Zero Knowledge University March-April 2022 Cohort

Week 2 Assignment

Course Registration Email: zhang.jeremy.2001@gmail.com

Discord Username: xyz5368#1102

1. Privacy & ZK VMs
    a. The existing blockchain transitions to a new state by taking some input data and the current state of the blockchain, running the State Transition Function, and creating a new state. If most of the nodes agree on this new state, this new state is added to the blockchain. There are 4 advantages of using verification over re-execution:

    1. Privacy – Users do not need to submit their input data to a public network and make it public. As a result, the input and the STF are not public.

    2. Scalability – The time it takes to verify a transaction will decrease, which makes the processing capability of the network better and the throughput higher.

    3. Computation restriction – Doing complex computations on chain costs lots of gas. Verification can circumvent this limitation, which allows for more complex computation on chain.

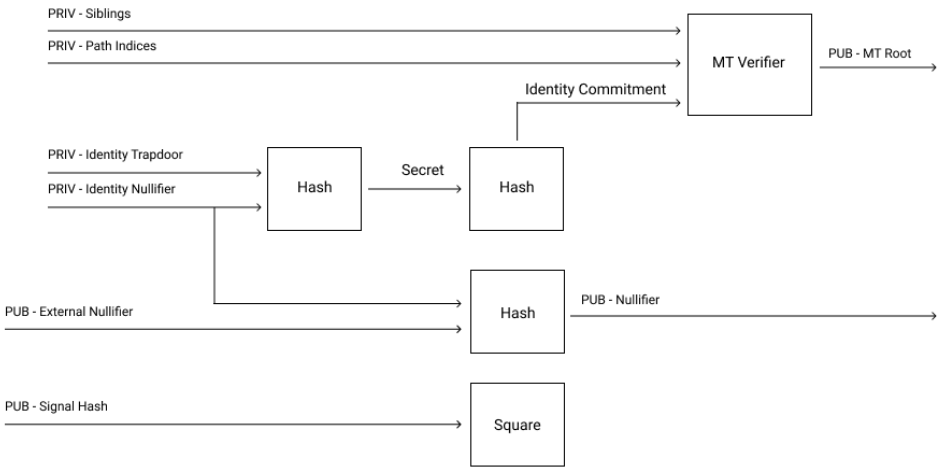
    4. Storage explosion – When doing re-execution, you have to store the state of the entire chain, which takes lots of space. When doing verification, you do not have to store all of the history, making it more storage efficient.

    b. A ZK VM is a circuit that executes bytecode. Basically, it allows a prover to show that given some input and the bytecode, they have correctly executed the program with the given inputs.
        i. Here are some current projects:
        Cairo – Turing complete language and VM, runs on StarkNet.
        Snarkvm – ZK-SNARK technology, runs the Leo Programming Language.
        Distaff – Proprietary program syntax, STARK-based proof of execution generated, zCloak uses this.
        ii. Bonus not attempted.
        iii. Bonus not attempted.
2. Semaphore
    a. Semaphore is a zk gadget that allows yourself to authenticate yourself and conduct some action (like voting, whistleblowing, etc.) while staying anonymous.
    b. Clone the repo

i.

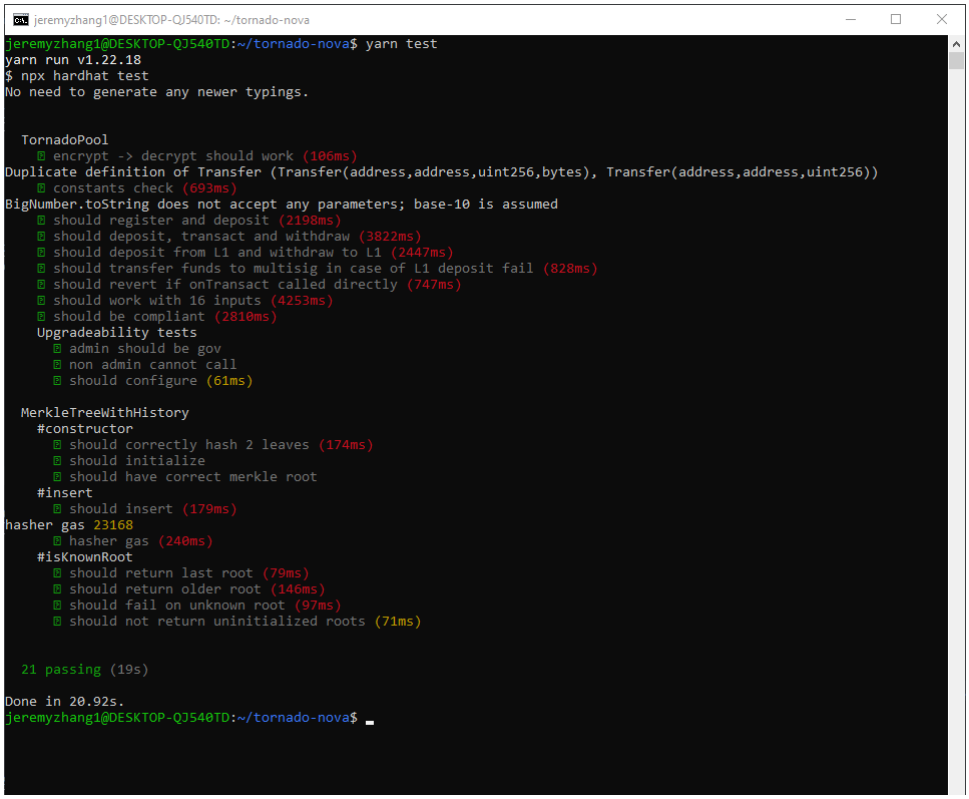
```
jeremyzhang1@DESKTOP-QJ540TD: /mnt/c/Users/Jeremy Zhang/Documents/Classes/zku/week2/semaphore
jeremyzhang1@DESKTOP-QJ540TD:/mnt/c/Users/Jeremy Zhang/Documents/Classes/zku/week2/semaphore$ yarn test
yarn run v1.22.18
$ hardhat test
Generating typings for: 2 artifacts in dir: ./build/typechain for target: ethers-v5
Successfully generated 5 typings!
Compiled 1 Solidity file successfully


  SemaphoreVoting
    # createPoll
      ✓ Should not create a poll with a wrong depth (87ms)
      ✓ Should not create a poll greater than the snark scalar field
      ✓ Should create a poll (566ms)
      ✓ Should not create a poll if it already exists
    # startPoll
      ✓ Should not start the poll if the caller is not the coordinator
      ✓ Should start the poll
      ✓ Should not start a poll if it has already been started
    # addVoter
      ✓ Should not add a voter if the caller is not the coordinator
      ✓ Should not add a voter if the poll has already been started
      ✓ Should add a voter to an existing poll (507ms)
      ✓ Should return the correct number of poll voters
    # castVote
      ✓ Should not cast a vote if the caller is not the coordinator
      ✓ Should not cast a vote if the poll is not ongoing
      ✓ Should not cast a vote if the proof is not valid (1622ms)
      ✓ Should cast a vote (618ms)
      ✓ Should not cast a vote twice
    # endPoll
      ✓ Should not end the poll if the caller is not the coordinator
      ✓ Should end the poll
      ✓ Should not end a poll if it has already been ended

  SemaphoreWhistleblowing
    # createEntity
      ✓ Should not create an entity with a wrong depth
      ✓ Should not create an entity greater than the snark scalar field
      ✓ Should create an entity (705ms)
      ✓ Should not create a entity if it already exists
    # addWhistleblower
      ✓ Should not add a whistleblower if the caller is not the editor
      ✓ Should add a whistleblower to an existing entity (462ms)
      ✓ Should return the correct number of whistleblowers of an entity
    # removeWhistleblower
      ✓ Should not remove a whistleblower if the caller is not the editor
      ✓ Should remove a whistleblower from an existing entity (853ms)
    # publishLeak
      ✓ Should not publish a leak if the caller is not the editor (61ms)
      ✓ Should not publish a leak if the proof is not valid (924ms)
      ✓ Should publish a leak (621ms)


  31 passing (36s)

Done in 93.70s.
jeremyzhang1@DESKTOP-QJ540TD:/mnt/c/Users/Jeremy Zhang/Documents/Classes/zku/week2/semaphore$
```

ii. Here is a diagram of how the circuits work from the semaphore repo:



iii. Bonus not attempted.

c. Use Elefria protocol on Harmony testnet
   i. From a user perspective, authentication takes a really long time. Normal authentication with Google takes perhaps a few seconds at most. This took much longer than that. Also, there are windows asking you to pay for gas fees, which the user might also not want to pay.
   ii. Bonus not attempted.

3. Tornado Cash
   a. Tornado pools were a fixed amount per deposit/withdrawal. Tornado Cash Nova allows users to withdraw and deposit arbitrary amounts inside of their tornado cash pools.
   b. Check out the tornado-trees repo
      i. When a user wants to withdraw, they construct a proof with circom circuits of their element being added to the Tornado pool, which is a merkle tree. Then, the smart contract is called and the proof is submitted onto the blockchain. When everything checks out, the user will have successfully added to the withdrawal tree.
      ii. The Poseidon hash is very expensive on chain, but cheap in circuits. Conversely, the sha256 hash function is cheap on chain, but expensive for circuits. Since we want to optimize for gas usage, we use the sha256 hash function.
   c. Clone/fork the tornado-nova repo
      i. 

      ii. Not attemped.
   d. Bonus not attempted.

4. Thinking in ZK

a. Will there be mobile app support for some of the applications you all have built? What are some of the challenges for doing such? I know that current mobile phones don't have enough processing power to generate witnesses/proofs, so how would we get around those bottlenecks as well?
b. Bonus not attempted.