Identity-Private Healthcare Data Sharing on Blockchain via Zero-Knowledge Proofs and Account Abstraction

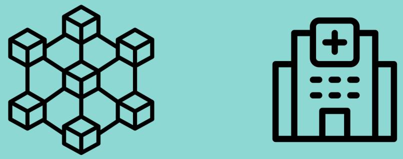
Kun Li, Dr. Ankur Lohachab, Dr. Visara Urovi
Department of Advanced Computing Sciences,
Maastricht University, The Netherlands

Maastricht University

BRAINS 2025

Why Identity Privacy Matters?

Current Practice



Blockchain ensures integrity & transparency

Problem

Transparency exposes identifier and metadata

Auditibility

Identity Privacy

We aim to achieve identity privacy without sacrificing auditability

What Happens When Identities Are Exposed?

Identity Linkage	Metadata Correlation	Behavioral Inference
Same user's addresses become linkable	Timing or gas patterns may imply user roles	Interaction traces imply medical conditions
Enables patient profiling	*Leads to re-identification*	*Causes discrimination or social stigma*

These issues undermine trust in blockchain-based healthcare systems

Research Question

How can blockchain-based healthcare data sharing preserve user identity privacy without compromising transparency and auditability?

- Blockchain transparency
- Public auditability

- Identity privacy

Building Blocks for Identity Privacy

Account Abstraction (AA)	Zero-Knowledge Proofs	Commitments
Smast Contract Accounts (SCAs) Replace EOAs Enables programmable identity logic	Prove correctness without revealing secrets Verify users privately	Commit now, prove later Link users and proofs privately

AA defines who can act; ZKPs prove validity; Commitments protect linkage – Together, they enable unlinkable yet verifiable identities

Privacy Gap in Existing Blockchain-based Healthcare Solutions

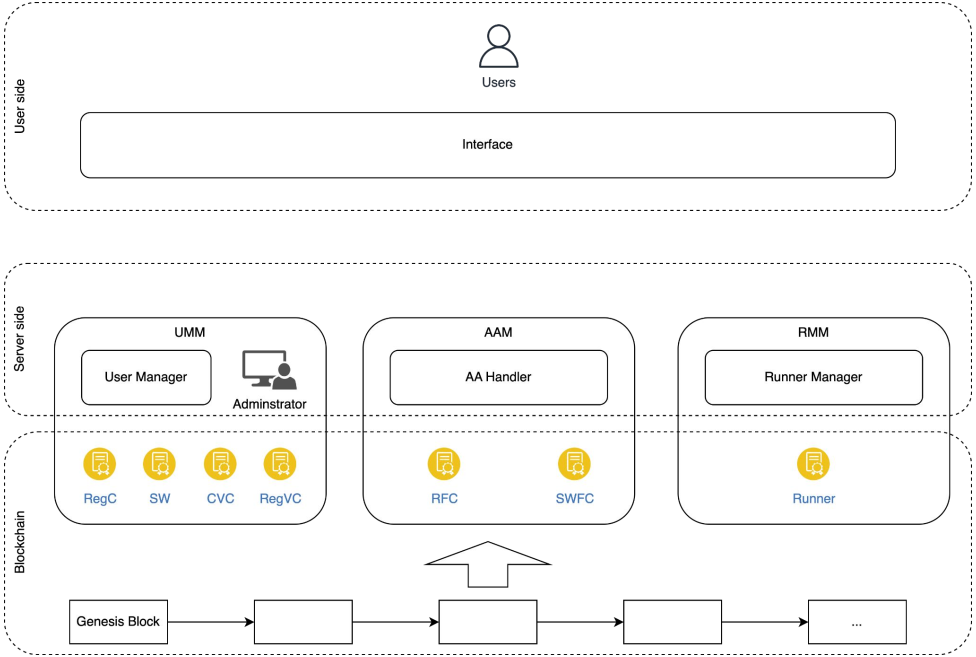
Most blockchain-based healthcare systems secure data content and enforce auditability - but identity privacy and behavioral unlinkability remain unsolved.

	Content Privacy	Identity Privacy	Auditability	Behavioral Unlinkability
CP-ABE/PRE*	✅	❌	✅	❌
ZK-based approaches	✅	❌	✅	❌
EOA-based Systems	❌	❌	✅	❌
Proposed Approach	✅	✅	✅	✅

*CP-ABE: Ciphertext-Policy Attribute-Based Encryption; PRE: Proxy Re-Encryption

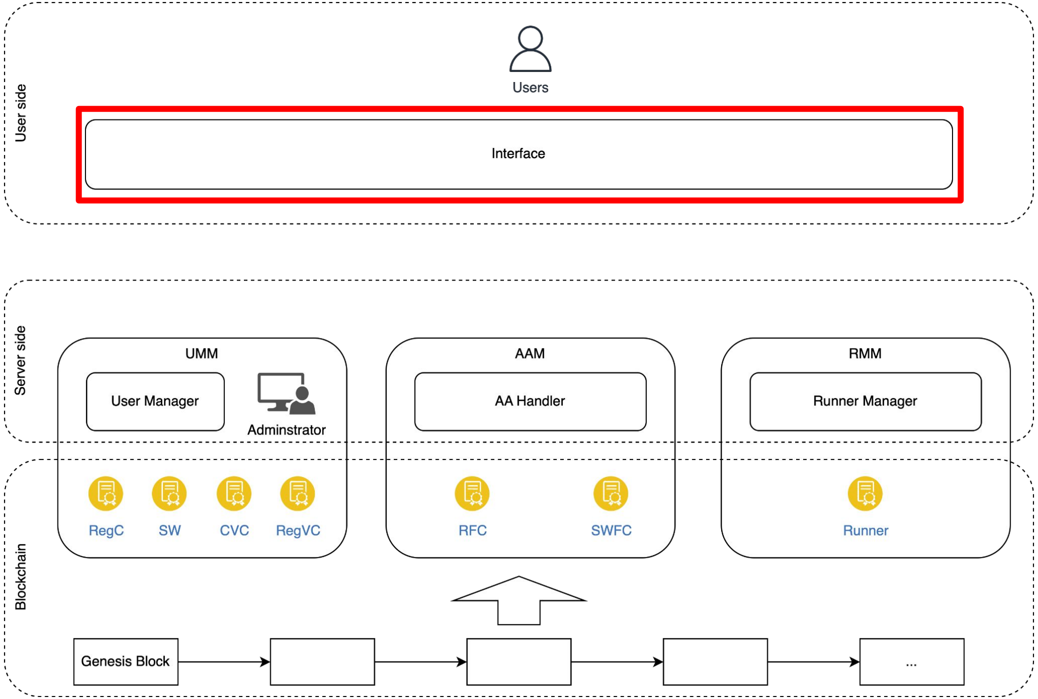
Proposed Approach Overview

Our approach integrates AA, ZKP and Pedersen Commitments to achieve identity privacy in healthcare data sharing.



Interface

- **Function**: Handles user input, commitment generation, and ZK proof generation
- **Role**: a bridge between user and blockchain
- **Interaction**: Sends commitments and ZK proofs to User Management Module; submits UserOp to Account Abstraction Module



User Management Module (UMM)

Function:

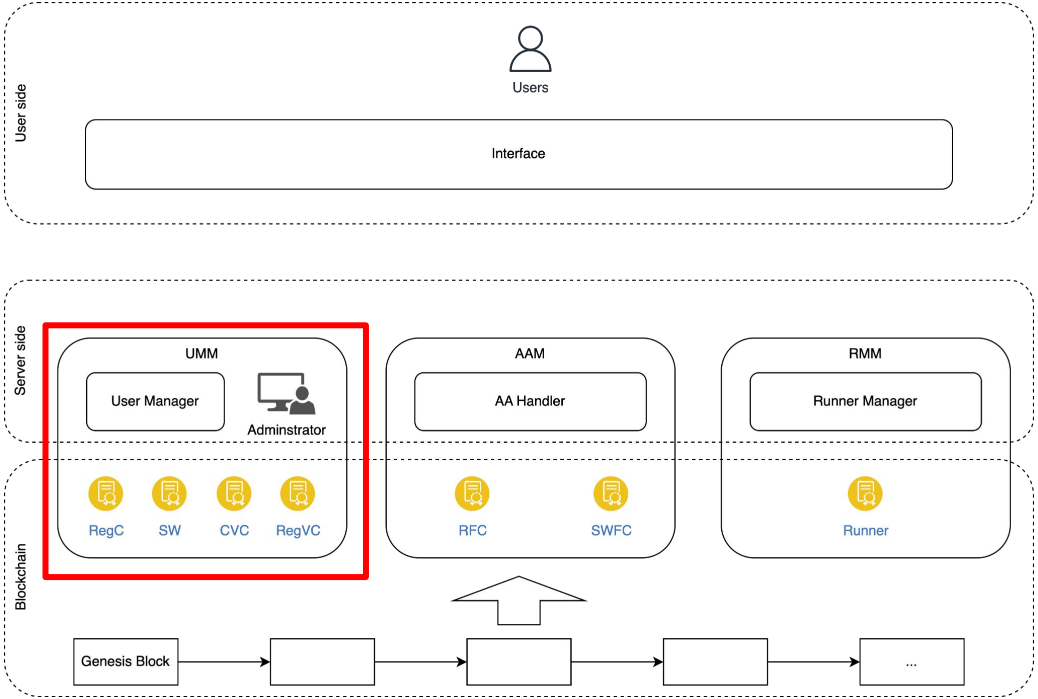
- Manages user registration and proof verification on-chain
- Stores registration commitments and verifies ZK proofs

Role:

- Serves as the trust anchor in the system - records valid users without exposing identities

Interaction:

- Receives commitments and ZK proofs from the Interface
- Prepares verified user references to AAM and RMM for execution



Account Abstraction Module (AAM)

Function:

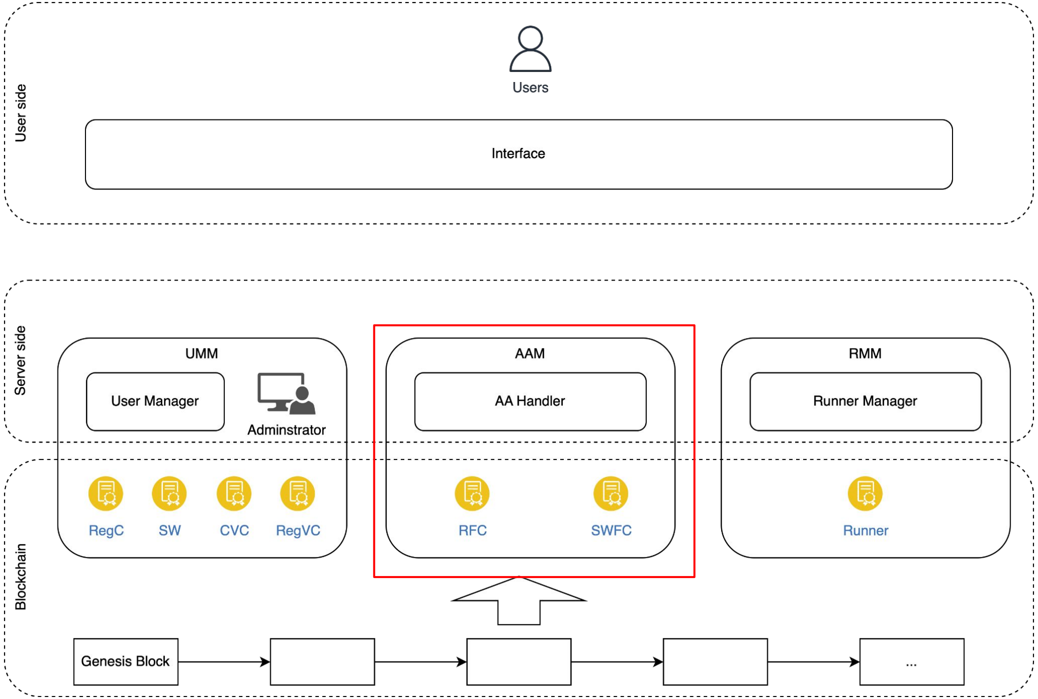
- Create and initializes SCAs and Runner
- Handles UserOp within the AA framework

Role:

- Acts as the execution layer of the system
- Provides unlinkable identities by decoupling user logic from EOAs

Interaction:

- Deploys and initializes user SCAs based on commitments
- Receives verified user references from the UMM
- Interacts with the RMM to execute UserOps



Runner Management Module (RMM)

Function:

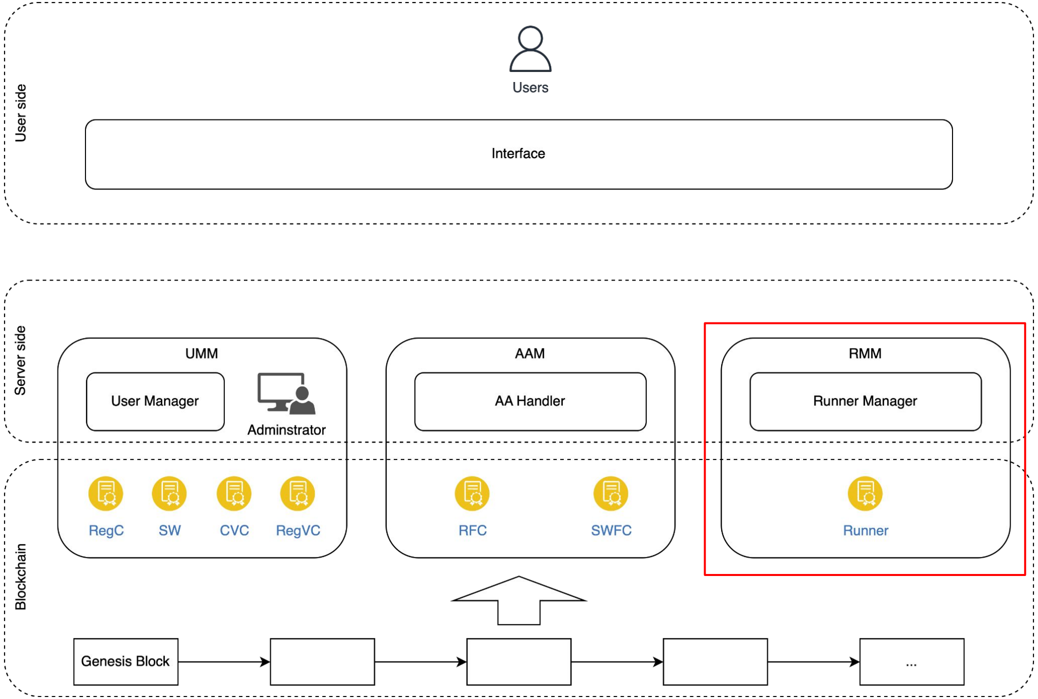
- Manage execution of UserOps via the Runner Contract

Role:

- Decouples user actions from observable on-chain identities

Interaction:

- Receives verified user operations from the AAM
- Executes these operations on-chain



Registration, Verification and Execution

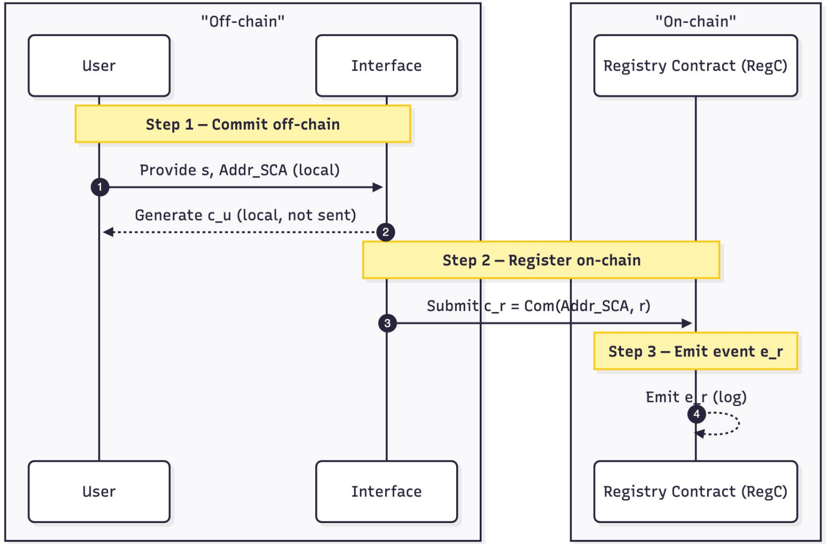
Registration	Verification	Execution
User commits $c_u = Com(s, r)$ and $c_r = Com(Addr_{SCA}, r)$;	User proves c_r exists via ZK prof (Merkle membership)	Runner execute User Operation (UserOp) via AA framework.
Only $Addr_{SCA}$ appears on-chain	No identifier revealed	User SCA acts unlinkably

Throughout the process, identity is verified but never revealed

Registration – Commit Without Revealing Identity

1. **Commit off-chain**
   User generates c_u=*Com(s, r)* and create a SCA with c_u
2. **Register on-chain**
   User sends c_r = *Com(Addr*$_{SCA}$*, r)* to Registry Contract
3. **Emit event**
   Registry confirms c_r and emits e_r

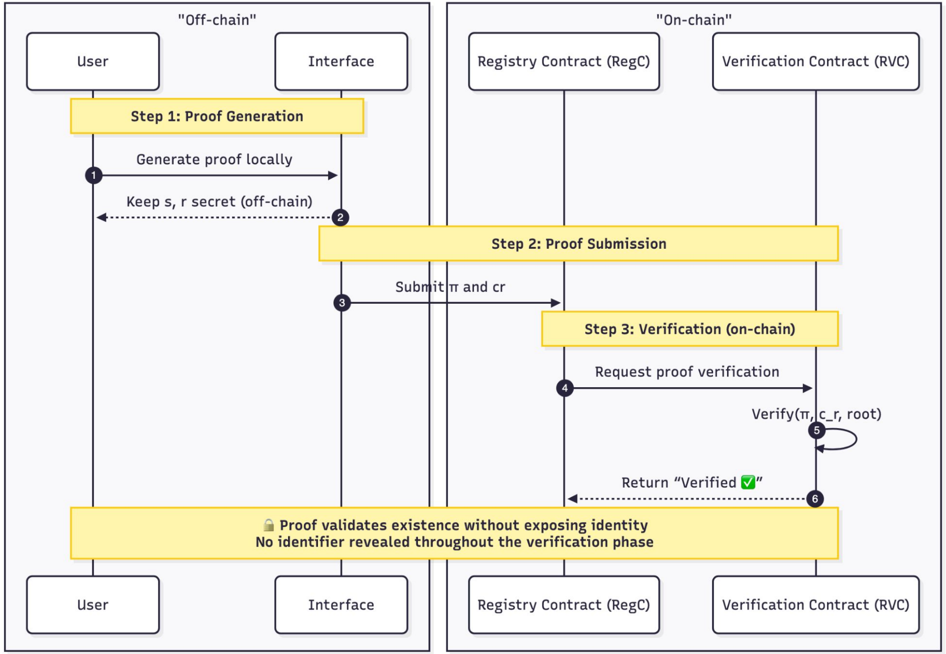
Only *Addr*$_{SCA}$ appears on chain



Verification – Proof of Membership without Revealing Identity

1. **Proof Generation**
   User generates zero-knowledge proof p_{mm} locally using Merkle path. *(s, r)* remain private off-chain.
2. **Proof Submission**
   User submits p_{mm} and c_r to Registry Contract
3. **Verification**
   RVC verifies proof correctness and confirms membership

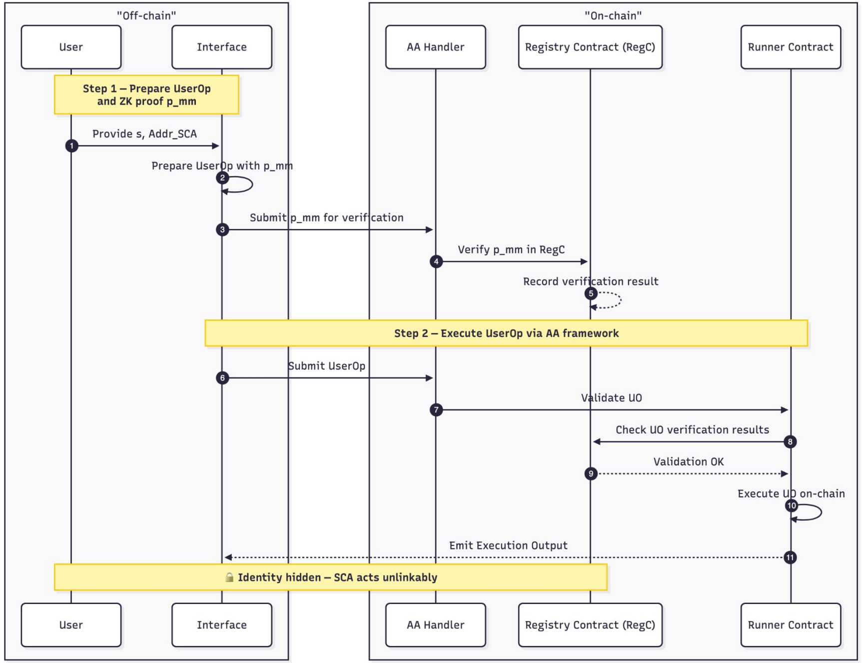
No identifier or secret is ever revealed.



Execution – Unlinkable Operation within AA Framework

1. **UserOp Preparation**
   User constructs UserOp and attaches ZK proof p_{mm}. All secrets *(s, r)* stay off-chain.
2. **Verification & Validation**
   p_{mm} is checked by UMM (RegC) through AA handler. Runner queries verification state.
3. **Execution**
   Runner executes UserOp on-chain on behalf of user's SCA. Identity remains unlinkable throughout.

Identity unobservable



Evaluation Scenarios

Experiment Setup

- All experiments were conducted on the Polygon Amoy Testnet
- Smart contracts include Registry (RegC), Runner (RC), and Verifier (VC, exported),
- Implemented using Solidity 0.8.27, Hardhat 2.26.3, Circom 2.0, and snarkjs 0.7.5.

Evaluation Scenarios

1. ERC-4337 vs. Proposed Approach
   Attacker knows full 〈user, SCA〉 mapping.

2. Account Shuffling vs. Proposed Approach
   Attacker knows one SCA per user and infers others via clustering.

Attacker Model

We adopt the taxonomy of Wagner & Eckhoff* and assume a **global**, **passive**, **external attacker** with bounded computation.

🌍Global
- Full visibility of ledger (transactions, state, events etc.)

👀Passive
- Only observes; no injection / modification / blocking

🏠External
- Not registered; holds no secrets; cannot produce ZK witnesses

⏰Static
- Use fixed inference heuristics (non-adaptive)

Prior knowledge

ERC-4337 setting
- Attacker knows <user, SCA>mapping
- Knows total number of users

Account shuffling setting
- Attacker knows one SCA per user & total user count
- Must infer extra addresses via clustering

*Wagner, Isabel, and David Eckhoff. "Technical privacy metrics: a systematic survey." *ACM Computing Surveys (Csur)* 51.3 (2018): 1-38.

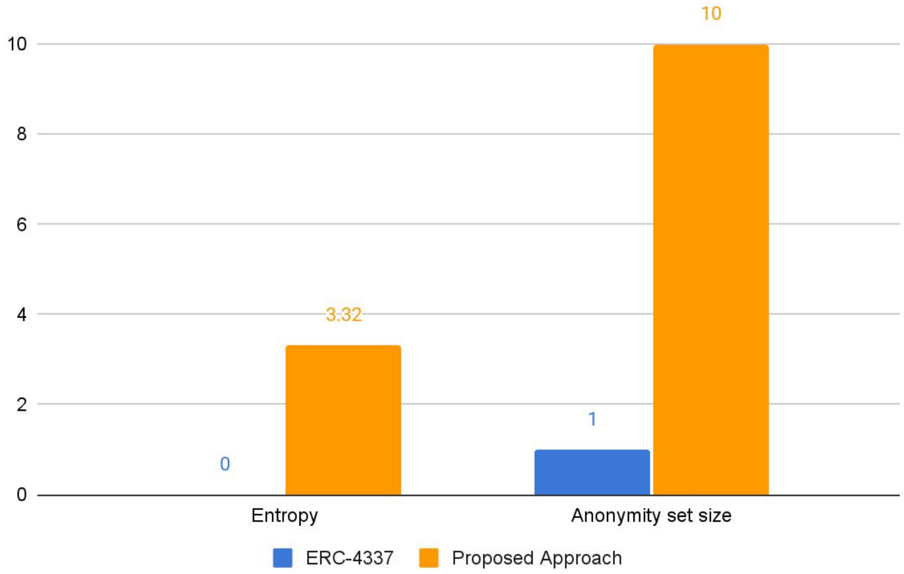
Evaluation Metrics

01	Entropy	• Measures identity uncertainty – How unpredictable user-address mapping is for the attacker • **Higher = Better privacy**
02	Anonymity Set Size	• Measures how many users are indistinguishable from each other • **Higher = Better privacy**
03	Adjusted Rand Index (ARI)	• Measures clustering accuracy between predicted and true user identities • **Lower = Better privacy**
04	Normalized Mutual Information (NMI)	• Measures information overlap between attacker's inference and ground truth • **Lower = Better privacy**

Identity Privacy Improvements over ERC-4337

- Higher Entropy: Attacker prediction becomes almost indistinguishable from random guessing
- Larger Anonymity Set: Each user is hiding among more indistinguishable users, making re-identification harder

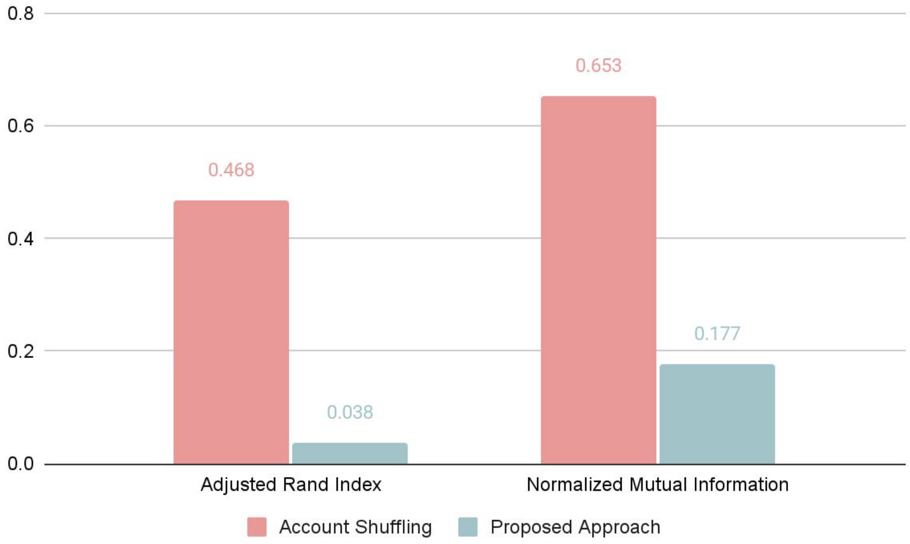
Metrics collectively show a clear privacy advantage over ERC-4337



Identity Privacy Improvement over Account Shuffling

- Attacker inference: Clustering via privileged operation heuristic
- Lower ARI and NMI: attacker cannot reliably cluster user addresses, making identity inference ineffective

Clustering attacks fail, proving unlinkability beyond shuffling-based defenses.



*Privileged operation heuristic: If an account performs a privileged operation, on a particular smart contract, it is assumed to be controlled by the owner of that contract

Discussion

Achievements

- Identity Unlinkability
- Verifiable Execution
- Quantitative Privacy Gains

Limitations

- Trusted setup (Groth 16)
- No dynamic credential management
- Limited evaluation (gas/scalability).
- Simplified attacker model.

Future Works

- Trying transparent proof (e.g., Plonk, Halo2).
- Add credential revocation/delegation
- Evaluate scalability and gas costs
- Map to GDPR/HIPAA compliance

Conclusion: Toward Identity-Private Blockchain Healthcare

1. We built a privacy-preserving healthcare data sharing system that protects *user identities* on blockchain.
2. Our approach integrates Account Abstraction, Zero-Knowledge Proofs, and Pedersen Commitments into a deployable framework.
3. It achieves unlinkability and verifiability.
4. Experiments show higher entropy and lower inference accuracy, confirming strong resistance to identity inference attacks.

Transparency and auditability can coexist with privacy

Q&A

Thank you for your attention

Kun Li: k.li@maastrichtuniversity.nl

Ankur Lohachab: ankur.lohachab@maastrichtuniversity.nl

Visara Urovi: v.urovi@maastrichtuniversity.nl