**Course #: 4301.002 Blockchain Exam 1**

**1. Describe in detail three of the following:**

🡪 Building secure and reliable systems is crucial, especially when dealing   
 with distributed ledgers like blockchains.

1. **Byzantine Fault Tolerance**

* Inspired by the Byzantine generals' problem, BFT ensures a system function even with malicious or faulty nodes.
* BFT is the most robust fault tolerance mechanism, but achieving consensus requires complex algorithms.
* Permissioned blockchains like Hyperledger Fabric can benefit from BFT for secure operation with known participants.

1. **Crash Fault Tolerance**

* Crash Fault Tolerance (CFT) handles situations where nodes crash or become temporarily unavailable, without any malicious intent.
* Unlike the more complex Byzantine Fault Tolerance (BFT), CFT focuses on keeping the system running even with these crashes.
* Blockchains like Bitcoin and Ethereum, which use Proof of Work (PoW), benefit from CFT because failing miners only slow down block creation, not tamper with data.

1. **Proof of Work**

* Proof of Work (PoW) is a system used in blockchains like Bitcoin where miners compete to solve complex puzzles to validate transactions and secure the network.
* The winning miner gets to add the next block to the blockchain and earn a reward.
* This competition secures the network because changing past records would require too much computing power to redo the PoW for all following blocks.
* However, Proof of Work (PoW) can be energy-intensive and slow down transaction processing.

**2. Describe in detail from start to finish how transactions are created, executed, and committed in a Bitcoin Blockchain.**

🡪 Bitcoin transactions are the main thing in the Bitcoin network. They help people

send money to each other. Creating, completing, and recording transactions in

the Bitcoin blockchain is really important for keeping the network safe and

reliable.

🡪 Following below steps are involved in the bitcoin transactions: -

* **Transaction Creation**
* In Bitcoin, initiating a transaction involves your wallet crafting a request specifying the recipient's address, amount, and additional details.
* This request is then cryptographically signed using your private key, proving ownership of the funds, and authorizing the transfer.
* **Transaction Propagation**
* Once your signed transaction is ready, it's sent out to the vast network of computers (nodes) that power Bitcoin.
* These nodes then relay the transaction to others, creating a ripple effect that spreads it for validation by miners throughout the network.
* **Transaction Validation**
* Bitcoin transactions go through a rigorous inspection by every node on the network.
* These guardians verify the sender's funds, ensure a valid digital signature using the sender's private key, and confirm the transaction follows established network rules.
* Only transactions passing all these checks are deemed legitimate and ready to be included in a block.
* **Transaction Inclusion in a Block**
* Miners bundle valid transactions into blocks, competing to solve a cryptographic puzzle that secures the block and earns them a reward.
* Transactions with higher fees are more likely to be included in the next block, officially confirming the transfer on the permanent ledger.
* **Block Confirmation**
* After a block containing your Bitcoin transaction is added to the blockchain, more blocks are stacked on top.
* The more blocks there are on top (confirmations), the more secure your transaction becomes.
* With each confirmation, the effort required to reverse the transaction skyrockets, making it practically impossible after just a few confirmations.
* **Transaction Finalization**
* After enough new blocks are added on top of the one containing your transaction, it reaches a state of finality.
* This means the transaction is considered permanent and cannot be reversed.
* Following this finalization, the bitcoin balances in the sender's and recipient's wallets are adjusted to reflect the completed transfer.

**3. Describe the installation of a Bitcoin client and the development of a test environment.**

🡪 Bitcoin client selection, in general, has two main common choices. They are as follows:

* Bitcoin Core: The official software for running a full Bitcoin node, offering full functionality but requiring significant storage space.
* Lightweight Test Clients: Specialized software for testing Bitcoin features, offering faster setup but with limited functionality.

🡪 Bitcoin offers two different types of test networks that can be used. They are as follows:

* Testnet lets you try things out on a public network with fake Bitcoin.
* Regtest is your own private playground to build and experiment with blockchains with complete control over the block generation and mining.

🡪 Client Installation and Configuration

* Download the software from the official source.
* Follow the installation wizard specific to your operating system (Windows, Mac, Linux).
* Enable the chosen test network mode (Testnet or Regtest).
* Create Testnet accounts for sending and receiving test bitcoins (if using Testnet).
* Adjust resource usage if needed for a smoother test environment.

🡪 Verifying and Running test Node

* Check for successful synchronization with the test network.

**4. Describe in detail from start to finish how transactions are created, executed, and committed in an Ethereum Blockchain.**

🡪 Ethereum is more versatile compared to Bitcoin. It can handle payments, but it also allows for more advanced applications through smart contracts.

🡪 Transactions steps are as follows:

* **Transaction Creation**
* The journey begins with a user sending a transaction through their Ethereum wallet or a connected interface.
* This transmission includes the recipient's address, any Ether amount to be sent, and relevant data (like smart contract interactions).
* Users set a gas fee, acting as a tip for miners to process the transaction swiftly and securely.
* The user's wallet cryptographically signs the transaction with their private key, proving ownership and authorization.
* **Transaction Broadcast**
* Once signed, then the transaction is broadcasted to the Ethereum peer-to-peer network.
* Nodes in the network receive and relay the transaction message, making it available to miners.
* **Transaction Execution & Validation**
* Miners compete to solve complex problems to validate transactions.
* The winning miner bundles transactions into a block and simulates them in a virtual machine.
* Valid transactions are executed and updated on the Ethereum network's database.
* **Block Committal & Confirmation**
* Validated transactions are bundled into a block, linked to the blockchain by miners, and become more secure with each new block added.
* **Transaction Finalization**
* A transaction is confirmed by miners adding it to blocks on a blockchain.
* More confirmations (typically 6-12 blocks) mean a higher chance the transaction is final.
* Once a transaction is final, wallets update to reflect the transfer.
* The sender's balance decreases, and the recipient's balance increases (for transfers) or adjusts based on the smart contract outcome.

**5. Define Smart contracts. Describe their properties, provide details on how they are deployed and executed in an Ethereum blockchain.**

🡪 Smart contracts are basically self-running agreements on blockchains like

Ethereum. They automatically execute the deal when certain conditions are

met, without needing a middleman.

🡪 Smart contracts properties are as follows:

* **Immutable:** The code behind a smart contract can't be altered once it's on the blockchain, guaranteeing transparency and secure execution of the agreement.
* **Autonomous:** Smart contracts run by themselves based on their built-in instructions, removing the need for middlemen.
* **Trustless:** There's no need for trust between parties because the code controls the execution, fostering trust in the system itself.
* **Secure:** Smart contracts inherit the security of blockchain technology, making them resistant to tampering and fraud.
* **Transparent:** All transactions involving a smart contract are visible on the blockchain, ensuring everyone can be held accountable.

🡪 Deployment and execution are as follows:

* First, you write the smart contract code in Solidity, a language designed for Ethereum.
* This code is then compiled into bytecode, a set of instructions the Ethereum network understands.
* The bytecode is submitted as a transaction, along with a fee, to the Ethereum network.
* Miners compete to validate the transaction and add it to a permanent record on the blockchain.
* Once validated, the Ethereum Virtual Machine on all computers in the network executes the smart contract's instructions.
* People can then interact with the smart contract by sending transactions to its unique address on the blockchain. These transactions can provide data or trigger specific actions within the contract, again executed by the Ethereum Virtual Machine.

**7. Describe in detail the form starts to finish how transactions are created, executed, and committed in Hyperledger Fabric.**

🡪 Hyperledger Fabric is a private blockchain for businesses, where transactions securely follow a specific process to keep things confidential.

* **Transaction initiated by Client and Defines**
* The client application initiates a transaction request, specifying the desired action (function) and relevant data.
* It also defines the endorsement policy, outlining which organizations need to approve the transaction.
* **Proposal Creation**
* A client builds a transaction proposal specifying the function to call, any required data, and which organizations must approve the transaction.
* **Sending proposal for Endorsement**
* The client chooses reviewers (endorsing peers) based on the proposal's rules, and these reviewers are network members who test the transaction and confirm its results.
* The client sends the transaction proposal to the chosen reviewers for their verification.
* **Endorsement validation and Collection by Peers**
* Each peer simulates the transaction on their own copy of the ledger to ensure its valid and follows the rules.
* If the simulation passes, a peer signs off, basically saying they approve of the transaction's legitimacy. The client collects responses from validators, which can be either approvals or rejections.
* **Transaction Construction & Ordering**
* The client combines the proposal with endorsements to create a complete transaction.
* This transaction is submitted to the ordering service, which sequences transactions for inclusion in blocks.
* The ordering service uses a consensus mechanism to order transactions and create a block.
* Then the block of all ordered transactions as one group is created by the ordering service.
* **Transaction Broadcasting and Validation**
* The created block is then broadcast to all peers in the network.
* Each peer validates the endorsements against the policy within the block.
* **Ledger Commitment**
* If validation is successful, peers update their local ledgers with the transaction results.
* The client may be notified of the successful transaction commitment.

**8. Describe with examples three of the following:**

1. **Hash functions**

* Think of a hash function like a special shredder. It takes any kind of data and turns it into a unique, fixed-sized code, like a fingerprint.
* It's always the same for the same data, so you can check if something hasn't been tampered with.
* You can't turn the code back into the original data, keeping it confidential.
* It's very difficult to find two different things that get shredded into the same code, preventing forgeries.
* For example: - Using hash method like MD5 on the sentence “I am Het Patel, a computer science major at the University of Texas at Dallas working towards my bachelor's degree” produces unique hash code as,
* MD5 - ebb76afa44a7435f773e488cce8f77e0

1. **SHA 256**

* SHA-256 is a secure tool that creates a unique digital fingerprint for any data.
* It's commonly used to safely store passwords and verify file integrity.
* Instead of storing passwords directly, SHA-256 turns them into a code that can't be reversed to reveal the original password.
* Similarly, it can verify if a downloaded file is corrupted by comparing its fingerprint with a known good one.
* For example: - Using hash method like SHA-256 on the sentence “I am Het Patel, a computer science major at the University of Texas at Dallas working towards my bachelor's degree” produces unique hash code as,
* SHA-256 -- 137964c2aa7c6cc81f87f02a7504406bb03b08e1c32760672c835758c5370168

1. **Message Authentication Code**

* A MAC code acts like a tamper-proof seal for digital messages.
* It uses a secret key to create a unique fingerprint for the message.
* The receiver, with the same key, can verify if the message arrived unchanged by recreating the fingerprint and comparing it to the one sent by the sender.
* Example as follows:
* Ashley wants to send a message to Tom and make sure it arrives exactly as she sent it. She can use a special code (MAC algorithm) and a secret key that only they know. This code generates a unique fingerprint for the message, which gets attached to it. When Tom receives the message, he can use the same code and secret key to create his own fingerprint. If the two fingerprints match, Tom knows the message is authentic and hasn't been tampered with.

**9. Describe a layered architecture for a Blockchain and the details of each layer.**

🡪 Blockchains use a stacked system to handle the intricate connections between   
 data storage, communication across the network, and the features of various   
 applications.

* **Network Layer**
* **Blockchains rely on the internet, acting as the most basic layer, to enable communication between computers on the network.**
* **P2P Layer**
* In a peer-to-peer network, devices share information directly with each other, relying on messaging techniques like gossip or flooding to spread data across the network.
* **Cryptography Layer**
* The cryptography layer acts as the blockchain's security shield. It employs cryptographic tools like digital signatures and hashing functions to safeguard data integrity and secure communication within the network.
* **Consensus Layer**
* The consensus layer is vital for blockchains. It uses different methods, like SMR, proof-based mechanisms, to make sure everyone on the network agrees on the validity of transactions. This keeps the system secure and trustworthy.
* **Execution Layer**
* This layer includes things like virtual machines (like the EVM), blocks, transactions, and smart contracts.
* Execution layer carries out tasks like moving digital assets around, executing smart contracts, and creating new blocks on the blockchain.
* **Application layer**
* The top layer of blockchain, called application layer, acts as the bridge between users and the blockchain.
* It holds all the user-friendly programs and tools needed to interact with the blockchain.
* These tools include smart contracts, decentralized applications (dApps), and even programs that can run on their own.
* This layer lets users interact with the blockchain through these dApps, instead of needing to directly access the complex underlying system.