User privacy in DAOs

Final Advice

3620557

Ignas Apšega

byont

Fontys
University of Applied Sciences

Table of Contents

1. Introduction

1.1 Purpose of the document

The purpose of this document is to give final advice for the company Byont

1.2 Context

The context of this document is within the Implementation and management of Zero-Knowledge solutions to Byonts WEB3 template.

2. Advise

1. Privacy Concerns in DAOs
    - No explicit solutions for privacy in DAOs
    - Use Zero-Knowledge Proofs (ZKPs) for privacy preservation
2. Benefits of ZKPs for DAOs
    - Hide payment sums, recipients, rewards, and voting participants
    - Verify user data without revealing it
3. zk-SNARKs Protocol
    - Mature and widely-used protocol
    - Adaptable for identity verification
    - Recommended implementation: PLONK
4. Advanced Languages and Libraries: Circom and SnarkJS
    - Adopted in real-life production use cases
    - Limited documentation but active Telegram support
    - Circom simplifies zk-SNARKs problem statements
    - SnarkJS translates circuits, generates keys, and creates verifier smart contracts

5. Semaphore Protocol for Enhanced Privacy
    a. Designed for privacy-preserving applications
    b. Built on Zero-Knowledge Proofs (ZKPs) technology
    c. Suitable for DAOs and blockchain applications
6. Benefits of Semaphore Protocol
    d. Allows anonymous actions while maintaining accountability
    e. Scalable and efficient querying through The Graph
    f. Integrates with EVM-based smart contracts

Practical

General

- Familiarize with Semaphore protocol and its use cases
- Deploy smart contracts implementing Semaphore on the blockchain
- Index events using The Graph
- Develop a front-end using Next.js, React, and Chakra UI
- Integrate with a Web3 wallet provider like RainbowKit
- Test, monitor, and maintain the application

Wagmi hooks:

- Understand Wagmi library
- Use custom hooks to simplify code
- Be aware of blockchain limitations

The Graph:

- Design schema and mapping functions for your use case
- Create a subgraph manifest
- Test the subgraph thoroughly
- Utilize The Graph's decentralized architecture
- Query data efficiently across multiple smart contract
- Use log function to debug the issues in the mappings

Future use cases

- For DAO context: create a voting system with a whitelist of addresses allowed to create ballots.

- Whitelist typically includes larger stakeholders with significant influence on the DAO.
- Modify SemaphoreVoting smart contract: add whitelist, update "only coordinator" modifier for whitelisted addresses.
- The dApp can be also easily integrated with any EVM-based smart contract for various use cases.
- Implementing whitelist feature requires minimal changes to SemaphoreVoting smart contract.
- Update contract address code within custom hooks for seamless interaction with the updated smart contract.
- Deploy a new subgraph on The Graph with the updated smart contract address for efficient querying and data accessibility.

3. Useful resources

Thorough the internship a lot of useful hard-to-find resources were found. Here a list of it:

Bible of ZKP - It is a GitHub repo that is updated and maintained frequently, it consists of all the needed information about Zero-Knowledge Proofs - from mathematics to tooling.

ZkLearn - The best learning material regards beginners of Zero-Knowledge Proofs. Explains it on a high level. The material involves workshop videos with code and questions from students. Teaches what is zk-SNARKS, and how to use Circom and SnarkJS (wish I found this resource early in the internship).

The ZKP blog - a good starting point to try to understand mathematics and the inner workings of Zero-Knowledge Proofs.

zkREPL - Online tool to write and test Circom Circuits.