**Data Structures and Algorithms**

**Blockchain Structure and Block Mining**

Blockchain is a data structure used for holding transactions. It consists of blocks linked together. Each block contains four transactions. The main contents of a block are the following:

* Block id
* Block hash: Unique hash key (256 bits) of the block that distinguishes it as a primary key
* Previous block hash: For linking to the previous block, each block contains its hash key as a pointer.
* Timestamp: Time of creation of the block
* Merkle tree: It is the tree that holds the signatures of transactions. A signature is the unique key which identifies a transaction.

Blocks are created by the people who are so called miners. In CrypDist, all peers are miners, which means all of them are responsible for creation of the blocks. Creation of a block is a slow process such that it requires finding a valid hash key. The hash key is produced by using the block metadata (id, timestamp, previous hash, merkle tree root), by using the SHA-256 cryptographic hash function, such that when its data is changed, the change can be detected from its hash key. However, it should be ensured that after changing the data, the hash key will not be the same. For this purpose, there is a constraint on the hash key which is its first 8 bits are zero among the total 256 bits. For satisfying this constraint, an integer value called nonce is added to the block metadata to produce its hash key. For finding an appropriate nonce, a miner should try all values up to some maximum number in a brute-force approach. The miner who finds the nonce value first has the right to add the block to the chain. The block is also broadcasted to other peers.

There is also a difficulty value of a block which indicates the difficulty of its mining process. The difficulty of the blockchain is computed by summing up all the difficulties of the blocks on its longest path. Longest path means there are some forks in the chain. So the chain is more likely a tree instead of a linked list. A fork happens when two peers attempt to add a block at the same time. The consensus protocol ensures that the blockchain is the same for all peers by using the longest path.

The blockchain is said to be immutable, however the real explanation is changing it is very difficult. When data of a block is changed, its unique hash key also changes depending on the merkle tree root. So the later block’s hash key also changes since it depends on the previous block’s hash key. By domino effect, the last block’s hash key changes. So, the blockchain of a peer is compared with the blockchains of the majority, and when a change is detected, system requires him to update his blockchain by pulling from the other peers. There is still a possibility of an attacker to create an alternative blockchain, but according to the recent research, the probability of it decreases exponentially as the length of chain grows.

**Merkle Tree Structure**

Merkle tree is a data structure where its leaves hold the signatures of transactions. The signatures are produced by double hashing their string representations by SHA-256 hash function. The internal nodes are generated by combining the leaves’ signatures and by the same process, the root is generated in a bottom-up manner. The merkle tree root is used to detect a change in the transaction data such that it also changes the block hash.