ZKU Week 4 Assignment

Email: qadir@xord.com
Discord Id: AbdulQadir#0432

Part 1 Theoretical Questions

Question One:

There are various ways in which people can sync to the network effectively. Some of them are as follows:

Simplified Payment Verification (SPV):

SPV was proposed in Nakamoto's whitepaper. As a result, we can verify the bitcoin payments by verifying the block header Merkle root without running the full node.

SPV allows users to synchronize the whole blockchain and verify the transactions without downloading the entire data. This can be done using the Merkle Inclusion Proof to verify the transactions. We need to pass the Merkle root along with the Merkle path elements to verify the transactions.

The Above system is for the POW consensus mechanism blockchain. For POS, we need to verify the set of validators' signatures.

Non-Interactive Proofs of Proof-of-Work:

NIPOPOWS allows the users to synchronize the blockchain by downloading a small sample of the block headers or superblocks. Superblocks are those blocks in which the miners have done most work.

NIPOPOWS is based on the block difficulty and allows blockchains to communicate and interoperate.

FlyClient:

FlyClient is a novel transaction verification light client for chains of variable difficulty. Fly Client solves the problem of fast synchronization by skipping the no. of blocks in between and downloading only a logarithmic number of block headers while storing only a single block header between executions.

Question Two:

In Plumo, a proposer proposes the block and a group/set of validators selected after a particular epoch to verify and sign the transaction. Plumo uses two mechanisms in his architecture to compress the four months of blocks in a single proof in a groth16 that proves in seconds.

- 2-chain Curves: Plumo uses two different curves, BLS12-377 and BW6. BLS12-377 scalar field is always equal to the BW6 base field; thus, scalar field - base field = 0; Plumo computes the Snarks proof on the BW6 and uses the BLS signature verification the BlS12-377.
- Plumo also uses the hybrid hash function to aggregate validator signature into a single signature. To do this, Plumo uses two hash functions, black2 and BH-Pederson hash. BH Pederson is used due to black2 hash being expensive. Intermediate value y of Pederson is used in black2.

Question Three:

1. BLS aggregation of the validators hash in the Plumo mechanism is already part of the harmony consensus mechanism. To use the Plumo in harmony, this change in EPOS will be needed.
2. Harmony also has the shards which cause compatibility problems while implementing the Plumo.

Part 2 Final Project

Question One:

There are couples of ideas that I have brainstormed last week, which are:

- ZK-Store: Buying things online without revealing the identity or address for payments
- Zk-Docs: Sending private documents on-chain without revealing them. But this will be a zero-knowledge-based project, not ZKP.

Question Two:

1. Implementation Difficulty:

   Zk-MultiSig > ZK-Tax ≈ ZK-Quiz

   In terms of implementation, I believe zk-MultiSig will be the difficult one due to the architecture design and usage of signatures; then, zk-tax and zk-quiz will have almost the same difficulty.

2. Potential User Base:

   Zk-MultiSig ≈ ZK-Tax < ZK-Quiz

   Zk-MultiSIg and ZK-Tax will have the same number of users due to complex implementation in the current system. Therefore, migrating users to pay tax using the ZK-Tax is hard.

3. Importance of ZK:

   Zk-MultiSig < ZK-Tax ≈ ZK-Quiz

   ZKP is essential in the Zk-Tax and ZK-Quiz system due to proving the salary user earns and in the first and second proving the correct answers without revealing the answers and removing the intermediate person

Question Three:

I'm more inclined towards the ZK-Tax (Tax on Salary - FBR) as this is the issue of revealing the salary; thus, people will know how much you earn. This can be implemented with the ZK-Payroll to generate the employees' payrolls and pay incoming salary tax to the FBR without revealing your identity and salary. We can also mint an NFT as a certificate of tax-paying, and the user can easily file the return on the tax paid using that NFT.

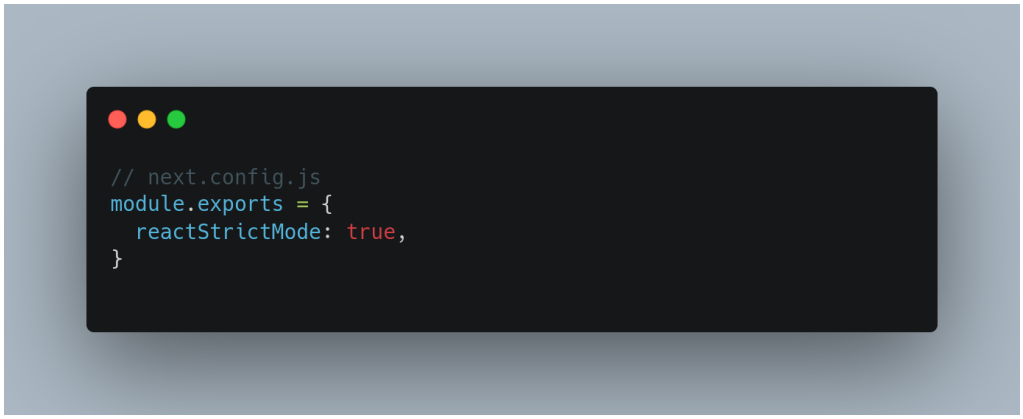
Part 3 Frontend Assignment

Question One:

webpack configurations in the NextJS config file contain multiple parts, which are:

- `reactStrictMode: true`

  React strict mode is used for highlighting potential problems in an application in the development mode. It helps to identify unsafe lifecycles, legacy API usage, and several other features.

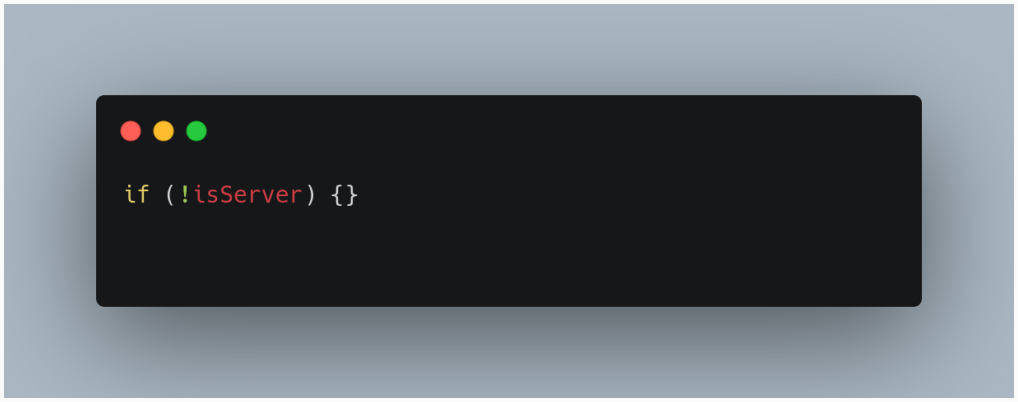


```
// next.config.js
module.exports = {
  reactStrictMode: true,
}
```

- The below lines of code are used to extend the usage of webpack. Config is used to access the webpack modules and other parts of the configs to extend.

```
module.exports = {
  webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {
    // Important: return the modified config
    return config
  },
}
```

- The isServer is used to check that the config is not called for the server. Nextjs config is called two times for server and frontend. IsServer is a boolean



- ProvidePlugin automatically loads modules instead of having to import or require them everywhere. So, for example, the **global** plugin will automatically be added to the code below.

- Resolve fallback is used to redirect module requests when normal resolving fails. In our condition, we have set the no. of modules to false, not to include their polyfills.

Question Two:

Code

Question Three:

Demo video