ZKU Assignment 2

SHIVAM SHARMA
shivam691999@gmail.com
Discord : shivam691999#8446

Question 1: Privacy & ZK VMs

1. **Explain in brief, how does the existing blockchain state transition to a new state? What is the advantage of using zk based verification over re-execution for state transition?**

Ans: The blockchain state works as a turing machine. The state of the blockchain remains static and does not change unless an external function is applied on it. The actions are applied in a sequential order and they change the state of the blockchain in that manner. The current way to sync a blockchain to the head in a full node is to get the rawBlockData and process and apply the blockdata on the state one by one. Using ZK based verification we can trust the data and instead of processing each block we can use the data as a snapshot.

2. **Explain in brief what is a ZK VM (virtual machine) and how it works? (Refer here)**
    a. **Give examples of certain projects building Zk VMs (at-least 2-3 projects). Describe in brief, key differences in their VMs.**
    b. ***[Bonus]* What are the advantages and disadvantages of some of the existing Zk VMs?**
    c. ***[Bonus]* Explain in detail one of the Zk VM architectures using diagrams.**

Ans : A) A ZK VM allows developers to relatively easily write programs whose execution can be verified while keeping all or some inputs private. Developers don't have to write a new circuit per application and handle the complexity of unique proving and verifying keys.

B) Advantages and Disadvantages of existing VMs are :

Advantages :

- Developer does not have to write a lower level code or create complex circuits that results in ease of development.
- Simpler proving and verification key management.
- Compatibility with other VMs.

Disadvantages :

Vendor Lock-in : This happens when a developer starts development on a particular VM but later wants to switch to another Vendor's VM. This could lead to problems and the developers will need to choose their VM very carefully.

Question 2. Semaphore

1. What is Semaphore? Explain in brief how it works? What applications can be developed using Semaphore (mention 3-4)?
2. Clone the semaphore repo (3bce72f).
    1. Run the tests and add a screenshot of all the test passing.
    2. Explain code in the sempahore.circom file (including public, private inputs).
    3. *[Bonus]* Create a frontend for the current semaphore version. You can use this as reference.

3. Use Elefria protocol on the Harmony Testnet, try to generate a ZK identity and authenticate yourself as a user.
    1. What potential challenges are there to overcome in such an authentication system?
    2. *[Bonus]* What potential improvements can one make to simplify the Elefria authentication protocol?

Answer : 1. Semaphore is a zero-knowledge gadget which allows Ethereum users to prove their membership of a set without revealing their original identity. At the same time, it allows users to signal their endorsement of an arbitrary string. It is designed to be a simple and generic privacy layer for Ethereum DApps. Use cases include private voting, whistleblowing, mixers, and anonymous authentication.

2.

3. The potential challenges to overcome in such authentication systems are :

- The user needs to store the nullifier keys and if lost he/she/they might lose their access forever.
- It can not easily handle multiple access control lists.

Question 3. Tornado Cash

You will need these resources:

- tornadocash/tornado-trees
- tornadocash/tornado-nova
- *[Bonus]* Lecture from Roman Semenov, Co-founder of Tornado Cash

1. Compare and contrast the circuits and contracts in the two repositories above (or consult this article), summarize the key improvements/upgrades from tornado-trees to tornado-nova in 100 words.
2. Check out the tornado-trees repo
    1. Take a look at the `circuits/TreeUpdateArgsHasher.circom` and `contracts/TornadoTrees.sol`. Explain the process to update the withdrawal tree (including public, private inputs to the circuit, arguments sent to the contract call, and the on-chain verification process).
    2. Why do you think we use the SHA256 hash here instead of the Poseidon hash used elsewhere?
3. Clone/fork the tornado-nova repo
    1. Run the tests and add a screenshot of all the tests passing.
    2. Add a script named `custom.test.js` under `test/` and write a test for all of the followings in a single `it` function
        1. estimate and print gas needed to insert a pair of leaves to `MerkleTreeWithHistory`
        2. deposit 0.08 ETH in L1
        3. withdraw 0.05 ETH in L2
        4. assert recipient, omniBridge, and tornadoPool balances are correct
4. *[Bonus]* Read Proposal #11 of Tornado.cash governance, what is the purpose of the newly deployed L1Unwrapper contract?

Ans3: 1. Users can now deposit and withdraw any amount of a given token within each pool. Nova will also provide the possibility to make shielded transfers of deposited tokens while staying within the pool. Users will be able to transfer a chosen amount of their deposited tokens (not necessarily all of them) to another address without needing to withdraw them from the pool.

2. 1.

2. Poseidon hash is around 30x slower than SHA256. Although Poseidon is more expensive than SHA256, it can be utilized here.

3. https://github.com/0xsharma/ZKU-Assignments/blob/main/Assignment-2/constom.test.js

Question 4. Thinking In ZK

1. If you have a chance to meet with the people who built Tornado Cash & Semaphore, what questions would you ask them about their protocols?
2. *[Bonus]* Regarding writing and maintaining circuits for each dapp separately, what are your thoughts about using just one circuit for all dapps? Is that even possible? What is likely to be a standard in the future for developing Zk dapps?

Ans: 1. The questions I would ask if I meet the magicians behind Tornado Cash and Semaphore are :
A. What inspired you to create a private transactions medium?
B. Will it be more efficient and safe than the Monero Blockchain Tech ?

C. Some people withdraw the same amount of tokens from mixers as they deposited. Do you have some points of wisdom for them ?

2. Only one circuit for all dapps could be achieved in production in the near future through ZK VMs. Polygon Miden is one of the few that are working on this technology. This could also result in writing better smart contracts that can run more OPCODES than what exist in EVM allowing for us to write more complex programs.