Middle East Technical University Department Of Cyber Security

CSEC 519

Blockchain and Cryptocurrency Technologies

Spring 2024-2025

Second Assignment

Name Surname: Alkım Doğan
Student ID: 2521482

1 Whitepaper Report

1.1 Introduction

- Today, mostly the central authority model is used for money transactions and banking systems.
- Central authority has some advantages but the most famous disadvantages is called **single point of failure**. This also means that users use the system based on trust. We mean the trust that the system will not fail and not do something unethical.
- Trust system also causes some costs that are not necessary if we use cash.
- The aim of Satoshi is to get rid of trust system by using distributed system that uses cryptographic proof rather than trust. Distributed system also means there is no need for third central party.
- Satoshi uses cryptography (hash) to make transaction irreversible. Because to reverse it, we need recomputation of cryptographic puzzle.
- Since there is a cryptographic puzzle (I know it is a hash from lectures), it will require some brute forcing mechanism. So, if anyone has more than 50% CPU power, then the possibility that he can solve the puzzle each time of block addition becomes relatively high. This means he can attack the system. For example, he can reverse his transactions and so on.

1.2 Transactions

- Transactions are hashed and signed repeatedly as users send coin to someone.
- Lets give an example. Assume Alice wants to send coin to Bob. Alice takes hash of the coin with the Bob's public key. Then, Alice signs this hash with her own sec256k1 private key. After Alice signed, the coin technically belongs to Bob. Other nodes can very this with taking hash of the coin with Bob's public key, then checking the sign with Alice's public key.
- The process above lacks checking double spending.

- Double spending is where Alice tries to send the same money to Bob and Carla. We somehow need to solve this.
- It is easy to prevent double spending if there exists a central authority. But we do not want it as a solution because of some problems mentioned earlier.

1.3 Timestamp Server

- The solution of the last problem includes a timestamp server.
- If I can take hash of some data, then that data must have existed beforehand. Otherwise, I would not be able to take hash.
- If I can publicly spread this hash with a timestamp, we can also say something about the time of the data.
- Finally, we chain the hashes as they are announced/spread publicly. This also supports that the last data approves the previous ones.

1.4 Proof-of-Work

- Publicly announcing the hashes includes proof of work. This is solving a hash puzzle. When we take hash of the data, we want resulting hash to be smaller than some target value. This is the same thing as starting with some specific number of 0's. Satoshi mentions SHA-256 algorithm for taking hash in whitepaper.
- Satoshi, in this section, says we solve it by using nonce value. But we also know from our lectures, changing merkle tree is also another option for solving hash puzzles.
- There could be a lot of system. They are mentioned in whitepaper. But our system is one-CPU-one-vote. (We can say it is not with today's technology but Satoshi says so).
- The longest chain is the valid chain. We should search for the longest. Others are invalid.
- This CPU issue along with longest valid chaing introduces 51% attack as well. This means majority of the CPU power must be controlled by good intentioned people.
- Mentioned difficulty (target value) is dynamic since we need some average block addition (puzzle solving) timing. Hashes produce randomized values. So, it is not easy to come up with a value starting with some specific number of zeros. We can also say more secure the hash, more difficult it is.

1.5 Network

- Satoshi mentions 6 steps to run a network in this section.
- There may be possibilities where two different nodes solve the hash puzzle at the same time. Therefore, nodes will receive one of the newly added blocks. There will be two branches with equal length. The moment one of the branches becomes longer, nodes switch to that branch.
- When one broadcasts the transaction, it does not have to be approved by each node in the network. If it reaches enough number of nodes, it will be fine. Distributed system solves message drop or lost problems.

1.6 Incentive

- Satoshi mentions coinbase transaction. This special transaction is an incentive for honest miner nodes. This is also good way to distribute the initial 21 million bitcoin.

- We also add transaction fees to the coinbase transaction. Transaction fee is equal to input−output for each transaction.

- Satoshi also mentions when 21 million of bitcoin is distributed, the incentive will only rely on transaction fees.

- Incentive makes more valuable to play the game with rules instead of being a hacker.

1.7 Reclaiming Disk Space

- Merkle Tree data structure is used to store transactions inside a block. This is a binary tree. The parent node is the hash of its children recursively defined from bottom to top.

- Good thing is that after some point we do not need to store entire transactions. For example, I can only store upper levels of Merkle Tree of Block number 23 to save my disk.

- He also calculates required RAM to store entire blockchain headers (Merkle Tree is not included). He uses Moore's Law to predict it.

- Moore's Law states that the number of transistors double each 2 years.

1.8 Simplified Payment Verification

- For payment verification, we do not need to store the entire Blockchain, which is a challenge if we consider this is more than 450 GB today. Alternatively, we can only store headers of the longest chain.

- For verification, we only need the specific block that included that blockchain. This is the reason why we do not need to store the entire blockchain with all data for verification purposes.

1.9 Combining and Splitting Value

- We cannot behave a unit coin independently. This would mean separate transaction for each unit. For instance, 0.01 is our unit. If I want to send 0.1 amount of money, It would require 10 transactions. Because 10 amount of unit would change the ownership. However, what I have mentioned is ridiculous and this is not the way.

- As a solution, one transaction consists of multiple inputs and outputs.

- Change is use to receive money back in case we do not want to spend the entire coin.

1.10 Privacy

- This section is mostly about pseudonymity.

- Briefly, transactions are public. Accounts are totally anonymous. We do not know which account belongs to whom unless someone reveals himself.

- If we do not want more privacy, we can make use of different public-private key for each transactions. This would mean someone cannot keep track of the specific account. But it is still not possible for multi input transactions.

1.11 Calculations

- We already mentioned about 51% attack. This section is about calculation of the probability.
- It can be understood that 51% does not necessarily mean an arbitrary changed blockchain. The attacker can only receive the money that he has spent so far by conducting 51% attack. The reason is we still have honest nodes that will not accept invalid transactions. By invalid, we mean double spent, unverified signed ones and so on.
- Satoshi makes use of some statistical problems to calculate the probability. Attacker will try to come from some specific number of blocks back from the longest chain. This number depends on attackers previously spent coins and their locations on the blockchain since attacker will try to undo those transactions.
- The race of hones and attacker nodes is Binomial Random Walk, whereas the attacker trying to catch the longest chain is similar to Gambler's Ruin problem. Finally, attacker's potential progress is a Poisson distribution.
- Satoshi calculates the probability of attacker catching up with a simple C code. The result is the fact that when honest nodes probability of finding the next block is higher than attacker's probability, the attacker's probability of catching up decreases as the number of blocks required the catch increases.

2 Satoshi's Mispredictions

- First, he only worries about CPU power. What about ASIC, FPGA and GPU?
- Mining has caused a lot of electricity consumption. The amount of electricity consumption is higher than most of the low population Europe countries. This is not very good for environment.
- There is still pseudonymity in Bitcoin. However, It may not be wanted always. For example, We mentioned in our lectures that sometimes we are required to reveal our accounts for sending some bitcoins for exchange.
- Pseudonymity is also not a very good thing for forensic people. Espacially we have to consider money laundering cases.
- He states that mining requires changing Nonce value. But we can also change the structure of the Merkle tree as well for mining purposes. In fact, Nonce is just a 32 bit integer. Brute forcing all the possibilities of the nonce value is very very easy. Therefore, with today's current computation power, it is preferred to change the transaciton locations in the Merkle tree and to change the coinbase transaction to solve the hash puzzle.
- Satoshi mentions SHA-256 for Hash puzzles. But we make use of a lot of hashes to prevent some ASIC, FPGA and GPU mining because we want it to be relatively fair system. Additionally, we do not want anyone to control the network by having more than 51% computation power by using the specific hardwares mentioned. For this purpose, X11, X13, X15, X16, X16r, X17 algorithms may be used in hash puzzles.
- People form mining pools. They share the price earned by mining within the pool. Satoshi did not predict it.