)
10. Crites, E., Kiayias, A., Kohlweiss, M., Sarencheh, A.: SyRA: Sybil-Resilient Anonymous Signatures with Applications to Decentralized Identity. In: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security. pp. 423–437. Association for Computing Machinery (2025). https://doi.org/10.1145/3719027.3744806
11. Davidson, A., Faz-Hernandez, A., Sullivan, N., Wood, C.A.: Oblivious Pseudorandom Functions (OPRFs) Using Prime-Order Groups. RFC 9497, RFC Editor (Dec 2023). https://doi.org/10.17487/RFC9497
12. Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R.S., Wood, C.A.: Hashing to Elliptic Curves. RFC 9380, RFC Editor (Aug 2023). https://doi.org/10.17487/RFC9380
13. Fdhila, W., Stifter, N., Kostal, K., Saglam, C., Sabadello, M.: Methods for decentralized identities: Evaluation and insights. In: International Conference on Business Process Management. pp. 119–135. Springer (2021)
14. Ford, B., Strauss, J.: An Offline Foundation for Online Accountable Pseudonyms. In: Proceedings of the 1st Workshop on Social Network Systems. pp. 31–36. SocialNets '08, Association for Computing Machinery (2008). https://doi.org/10.1145/1435497.1435503
15. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over Lagrange-Bases for Oecumenical Noninteractive Arguments of Knowledge. Cryptology ePrint Archive, Paper 2019/953 (2019)
16. Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., Schofnegger, M.: Poseidon: A New Hash Function for Zero-Knowledge Proof Systems. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 519–535. USENIX Association (2021)
17. Grassi, L., Khovratovich, D., Schofnegger, M.: Poseidon2: A Faster Version of the Poseidon Hash Function. Cryptology ePrint Archive, Paper 2023/323 (2023)
18. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In: Advances in Cryptology – CRYPTO '95. Lecture Notes in Computer Science, vol. 963, pp. 339–352. Springer Berlin Heidelberg (1995). https://doi.org/10.1007/3-540-44750-4_27
19. Homoliak, I., Li, Z., Szalachowski, P.: Bbb-voting: Self-tallying end-to-end verifiable 1-out-of-k blockchain-based boardroom voting. In: 2023 IEEE International Conference on Blockchain (Blockchain). pp. 297–306. IEEE (2023)
20. Homoliak, I., Venugopalan, S., Reijsbergen, D., Hum, Q., Schumi, R., Szalachowski, P.: The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. IEEE Communications Surveys & Tutorials **23**(1), 341–390 (2020)
21. Jarecki, S., Kiayias, A., Krawczyk, H., Xu, J.: TOPPSS: Cost-Minimal Password-Protected Secret Sharing Based on Threshold OPRF. In: Applied Cryptography and Network Security. pp. 39–58. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-61204-1_3
22. Kales, D., Walch, R.: A Nullifier Protocol Based on a Verifiable, Threshold OPRF. Tech. rep., TACEO (nd), https://github.com/TaceoLabs/oprf-service, file `docs/oprf.pdf` at revision df4de8e
23. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: Advances in Cryptology – ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 177–194. Springer Berlin Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_11

24. Košt'ál, K., Bencel, R., Ries, M., Kotuliak, I.: Blockchain e-voting done right: Privacy and transparency with public blockchain. In: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). pp. 592–595. IEEE (2019)
25. Kovalenko, D., Morháč, D., Koštál, K.: Zero-knowledge–enabled verification of nft-issued skill certificates. In: 2026 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (2026). https://doi.org/10.1109/ICBC67748.2026.11575535
26. Least Authority TFA GmbH: OPRF Noir Circuits: Security Audit Report. Final audit report, Least Authority TFA GmbH (Jan 2026), https://leastauthority.com/wp-content/uploads/2026/03/Least-Authority-TACEO-OPRF-Noir-Circuits-Final-Audit-Report.pdf
27. Longley, D., Sporny, M., Herman, I.: Verifiable Credential Data Integrity 1.0. W3c recommendation, World Wide Web Consortium (May 2025)
28. Maram, D., Malvai, H., Zhang, F., Jean-Louis, N., Frolov, A., Kell, T., Lobban, T., Moy, C., Juels, A., Miller, A.: CanDID: Can-Do Decentralized Identity with Legacy Compatibility, Sybil-Resistance, and Accountability. In: 2021 IEEE Symposium on Security and Privacy. pp. 1348–1366 (2021). https://doi.org/10.1109/SP40001.2021.00038
29. Mariani, J., Homoliak, I.: Proof-of-Social-Capital: A Consensus Protocol Replacing Stake for Social Capital (2026). https://doi.org/10.48550/arXiv.2505.12144
30. Mariani, J., Homoliak, I.: Social Capital Consensus for Blockchains: Exploring Design Options for Replacement of Proof-of-Stake by Social Influence. In: 2026 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–5 (2026). https://doi.org/10.1109/ICBC67748.2026.11575468
31. Mayrhofer, R., Lehmann, A., abhi shelat: A Brief Note on Cryptographic Pseudonyms for Anonymous Credentials (2025). https://doi.org/10.48550/arXiv.2510.05419
32. Mukherjee, S., Ravi, S., Schmitt, P., Raghavan, B.: SoK: The Ghost Trilemma (2024). https://doi.org/10.48550/arXiv.2308.02202
33. Noir Team: Noir v0.36.0. Official GitHub release (Oct 2024), commit 801c718
34. Rosenberg, M., White, J., Garman, C., Miers, I.: zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure. In: 2023 IEEE Symposium on Security and Privacy. pp. 790–808 (2023). https://doi.org/10.1109/SP46215.2023.10179430
35. Self: Self Protocol Architecture. https://docs.self.xyz/technical-docs/architecture (nd), accessed 2026-07-30
36. Semaphore Community: What Is Semaphore? https://docs.semaphore.pse.dev/V4/ (nd), version 4; accessed 2026-07-30
37. Sonnino, A., Al-Bassam, M., Bano, S., Meiklejohn, S., Danezis, G.: Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers. In: 26th Annual Network and Distributed System Security Symposium (NDSS). The Internet Society (2019). https://doi.org/10.14722/ndss.2019.23272
38. Sporny, M., Longley, D., Chadwick, D., Herman, I.: Verifiable Credentials Data Model v2.0. W3c recommendation, World Wide Web Consortium (May 2025)
39. Stančíková, I., Homoliak, I.: Sbvote: Scalable self-tallying blockchain-based voting. In: Proceedings of the 38th ACM/SIGAPP symposium on applied computing. pp. 203–211 (2023)
40. Tools for Humanity: World ID 4.0 Protocol Specifications. https://github.com/worldcoin/world-id-protocol (2025), directory `docs/world-id-4-specs`
41. WhiteHat, B., Bellés, M., Baylina, J.: ERC-2494: Baby Jubjub Elliptic Curve. ERC 2494, Ethereum Improvement Proposals (Jan 2020)