BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

Privacy-Preserving Proof of Uniqueness for Decentralized Systems

Adam Vozda*

Abstract
Decentralized systems often need scarce human participation: one vote, one allocation, one reputation profile, or one unit of social capital. This paper presents a Proof-of-Uniqueness prototype, motivated by Proof-of-Social-Capital (PoSC) consensus but designed as a reusable identity layer, that combines W3C Verifiable Credentials, client-side zero-knowledge proofs, a threshold OPRF network, and an Ethereum smart contract registry to enforce one enrollment per identity without publishing personal data. The implementation proves credential ownership, binds the OPRF query to the credential-derived private identifier, and stores only an OPRF-derived nullifier on-chain. Benchmarks show fast preprocessing, succinct proofs, stable registry gas, and two remaining bottlenecks: multi-second Barretenberg proving and multi-million-gas L1 verifier execution.

*xvozdaa00@vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Many decentralized applications need human uniqueness: one ballot, allocation, reputation profile, or unit of social capital per person. The same requirement appears in Proof-of-Social-Capital (PoSC), a consensus model in which network influence is backed by social capital rather than by electricity or financial stake [1]. PoSC is a strong motivating case because passive social capital must be assigned once per human: if a participant can cheaply create many identities, they can endorse themselves repeatedly and turn the consensus mechanism into a Sybil amplifier.

The problem addressed in this work is therefore not general identity verification, nor is it specific only to PoSC. The goal is narrower and reusable: given a credential issued after real-world identity verification, a user should be able to register exactly once in a decentralized system, while the public blockchain learns neither the credential fields nor a reusable identifier derived directly from those fields. A proper solution must provide deduplication, credential authentication, revocation and expiry handling, privacy against public observers, and a practical path to client-side use on ordinary devices.

Existing proof-of-personhood systems make different trade-offs. Social-graph approaches reduce dependence on governments or biometrics but expose relationship structures and are difficult to bootstrap securely. Biometric approaches can provide strong uniqueness but introduce specialized hardware, sensitive biological data, and centralization concerns. Government-backed credentials can provide strong real-world assurance; however, legacy ePassports and eID certificates are not ideal for zero-knowledge circuits because they often rely on RSA signatures or low-entropy certificate fields. This work takes a pragmatic middle path: it assumes a trusted issuer capable of producing a W3C Verifiable Credential (VC) [2], represented with decentralized identifier semantics where appropriate [3], and focuses on building the cryptographic layer that turns such a credential into private, on-chain uniqueness.

The contribution is an end-to-end Proof-of-Uniqueness (PoU) prototype. It uses BabyJubJub EdDSA and Poseidon commitments inside Noir circuits, Barretenberg proofs for browser-side proving, TACEO-style threshold OPRF evaluation for unlinkable nullifier derivation [5], and a Solidity registry that accepts enrollments only when proof, issuer, expiry, OPRF key, and duplicate checks all succeed. The resulting registry can serve different decentralized protocols that need a privacy-preserving one-human-one-entry constraint.

The design is intentionally explicit about its assumptions: the issuer must verify people honestly, the OPRF network must satisfy threshold non-collusion assumptions, and the client device must protect the holder's private key.

2. Protocol Design

The protocol separates identity issuance, private nullifier derivation, and on-chain enforcement. The issuer first creates a VC following the W3C data model [2]. The credential contains semantic identity fields, a holder BabyJubJub public key, a validity interval, and a Data Integrity proof. Rather than signing raw JSON inside the circuit, the implementation maps credential fields into field elements, hashes labeled leaves with Poseidon, reconstructs a 16-leaf Merkle root, and verifies the issuer's BabyJubJub EdDSA signature over that root. This keeps the circuit arithmetic-friendly and avoids expensive non-native primitives.

Inside the authorization circuit, the private uniqueness seed is computed as a Poseidon hash over selected credential attributes:

$$\texttt{HashID} = \mathit{Poseidon}(\texttt{subjectId}, \texttt{name}, \texttt{dob}, \texttt{placeOfBirth}, \texttt{sex}, \texttt{nationality}, \texttt{validFrom})$$

This value is never published. Publishing it directly would be simpler, but it would also create a long-lived secret-derived identifier: if the input ever leaked or became guessable, previous and future actions could be linked. Instead, the system obtains the public deduplication value through an Oblivious Pseudorandom Function (OPRF), a primitive standardized for prime-order groups in RFC 9497 [4].

The OPRF query is protected by a separate authorization proof. The browser proves that the blinded query was derived from the same hidden `HashID` that came from a valid issuer-signed VC. The public outputs of this proof are only the blinded query coordinates and the holder public key. The OPRF node then requires a fresh holder signature over the request identifier and blinded query. This live signature is important: possession of a copied VC file is not enough to obtain the OPRF response unless the attacker also controls the holder key.

After threshold OPRF nodes evaluate the blinded query, the client verifies the discrete-log equality proof, unblinds the result, and obtains a deterministic nullifier. The final enrollment proof binds together the verified credential, OPRF transcript, trusted OPRF public key, holder and issuer keys, expiry, and nullifier. The smart contract receives only the proof and needed public signals in a fixed order.

The `IdentityRegistry` contract is the public enforcement layer. It checks field bounds, requires the VC ownership OPRF key identifier, anchors the enrollment to the owner-managed trusted OPRF public key, verifies the Noir/Barretenberg proof, rejects duplicate nullifiers, rejects expired credentials, and accepts only whitelisted issuer keys. Successful enrollment stores a compact `IdentityRecord` in a mapping keyed by the nullifier. The same contract also supports holder-key revocation: the user proves a signature over the nullifier and a recent blockhash, and the contract deletes the record only if the proof public key matches the stored holder key. Finally, permissionless purging removes records that are expired or tied to issuers that were later removed from the trusted set.

3. Implementation and Evaluation

The prototype spans four toolchains. TypeScript and React orchestrate the browser flow, VC processing, OPRF client calls, proof generation, wallet interaction, and contract submission. Rust implements the OPRF node and the `vc-ownership` authorization module. Noir defines the three circuits: blinded-query authorization, VC+OPRF enrollment, and revocation. Solidity implements the registry and connects to generated Ultra verifier contracts. Noir was selected because it provides a high-level language for zero-knowledge programs and defaults to Barretenberg as a proving backend [6, 7].

The most important evaluation question is whether the design is practical enough to use. Micro-benchmarks show that the ordinary cryptographic preprocessing is not the bottleneck: Poseidon hashing over the 9-field `HashID` preimage averages **0.173 ms**, BabyJubJub EdDSA signing averages **13.47 ms**, and VC preprocessing with 13 labeled leaves and a Merkle root averages **1.24 ms**. The cost is dominated by proving. The `vc_blinded_query_auth_proof` circuit, used to authorize OPRF access before the enrollment transcript is fetched, has **59,338 ACIR opcodes** and **61,794 gates**; it averages **593.99 ms** for witness generation, **9.91 s** for proof generation, and **6.99 s** for local verification. The final `vc_oprf_enrollment_proof`, which binds the verified VC to the verified OPRF transcript and produces the 10 public enrollment signals accepted by the contract, is the heaviest client-side proof: it has **121,335 ACIR opcodes** and **120,673 gates**, averaging **1,407.44 ms** witness generation, **18.52 s** proof generation, and **14.01 s** local verifica-

tion. Both proofs remain succinct at **2,144 bytes**; the authorization proof exposes **4 public inputs**, while the enrollment proof exposes **10 public inputs** matching the on-chain signal layout. These timings are not instant, but they show that first-time enrollment is feasible for occasional identity lifecycle operations.

Gas profiling confirms that the registry logic is predictable. `IdentityRegistry` deploys for **1,510,960 gas** with a **6,345-byte** runtime. With mock verifiers, enrollment averages **243,245 gas**, revocation **42,168 gas**, and permissionless purge **53,166 gas**. A 1,000-enrollment scaling test remains stable after the first write: enrollment #1 costs **245,707 gas**, while #100 and #1000 both cost **210,760 gas**. Thus, mapping-based deduplication behaves as intended; the dominant L1 cost is proof verification itself, with generated Ultra verifier execution reaching **5,604,895 gas** in an end-to-end malformed-proof rejection test.

4. Security and Limitations

The security argument follows the system boundaries. The issuer is trusted to perform the real-world check before signing the credential; zero-knowledge can prove that a VC was signed, but it cannot prove that the issuer behaved honestly. The OPRF network is trusted for availability and threshold non-collusion. The smart contract is treated as the public, auditable policy engine. The client device remains the most practical weak point: if malware extracts both the VC material and the holder private key, software-only non-transferability is lost.

Within those assumptions, the prototype gives the desired uniqueness property. A valid enrollment requires a credential signed by a trusted issuer, live holder-key authorization, an OPRF transcript bound to the hidden credential-derived `HashID`, and a final proof accepted by the on-chain verifier. Reusing the same effective identity under the same trusted OPRF key produces the same nullifier and is rejected. Public observers see only issuer and holder public keys, expiry metadata, OPRF metadata, and the nullifier; they do not see raw VC fields or the `HashID` preimage.

The design still has three major limitations. First, issuer governance is centralized in the current prototype. A production deployment should move trusted-issuer management to a stronger governance process, for example a DAO or legally anchored trust framework. Second, browser-side proving creates noticeable latency. Better Wasm execution, mobile proving acceleration, or lighter proof systems would improve usability. Third, direct Layer-1 state is not a global-scale storage solution. The mapping architecture is simple and stable, but hundreds of millions of records would still create unacceptable state growth. A Layer-2 zk-rollup is the most natural next deployment target.

The broader outlook is promising. The European Digital Identity Wallet architecture already discusses W3C credential formats and high-assurance wallet cryptographic devices [8]. If standardized, high-entropy, privacy-preserving credentials become widely available, the ideal issuer assumption used in this prototype can be replaced by real public infrastructure. The main result of this work is therefore a practical bridge: it shows how a legally or institutionally issued credential can become a private, one-person-one-enrollment primitive suitable for Sybil-resistant decentralized systems, including PoSC-style consensus, governance, allocation, and reputation.

Acknowledgements

I would like to thank the supervisor of this work and the Faculty of Information Technology at Brno University of Technology for their guidance and support.

References

[1] J. Mariani and I. Homoliak. Proof-of-Social-Capital: A Consensus Protocol Replacing Stake for Social Capital. *arXiv:2505.12144*, 2025. https://arxiv.org/abs/2505.12144

[2] I. Herman, M. Jones, M. Sporny, T. Thibodeau Jr., and G. Cohen. Verifiable Credentials Data Model v2.0. W3C Recommendation, 15 May 2025. https://www.w3.org/TR/vc-data-model-2.0/

[3] W3C DID Working Group. Decentralized Identifiers (DIDs) v1.0. W3C Recommendation, 19 July 2022. https://www.w3.org/TR/did-core/

[4] A. Davidson, A. Faz-Hernandez, N. Sullivan, C. A. Wood, and C. Zaverucha. Oblivious Pseudorandom Functions (OPRFs) Using Prime-Order Groups. RFC 9497, IETF, 2023. https://datatracker.ietf.org/doc/html/rfc9497

[5] TACEO. Introducing TACEO:OPRF. https://core.taceo.io/articles/taceo-oprf/

[6] Noir. Noir Documentation. https://noir-lang.org/docs

[7] Aztec. Barretenberg Documentation. https://barretenberg.aztec.network/docs/

[8] European Commission. European Digital Identity Wallet Architecture and Reference Framework. https://eudi.dev/latest/architecture-and-reference-framework-main/