Explain some of the ways hashing functions enable blockchain technology

Hash functions are one of the most extensively used cryptographic algorithms in blockchain technology. They are designed to protect data integrity. A hash algorithm is a mathematical function that transforms any input into a fixed size output. The following characteristics of the hash function enable blockchain technology:

1. Deterministic - it generates the same result for the particular input, otherwise it will be impossible to keep track of the input

2. Quick computation - it should return a hash of the input quickly, otherwise the system won't be efficient

3. Pre-Image resistance - it is infeasible to determine the original input from its hash value

4. Small changes in the input changes the hash - even small changes in input will result in different output, hash value. This property enables immutability of the blockchain.

5. Collision resistant - it is difficult to find two inputs that produce the same output.

6. Puzzle friendly - it solves computation task through generating a hash value out of the block and nonce (a random string) to compare with difficulty level. By completing this work, a new block is added into blockchain. This process (mining) is enabled in blockchains with a proof of work protocol.

So, hash functions are widely used in blockchain technology, which includes: addresses, digital signature, searching transaction, mining, hash rate and much more…

Briefly explain Bitcoin's UTXO model of transaction validation (separate from POW)

Bitcoin blockchain differs from traditional financial system: there is no user account and user balance. A private key is used to identify an owner of a Bitcoin wallet, a public key is generated out of private key and Bitcoin address is generated out of public key. To transfer a value between different Bitcoin addresses transactions are used with sender's digital signature.

If transaction is confirmed it's published on the blockchain. Each transaction contains a list of inputs and a list of outputs. Each input identifies a Bitcoin address that is providing the funds and an unspent transaction that address has received in the past.

Each output represents the Bitcoin address receiving the funds and the amount that address receives. Each output can be used as an input only once.

The difference between input and output is transaction fee, which will be earned by Bitcoin miner. A coinbase transaction is a special type of transaction, which creates new Bitcoin as a reward for the miner of a block. The coinbase transaction has no inputs and one or more outputs. The output of a coinbase transaction is a new UTXO. Bitcoin balance which is shown in our mobile wallet is generally the sum of unspent transaction output (UTXO) list.

What is the structure of a Block in bitcoin and how does it relate to the 'blockchain' (merkle tree vs merkle list of merkle trees)

A Bitcoin block records the data related to a Bitcoin transaction. The new blocks are mined one after another with all new transactions in the network recorded permanently. Bitcoin blockchain makes it very difficult to modify or delete the data that is registered

on a block. Bitcoin blockchain is created by linking individual block one after another. A block structure has several elements:

1. Block identifiers

2. Block header

3. Transactions

1. Block identifiers

The block identifiers are the elements that identify a block's address, its height and size. Here are the main block identifiers:

a. Hash: A hash is a unique identifier that distinguishes one block from the other. A block hash is generated by passing the block header metadata through the cryptographic algorithm SHA256 (Secure Hash Algorithm). The hashing result is also called block header hash. Each new block on the blockchain is connected to the previous block via the previous block's hashed address. Every new block mined on the chain becomes the child of the previous block, which is called a parent block. So, a child block carries the hash of its parent block in its block header, in addition to its own hash value. This way all blocks are connected to form a long chain. If the parent block is changed, the child block hash field also changes, which further changes the child's child hash field and so on.

b. Block height: The first-ever Bitcoin block was created in January 2009 and is called "Genesis Block". As it was the first block, it was assigned a zero height. The height of a block is the number of blocks that have been mined between genesis block and current block.

2. Block header

The Bitcoin block header contains all information about the block. The length of the header is 80 bytes and it consists of the following metadata:

1. 4-bytes Version
2. 4-bytes Timestamp
3. 4-bytes Difficulty target
4. 4-bytes Nonce
5. 32-bytes Previous block hash
6. 32-bytes Merkle root

Version: Through version number miners can track any changes or upgrades made in protocol

Timestamp: It indicates the time at which a particular block was hashed.

Difficulty Target (Bits): Difficulty measures the computational power to mine a new block. A higher difficulty target indicates that more computational power is required to mine a new block. This means miners need to use a higher hashing power. Difficulty is proportional to the hashing power and it keeps changing. When hashing power is higher, the difficulty level is increased and vice versa.

Nonce (Numbers Only Used Once): The nonce string is appended to the block hash and rehashed, the result hash is compared to the target difficulty number. If it is less than the target value the block is added to the blockchain. If it is not, the nonce is

changed and the process repeats until a nonce is found. This process of repeatedly guessing the right nonce is called Proof of Work. The higher the difficulty target, the more time would be taken by miners to find right nonce.

Previous block hash: Each block carries the hash of the block that was added before it in blockchain. This renders immutability to the blocks.

Merkle root: A Merkle Root is the hash of all transaction hashes in a block. Every block transaction has a unique hash associated with it and these transaction hashes are saved in the form of an upside-down Merkle Tree. The Merkle Root is at its top. Transactions are structured in a Merkle Tree in such a way that the data is organized efficiently. All transactions are hashed first and then paired with one another. A Merkle Root has information about all transactions in a block since it is the hashed version of all hashes of all transactions. Due to the tree structure (*balanced tree*) of a Merkle Tree verification of the transactions can be done quickly (*log(n)*). The Merkle Root hash is added to the block header.

3. Transactions

There are four main stages a transaction goes through:

1. Broadcast. The first step is generating a valid bitcoin transaction and then broadcasting the transaction details to the Bitcoin network.

2. Unconfirmed/Mempool. As every miner receives the transaction, it places that transaction into its *memory pool (mempool)*. The mempool is a collection of all Bitcoin transactions that are in an unconfirmed state and are still considered active. By default if a transaction has been sitting in the mempool for more than two weeks, it is considered inactive and is dropped from the mempool.

3. Confirmed by miner. When a miner discovers a new block, it decides which transaction to add in that block from its mempool. Miners choose transactions in order of transaction fees, starting with the highest one. A transaction is considered confirmed by a miner when that miner adds a block containing that transaction to the blockchain.

4. Confirmed by the network. A transaction is considered to be confirmed by the entire Bitcoin network, when the network has achieved consensus to include the transaction's block in the blockchain.

What problem/s are POW/POS trying to solve? discuss/compare (byzantine fault tolerance, reaching a single consensus on a p2p network)

A P2P system is maintained by a distributed network of users. Usually, they have no central administrator or server because each node holds a copy of the files acting both as a client and as a server to other nodes. Each node can download files from other nodes or upload files to them. Since every node stores, transmits and receives files, P2P networks tend to be faster and more efficient as their user base grows larger. Also, their distributed architecture makes P2P systems very resistant to cyberattacks. Unlike traditional models, P2P networks don't have a single point of failure.

Satoshi Nakamoto defined Bitcoin as a *"Peer-to-Peer Electronic Cash System"*. Bitcoin is a digital form of money that can be transferred from one user to another through a P2P network, which manages a distributed ledger called blockchain.

The P2P architecture that is inherent to blockchain technology is what allows Bitcoin to be transferred worldwide, without the need for intermediaries nor any central server. Bitcoin blockchain acts as a digital ledger that publicly records all activity, each node holds a copy of the blockchain and compares it to other nodes to ensure the data is accurate. P2P networks offer greater security (vaccinated from DoS attacks), resistance to malicious activity and to censorship by central authorities.

All participants of distributed network need to regularly agree on the current state of the blockchain and that is termed to consensus achievement. However, reaching consensus on distributed network in a safe and efficient way is far from being an easy task. How can a distributed network nodes agree on a decision if some of the nodes fail or act dishonestly? This is the fundamental question of the so-called Byzantine Generals' Problem (BGP), which gave birth to the concept of Byzantine fault tolerance (BFT). The main idea behind this problem is reaching consensus among all generals to act in a synchronized manner but the problem is that some of the generals may choose to act maliciously and send a fraudulent message to confuse the other generals, leading to a total failure.

In context of Bitcoin blockchain each general represents a network node and the nodes need to reach consensus on the current state of the system. To avoid complete failure the majority of participants within a distributed network have to agree and execute the same action. Therefore, the only way to achieve consensus in distributed system is by having at least 2/3 or more reliable and honest network nodes.

Byzantine fault tolerance is a property of a system that is able to resist the class of failures derived from the Byzantine Generals' Problem. It means that a BFT system is able to continue operating event if some of the nodes fail or act maliciously. There is more than one possible solution to that problem and multiple ways of building a BFT system. There are multiple approaches for a blockchain to achieve BFT and this leads to the term of *consensus algorithms*.

Consensus algorithm is a mechanism through which a blockchain network reach consensus. The most common implementations are Proof of Work (PoW) and Proof of Stake (PoS). The Bitcoin protocol describes the primary rules of the system, the PoW consensus algorithm defines how these rules will be followed in order to reach consensus (for instance, transactions validation and verification).

Satoshi Nakamoto developed a modified version of PoW concept as an algorithm that enabled creation of Bitcoin as a BFT system. PoW algorithm is not 100% tolerant to Byzantine faults but due to the cost intensive mining process and the underlying cryptographic techniques, PoW has proven to be one of the most secure and reliable implementations for blockchain networks.

Proof of Work mechanism enables Bitcoin transactions to be confirmed and blocks to be published on the blockchain. It also prevents double-spend problem. PoW requires that miners use their computational power to hash the block's data until a solution to a puzzle is found. When the right hash is found, the new block will broadcast to the network and other participants of the network update their blockchain to include the new block. The miner will be rewarded for wasting computational power and electricity. PoW mechanism motivates participants act honestly!