ZKU Assignment-1

Shivam Sharma

EMAIL : shivam691999@gmail.com

GITHUB : https://github.com/0xsharma/ZKU-Assignments/tree/main/Assignment-1

DISCORD : shivam691999#8446

Q1)

1) merkletree.circom

```
pragma circom 2.0.0;
include "node_modules/circomlib/circuits/mimc.circom";

// Calculate next layer of the tree
template branch(height) {
 var items = 1 << height;
 // input values
 signal input vals[items * 2];
 // output values
 signal output outs[items];

 component hash[items];
 for(var i = 0; i < items; i++) {
   hash[i] = MultiMiMC7(2,91);
   hash[i].in[0] <== vals[i * 2];
   hash[i].in[1] <== vals[i * 2 + 1];
   hash[i].k <== 0;
   hash[i].out ==> outs[i];
 }
}

// merkle tree construction from values
template merkleProof(levels) {
 signal input leaves[1 << levels];
 signal output root;

 component layers[levels];
 for(var level = levels - 1; level >= 0; level--) {
```

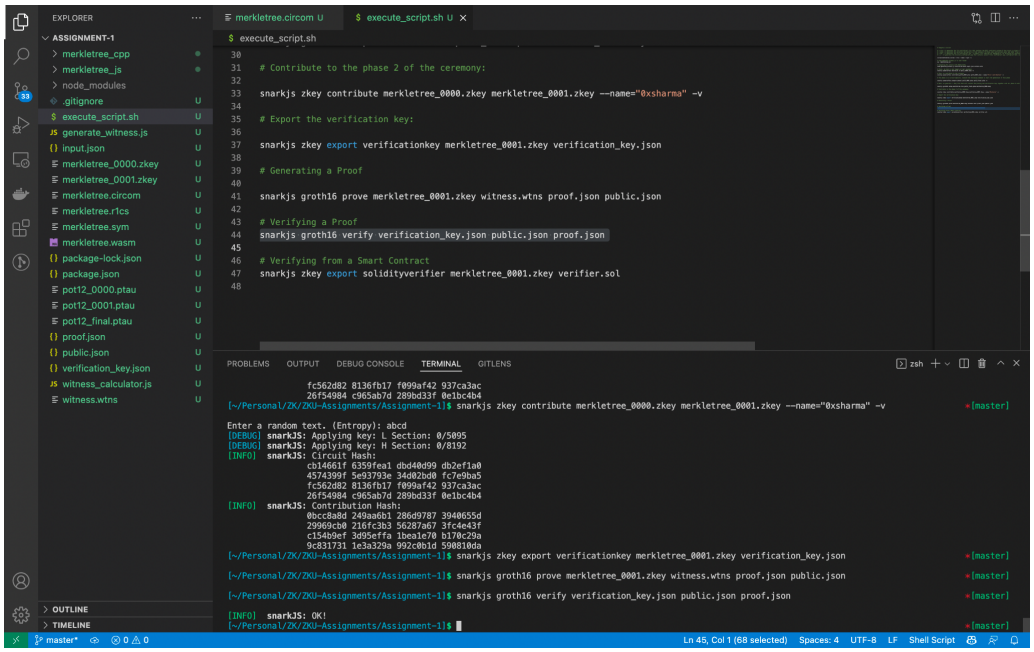
```
   layers[level] = branch(level);
   for(var i = 0; i < (1 << (level + 1)); i++) {
     layers[level].vals[i] <== level == levels - 1 ? leaves[i] : layers[level +
1].outs[i];
   }
 }
 root <== levels > 0 ? layers[0].outs[0] : leaves[0];
}

component main{public [leaves]} = merkleProof(3);
```

2)While using 8 inputs I faced the following error :
“circuit too big for this power of tau ceremony. 5096*2 > 2**12”

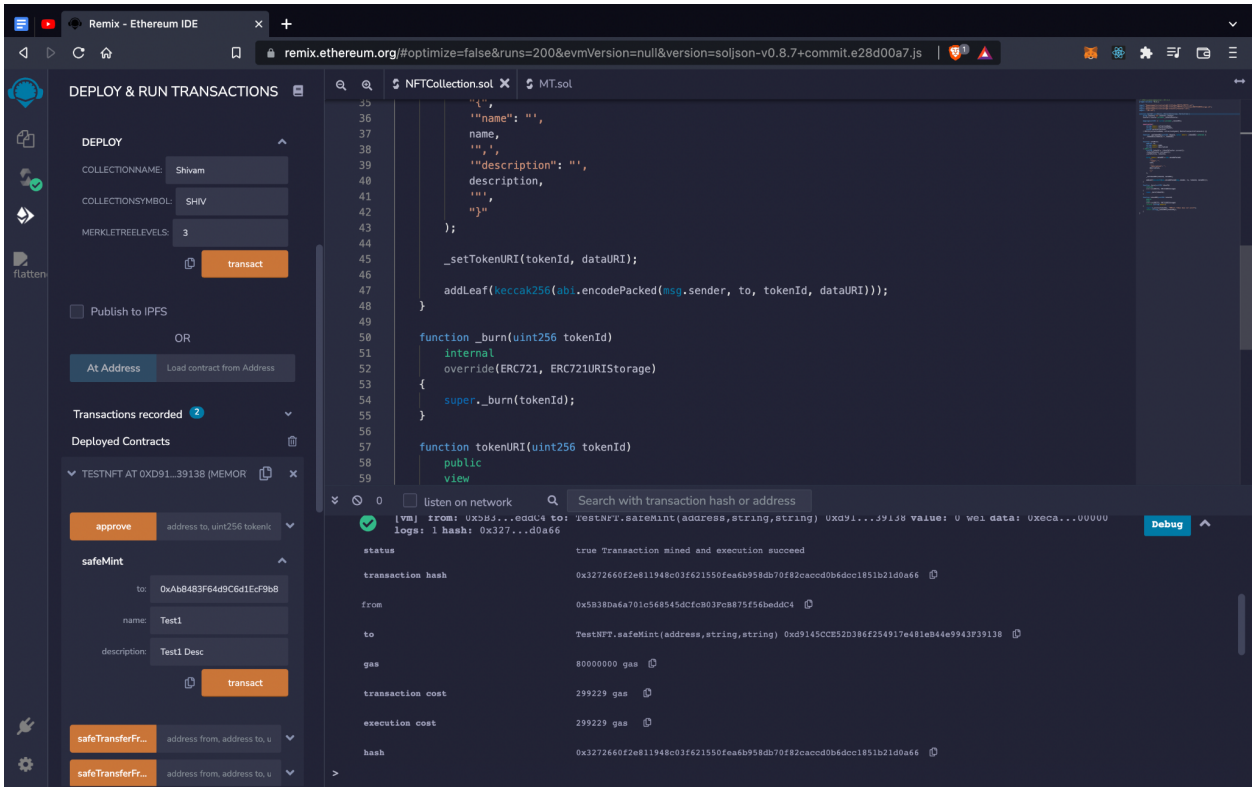
3) Yes, According to me we require ZK proofs to verify this. Yes, public verifiable smart contracts can be used to verify zk proof using the “generate call” methods. This technology is being used in various applications like Tornado Cash etc.

4) The script can be found here :
https://github.com/0xsharma/ZKU-Assignments/blob/main/Assignment-1/Question-1/execute_script.sh

OUTPUT :



Q2)
Output :



Q3)

A) Differences between SNARKs and STARKs

The proof size of SNARKs are much smaller than STARKs meaning it would take less storage on-chain. Therefore SNARKs are estimated to require much more gas than SNARKs and that makes STARKs economically expensive for end users in comparison to SNARKs.

zk-SNARK proofs are dependent on an initial trusted setup between a prover and verifier and this could create a potential centralization issue. They can be used in blockchain based applications much easily as they are smaller in size and can be verified on-chain.

zk-STARKs doesn't require a trusted setup but the con is that a prover with enough computational power could create fake proofs. They can be used in advanced p2p systems where there is no trusted setup (Binance p2p). They can also be used in confidential exchange of information between different countries.

B) Differences btw trusted setup process between groth16 and PLONK.

The setup process in groth16 has to be repeated for each circuit. For example, we need to create a unique setup for every different circuit.
In PLONK, **one trusted setup** can be used to validate all the circuits.

C) Give an idea of how we can apply ZK to create unique usage for NFTs.

ZK can be applied on NFT's in the following way:

- ZK can be used to check if you have any NFT of a particular collection without revealing the tokenID.
- It can also be used to hide certain attributes/metadata of an NFT.

D) Give a novel idea on how we can apply ZK for Dao Tooling.

ZK can be used in DAO in novel manner in the following way :

- It can be used in hiding the identity of a proposer and voter.
- ZK can help a Grants DAO to hide its abstract details.
- It can be used in making acquisitions/investments (in research based investment DAO's) while hiding the information on deals to provide front-run protection.