Poster: Privacy-Preserving Compliance Checks on Ethereum via Selective Disclosure

Supriya Khadka
Coventry University
khadkas25@uni.coventry.ac.uk

Dhiman Goswami
George Mason University
dgoswam@gmu.edu

Sanchari Das
George Mason University
sdas35@gmu.edu

***Abstract*—Digital identity verification often forces a privacy trade-off, where users must disclose sensitive personal data to prove simple eligibility criteria. As blockchain applications integrate with regulated environments, this over-disclosure creates significant risks of data breaches and surveillance. This work proposes a general Selective Disclosure Framework built on Ethereum, designed to decouple attribute verification from identity revelation. By utilizing client-side zk-SNARKs, the framework enables users to prove specific eligibility predicates without revealing underlying identity documents. We present a case study, *ZK-Compliance*, which implements a functional *Grant, Verify, Revoke* lifecycle for age verification. Preliminary results indicate that strict compliance requirements can be satisfied with negligible client-side latency (< 200 ms) while preserving the pseudonymous nature of public blockchains.**

I. Motivation

Public blockchains operate on the principle of radical transparency to ensure trustless verification of value transfer [1]. While this architecture ensures data integrity and auditability, it creates a hostile environment for Personally Identifiable Information (PII) [2]. As the blockchain ecosystem expands into regulated sectors, the conflict between this transparency and the need for user privacy has become a critical bottleneck.

The fundamental issue is that data placed on a public blockchain is immutable and globally visible. Although networks like Bitcoin provide a degree of pseudonymity, research has demonstrated that simple clustering heuristics can easily link on-chain behavior to real-world identities [3], [4], [5]. This issue is intensified as decentralized applications (dApps) integrate with regulated financial markets. To comply with Know Your Customer (KYC) and Anti-Money Laundering (AML) regulations, protocols often require users to verify their identities [6]. Under current paradigms, this creates a *transparency trap*: participation in a compliant economy results in the persistent linkage of a verified real-world identity with an immutable transaction history [2].

To address these concerns, the industry currently relies on centralized verification models where users upload documents to a Trusted Third Party (TTP) [7]. However, this reintroduces the centralization risks that blockchain technology aims to eliminate [8]. Centralized points of failure are prime targets for attacks, and history includes numerous instances of custodians suffering catastrophic breaches [1]. Furthermore, reliance on intermediaries grants third parties the power to censor users, failing to provide a truly sovereign solution [2].

In response to these limitations, this work contributes a **Selective Disclosure Framework** designed to address the structural tension between regulatory compliance and user privacy on public blockchains. By utilizing Zero-Knowledge Proofs (ZKPs), specifically zk-SNARKs, our framework enables attributes to be verified on-chain without exposing the underlying identity data. This approach effectively decouples identity storage from verification logic, mitigating the risks of both the *transparency trap* and centralized data breaches.

II. System Architecture

The proposed framework is designed around a *client-side proving* model, ensuring that raw identity data never leaves the user's device. The system coordinates interactions between the User (Prover), a trusted Issuer (e.g., a government agency), and a Service dApp (Verifier) through a set of on-chain and off-chain components.

A. Core Components

The architecture consists of four primary components:

- *Identity Vault (Client/Browser)*: A local secure storage mechanism that holds the user's raw attributes (e.g., birthdate) in an encrypted format, ensuring PII remains strictly non-custodial.
- *ZK Prover (SnarkJS)*: A client-side module running inside the browser that generates proofs using the user's private inputs and a cryptographic salt.
- *Verifier Contract*: An auto-generated Solidity contract that validates the cryptographic proof on-chain without viewing the inputs.
- *Access Registry*: A *state manager* contract that records valid access grants and manages the revocation lifecycle.

B. The Execution Lifecycle

The framework coordinates interactions through a structured *Grant, Verify, Revoke* lifecycle, as shown in Figure 1, that transforms private identity attributes into a public, revocable access grant:

Phase 1: Grant (Proof Generation): The lifecycle begins when a Verifier requests eligibility (e.g., $Age > 18$) for a specific duration. The user's client retrieves the private attribute from the Identity Vault, mixes it with a cryptographic salt to prevent rainbow table attacks, and inputs it into the ZK Prover. The client generates a zk-SNARK proof locally and

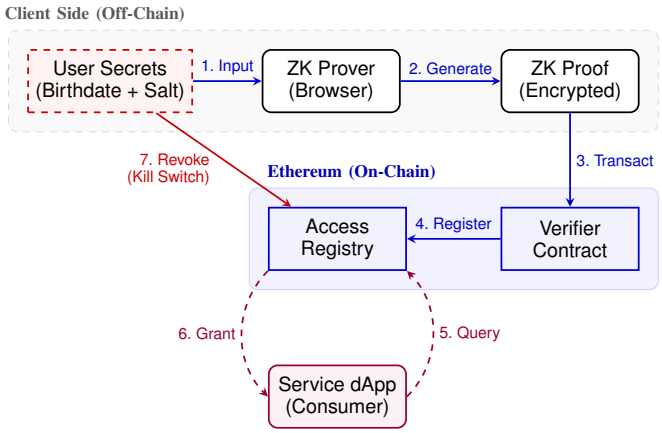

Fig. 1. **"Grant, Verify, Revoke"** lifecycle. (1-4) User grants access via a ZK proof. (5-6) dApp consumes the access. (7) User can revoke permission at any time.

submits a `grantAccess` transaction to the Access Registry. The Registry automatically validates the cryptographic proof on-chain; if valid, it mints an `AccessRecord` representing the user's consent.

Phase 2: Verify (Access Consumption): Once granted, verification becomes a lightweight, constant-time lookup. The Service dApp queries the Registry's `validateAccess` function. Instead of processing heavy cryptography, the dApp simply verifies that a valid, non-expired `AccessRecord` exists for the user. This decouples the computational cost of zero-knowledge proofs from the frequency of user sessions, allowing for repeated verification without repeated transactions.

Phase 3: Revoke (Sovereign Control) Unlike static credentials, this architecture provides a stateful kill switch. The user can trigger the `revokeAccess` transaction at any time to delete their `AccessRecord`. This instantly invalidates the dApp's ability to verify eligibility, restoring the user's privacy boundary immediately and preventing indefinite data retention.

III. Preliminary Results: ZK-Compliance

To validate the feasibility of this architecture, we developed *ZK-Compliance*, a proof-of-concept implementation focused on Age Verification.

A. Implementation Stack

Written in Circom, the circuit performs a range check ($Year - BirthYear \geq Threshold$) and squares the random salt to enforce entropy. We utilized SnarkJS to enable proof generation within a standard web browser. Smart contracts were deployed on the Ethereum Sepolia testnet, utilizing the BN128 elliptic curve for efficient on-chain pairing checks.

B. Performance & Cost Analysis

We evaluated the system across two critical dimensions: user latency and economic viability. On standard consumer hardware, the browser-based prover generates valid proofs in $<$ 200 ms. This indicates that client-side zk-SNARKs introduce negligible friction to the user experience.

Verification on Ethereum Mainnet, however, is cost-prohibitive: the on-chain Groth16 pairing check consumes $\approx 240,512$ gas, costing $\approx \$15.00$ per session (assuming 20 gwei at $\$3,000$ ETH). In contrast, deploying the Registry on Layer 2 scaling solutions (e.g., Arbitrum) reduces the cost to $< \$0.50$, making the system economically viable for high-volume applications.

IV. Conclusion and Future Work

This work demonstrates that regulatory compliance on public blockchains does not require the *transparency trap* of over-disclosure. By using client-side zk-SNARKs, our Selective Disclosure Framework allows users to satisfy strict eligibility predicates while maintaining pseudonymity and sovereign control. The current browser-based prototype provides early evidence that this architecture is capable enough for consumer applications, while restoring the user's ability to *Grant, Verify, and Revoke* access without permanent data retention.

In the future, to mitigate the risk of mempool front-running, we will update the circuit design to cryptographically bind proofs to the user's `msg.sender` address. We will also address the risk of deanonymization via cross-application clustering by implementing dApp-scoped nullifiers, ensuring that reusable identities do not create a global correlation graph. Finally, acknowledging the security risks of standard web environments, we aim to investigate isolating proving logic within secure enclaves and exploring off-chain revocation signaling to ensure the on-chain *Grant* history does not leave a persistent inferential trail.

Acknowledgment

We would like to thank the Data Agency and Security (DAS) Lab at George Mason University and Coventry University where the study was conducted. The opinions expressed in this work are solely those of the authors.

References

[1] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE communications surveys & tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[2] V. Buterin, "Privacy on the blockchain," https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain, Jan. 2016, ethereum Foundation Blog.

[3] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International conference on financial cryptography and data security*. Springer, 2013, pp. 6–24.

[4] D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the bitcoin blockchain: an analysis of the full users graph," in *2016 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 2016, pp. 537–546.

[5] N. Agarwal, E. Mackin, F. Tazi, M. Grover, R. More, and S. Das, "Systematic literature review of vulnerabilities and defenses in vpns, tor, and web browsers," in *International Conference on Information Systems Security*. Springer, 2025, pp. 357–375.

[6] Financial Action Task Force (FATF), "Guidance for a risk-based approach to virtual assets and virtual asset service providers," 2019.

[7] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *Acm Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[8] C. M. Bruhner, D. Hasselquist, and N. Carlsson, "Bridging the privacy gap: Enhanced user consent mechanisms on the web," in *Proc. NDSS Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb@ NDSS)*, 2023.

Privacy-Preserving Compliance Checks on Ethereum via Selective Disclosure

Supriya Khadka[1], Dhiman Goswami[2], Sanchari Das[2]

[1] Coventry University, Coventry, United Kingdom

[2] George Mason University, Fairfax, Virginia, United States

Abstract

Digital identity verification often forces a privacy trade-off, where users must disclose sensitive personal data to prove simple eligibility criteria. This work proposes a **Selective Disclosure Framework** on Ethereum designed to decouple attribute verification from identity revelation. By utilizing **client-side zk-SNARKs**, the framework enables users to prove specific eligibility predicates (e.g., Age > 18) without revealing underlying identity documents. We present a case study, **ZK-Compliance**, which implements a functional *Grant, Verify, Revoke* lifecycle. Preliminary results indicate that strict compliance requirements can be satisfied with negligible client-side latency (< 200 ms) while preserving the pseudonymous nature of public blockchains.

Introduction

Conflict: Public blockchains require radical transparency, creating hostile environment for (PII) [1,2].

Trap: Current KYC paradigms force users to link real-world identities to immutable transaction histories, destroying pseudonymity [3-5].

Solution: Selective Disclosure Framework using client-side zk-SNARKs [6] to decouple attribute verification (e.g., Age > 18) from identity revelation.

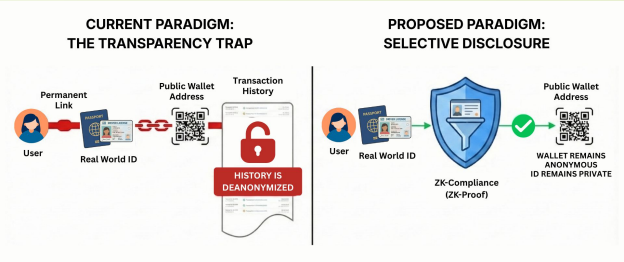

Figure 1: The Transparency Trap vs. Privacy-Preserving Selective Disclosure.

System Architecture

Identity Vault (Client): Encrypted local storage for raw attributes

ZK Prover (Browser): A SnarkJS module that generates proofs locally using private inputs and a cryptographic salt.

Verifier Contract: On-chain logic that validates cryptographic proofs without viewing inputs.

Access Registry: A state manager that records valid access grants and handles revocation

The "Grant, Verify, Revoke" Lifecycle

Grant (Proof Generation): The user inputs an attribute and salt. The client generates a zk-SNARK proof (< 200 ms) and submits it to the Access Registry.

Verify (Access Consumption): The dApp queries the Registry. Verification is a constant-time lookup of the AccessRecord, avoiding heavy on-chain cryptography.

Revoke (Sovereign Control): The user can trigger a "Kill Switch" transaction at any time, instantly deleting the AccessRecord and restoring privacy boundaries.

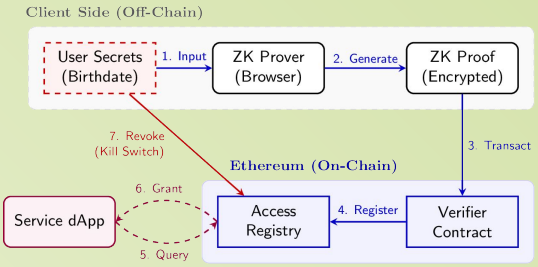

Figure 2: The **Grant, Verify, Revoke** lifecycle. (1-4) The user grants access via a ZK proof. (5-6) The dApp consumes the access. (7) The user can revoke permission at any time.

Case Study: ZK-Compliance and Results

Implementation Stack

- **Circuit**: Circom (Range Check + Salt Hashing)
- **Prover**: SnarkJS (Browser-based)
- **Network**: Ethereum Sepolia Testnet (BN128 Curve)

Performance Metrics

- **User Experience**: Client-side proof generation takes < 200 ms on consumer hardware.
- **Gas Costs**: Compared the cost of the on-chain Groth16 pairing check (~240,512 gas) across different environments:

Deployment	Cost (Est.)	Status
Ethereum Mainnet	~$15.00	Infeasible
Layer 2 (e.g.,Arbitrum)	**<$0.50**	**Viable**

Conclusion and Future Work

Conclusion

Regulatory compliance on public blockchains doesn't require the *transparency trap* of over-disclosure. By using client-side zk-SNARKs, the proposed framework allows users to satisfy strict eligibility predicates while maintaining pseudonymity and sovereign control.

Future Work

- Cryptographically binding proofs to the user's msg.sender address to prevent mempool theft.
- Implementing dApp-scoped *nullifiers* to prevent de-anonymization via cross-application clustering.
- Investigating the isolation of proving logic within secure enclaves.

References

[1] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," IEEE communications surveys & tutorials,vol. 20, no. 4, pp. 3416–3452, 2018.
[2] V. Buterin, "Privacy on the blockchain," https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain, Jan. 2016, ethereum Foundation Blog.
[3] Financial Action Task Force (FATF), "Guidance for a risk-based approach to virtual assets and virtual asset service providers," 2019.
[4] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in International conference on financial cryptography and data security. Springer, 2013, pp. 6–24.
[5] D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the bitcoin blockchain: an analysis of the full users graph," in 2016 IEEE international conference on data science and advanced analytics (DSAA). IEEE, 2016, pp. 537–546.
[6] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct {Non-Interactive} zero knowledge for a von neumann architecture," in 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 781–796