Foundation of Blockchain (4361603) - Summer 2025 Solution

Milav Dabgar

May 14, 2025

Question 1(a) [3 marks]

Differentiate between Private key and Public key in Blockchain.

Aspect	Private Key	Public Key
Purpose	Used for signing transactions	Used for verification
Sharing	Must be kept secret	Can be shared publicly
Function	Decrypts data, creates signatures	Encrypts data, verifies signatures
Ownership	Only owner knows it	Everyone can access it

- **Private Key**: Secret mathematical code that proves ownership
- **Public Key**: Open address that others use to send transactions
- **Security**: Private key loss = permanent fund loss

Mnemonic

Private is Personal, Public is Posted

Question 1(b) [4 marks]

Explain Distributed Ledger in detail.

Distributed Ledger is a database spread across multiple locations and participants.

Feature	Description
Decentralized	No single control point
Synchronized	All copies stay updated
Transparent	All participants can view
Immutable	Cannot be easily changed

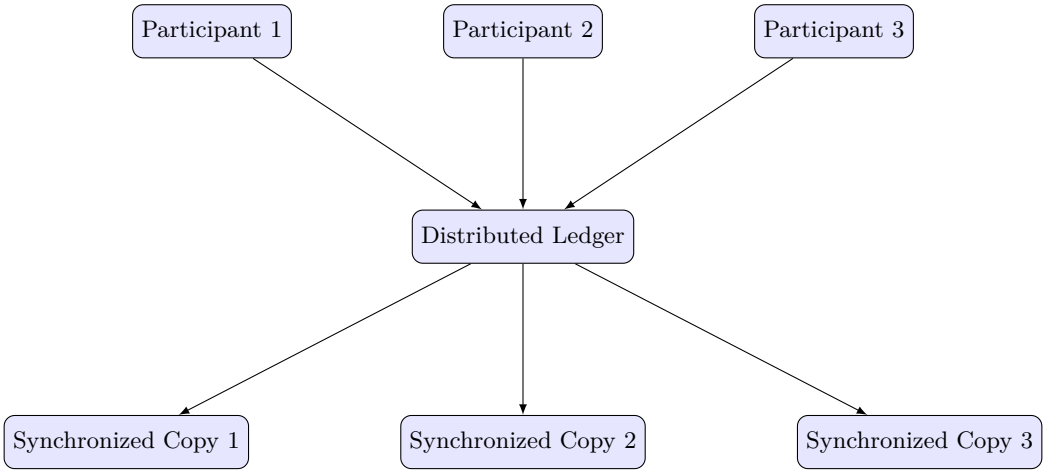

Figure 1. Distributed Ledger System

- **Benefits**: Eliminates intermediaries, increases trust, reduces fraud
- **Working**: All participants maintain identical copies of records

Mnemonic

Distributed = Divided but Identical

Question 1(c) [7 marks]

Define Blockchain. Describe applications and limits of Blockchain.

Blockchain Definition: A chain of blocks containing transaction records, linked using cryptography.

Applications:

Sector	Application	Benefit
Finance	Cryptocurrency, payments	Faster, cheaper transfers
Healthcare	Patient records	Secure, accessible data
Supply Chain	Product tracking	Transparency, authenticity
Real Estate	Property records	Fraud prevention
Voting	Digital elections	Transparent, tamper-proof

Limits:

Limitation	Impact
Scalability	Slow transaction processing
Energy Usage	High electricity consumption
Complexity	Difficult for users to understand
Regulation	Legal uncertainty
Storage	Growing data size problems

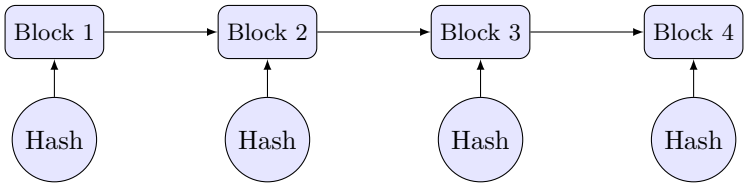

Figure 2. Blockchain Architecture

- **Security**: Cryptographic linking makes tampering difficult
- **Transparency**: All transactions visible to network participants

Mnemonic

Blocks Chained = Blockchain, Apps Many = Limits Many

OR

Question 1(c) [7 marks]

Write a short note on: CAP Theorem in Blockchain

CAP Theorem states that distributed systems can only guarantee 2 out of 3 properties simultaneously.

Property	Description	Example
Consistency	All nodes have same data	Same balance shown everywhere
Availability	System always responds	Network never goes down
Partition Tolerance	Works despite network failures	Functions even if nodes disconnect

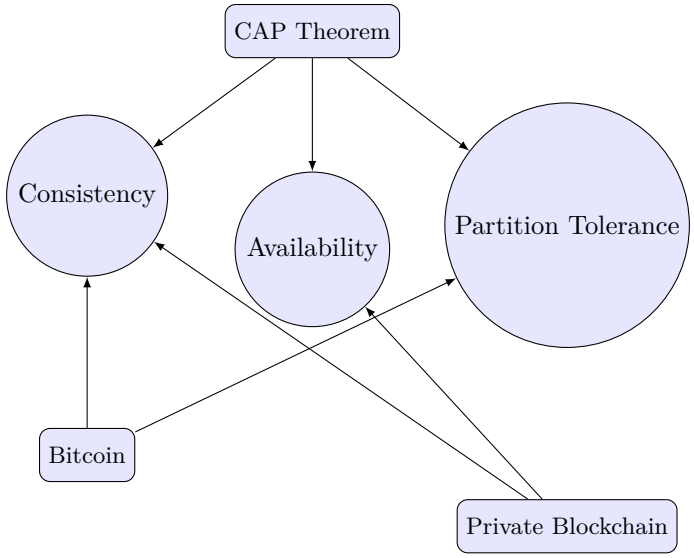

Figure 3. CAP Theorem and Blockchain Trade-offs

Real-world Applications:

Blockchain Type	Chooses	Sacrifices
Bitcoin	Consistency + Partition	Availability
Ethereum	Consistency + Partition	Availability
Private Networks	Consistency + Availability	Partition Tolerance

- **Impact**: Blockchain designers must choose which property to sacrifice
- **Trade-off**: Perfect systems impossible in distributed networks

Mnemonic

Can't Always Please - Choose 2 of 3

Question 2(a) [3 marks]

Explain Data Structure of a Blockchain.

Blockchain Data Structure consists of linked blocks containing transaction data.

Component	Purpose
Block Header	Contains metadata
Previous Hash	Links to previous block
Merkle Root	Summary of all transactions
Timestamp	When block was created
Transactions	Actual data/transfers

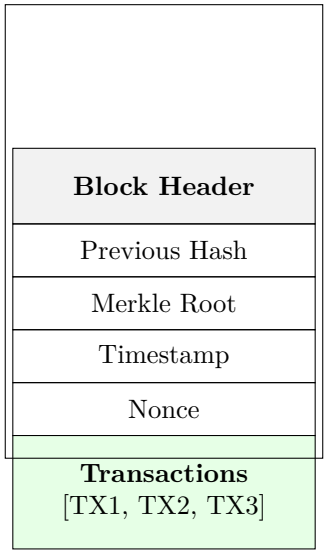

Figure 4. Structure of a Block

- **Linking**: Each block points to previous block using hash
- **Integrity**: Changing one block breaks the entire chain

Mnemonic

Header Holds, Transactions Tell

Question 2(b) [4 marks]

What are the benefits of Decentralization?

Decentralization Benefits:

Benefit	Explanation
No Single Point of Failure	Network continues if one node fails
Censorship Resistance	No authority can block transactions
Transparency	All participants see same information
Reduced Costs	Eliminates intermediary fees
Trust	No need to trust central authority

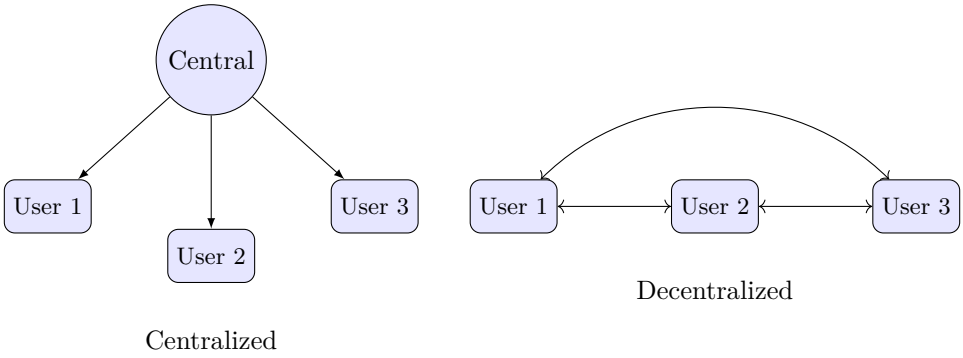

Figure 5. Centralized vs. Decentralized Networks

- **Security**: Multiple copies prevent data loss
- **Democracy**: All participants have equal rights
- **Resilience**: System survives individual failures

Mnemonic

Distributed = Durable, Democratic, Direct

Question 2(c) [7 marks]

Differentiate between Public Blockchain and Private Blockchain.

Comprehensive Comparison:

Aspect	Public Blockchain	Private Blockchain
Access	Open to everyone	Restricted to specific users
Permission	Permissionless	Requires permission
Control	Decentralized	Centralized control
Speed	Slower (consensus needed)	Faster (fewer validators)
Security	High (many validators)	Medium (fewer validators)
Cost	Transaction fees required	Lower operational costs
Transparency	Fully transparent	Limited transparency
Examples	Bitcoin, Ethereum	Hyperledger, R3 Corda

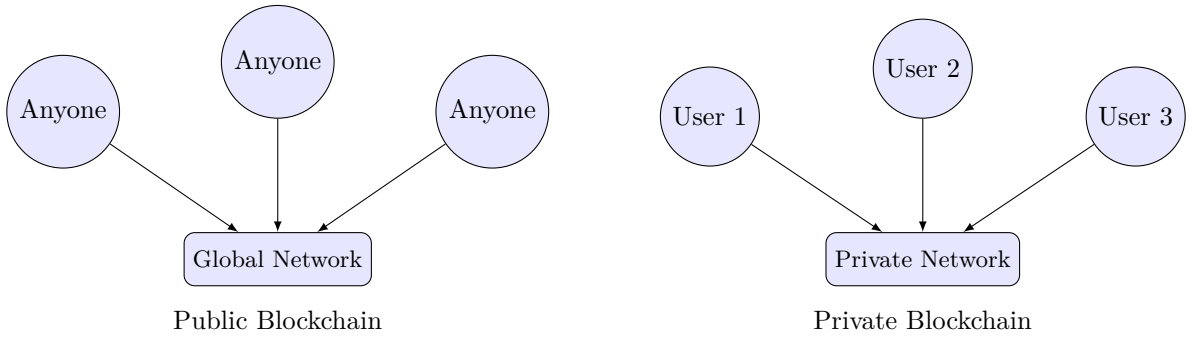

Figure 6. Public vs Private Architecture

- **Trade-offs**: Public offers more security, Private offers more control
- **Choice**: Depends on transparency vs. privacy needs

Mnemonic

Public = People's, Private = Permitted

OR

Question 2(a) [3 marks]

Describe Core Components of Block Chain with suitable diagram.

Core Components:

Component	Function
Blocks	Store transaction data
Hash Functions	Create unique fingerprints
Digital Signatures	Verify transaction authenticity
Consensus Mechanism	Agree on valid transactions
Peer-to-Peer Network	Connect all participants

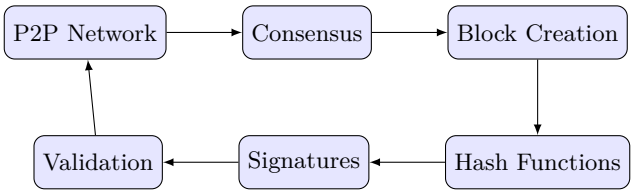

Figure 7. Blockchain Core Components Interaction

- **Integration**: All components work together for security
- **Purpose**: Each component serves specific blockchain function

Mnemonic

Blocks Build, Hash Holds, Signatures Secure

OR

Question 2(b) [4 marks]

Define and explain permissioned blockchain in detail.

Permissioned Blockchain Definition: A blockchain where participation requires explicit permission from network administrators.

Feature	Description
Access Control	Only approved users can join
Validation Rights	Selected nodes validate transactions
Governance	Central authority manages network
Privacy	Transaction details can be private

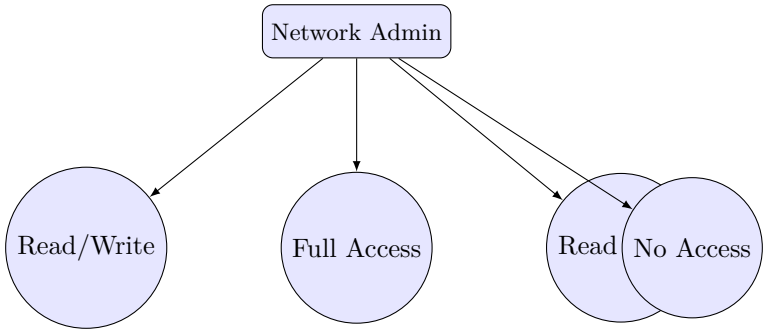

Figure 8. Permission Levels in Permissioned Blockchain

- **Benefits**: Better privacy, regulatory compliance, faster processing
- **Drawbacks**: Less decentralized, requires trust in administrators

Mnemonic

Permission = Participation Permitted

OR

Question 2(c) [7 marks]

Explain sidechain in brief.

Sidechain Definition: A separate blockchain connected to main blockchain, allowing asset transfer between chains.

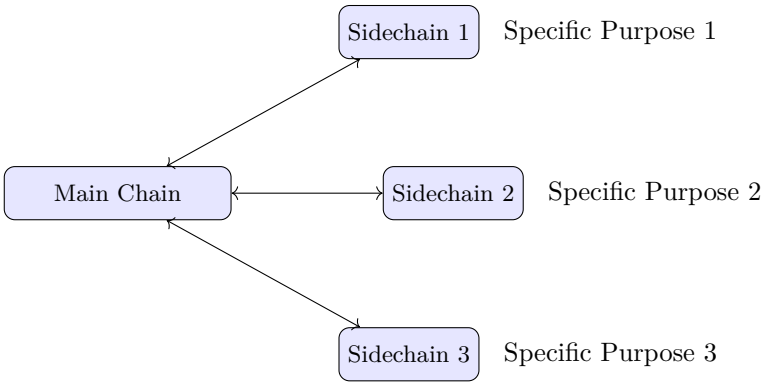

Figure 9. Sidechain Architecture

Benefits and Features:

Aspect	Benefit
Scalability	Reduces main chain load
Experimentation	Test new features safely
Specialization	Optimized for specific use cases
Interoperability	Connect different blockchains

Transfer Process:

1. **Lock**: Assets locked on main chain
2. **Proof**: Cryptographic proof generated
3. **Release**: Equivalent assets released on sidechain
4. **Use**: Assets used on sidechain
5. **Return**: Reverse process to return assets

Real Examples:

Sidechain	Purpose
Lightning Network	Fast Bitcoin payments
Plasma	Ethereum scaling
Liquid	Bitcoin trading

- **Security**: Maintains connection to secure main chain
- **Flexibility**: Each sidechain can have different rules

Mnemonic

Side Supports, Main Maintains

Question 3(a) [3 marks]

Define Consensus Mechanism and explain any one in detail.

Consensus Mechanism Definition: A protocol that ensures all network participants agree on the blockchain's current state.

Proof of Work (PoW) Explanation:

Component	Function
Mining	Solving complex mathematical puzzles
Competition	Miners compete to solve first
Verification	Network verifies solution
Reward	Winner gets cryptocurrency reward

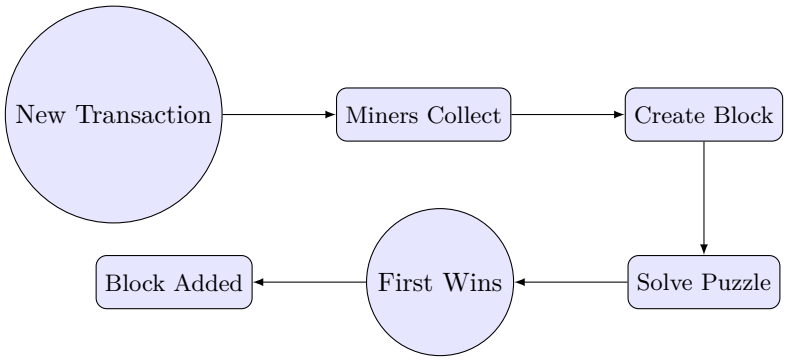

Figure 10. Proof of Work Process

- **Security**: Computational work makes tampering expensive
- **Example**: Bitcoin uses Proof of Work consensus

Mnemonic

Consensus = Common Sense, Work = Win

Question 3(b) [4 marks]

Why is Forking needed in Blockchain? List various types of Forks in Blockchain.

Why Forking is Needed:

Reason	Purpose
Upgrades	Add new features to blockchain
Bug Fixes	Correct security vulnerabilities
Rule Changes	Modify consensus rules
Community Disagreement	Split when no consensus reached

Types of Forks:

Fork Type	Description	Compatibility
Soft Fork	Tightens rules	Backward compatible
Hard Fork	Changes rules completely	Not backward compatible
Accidental Fork	Temporary split	Resolves automatically
Contentious Fork	Community disagreement	Permanent split

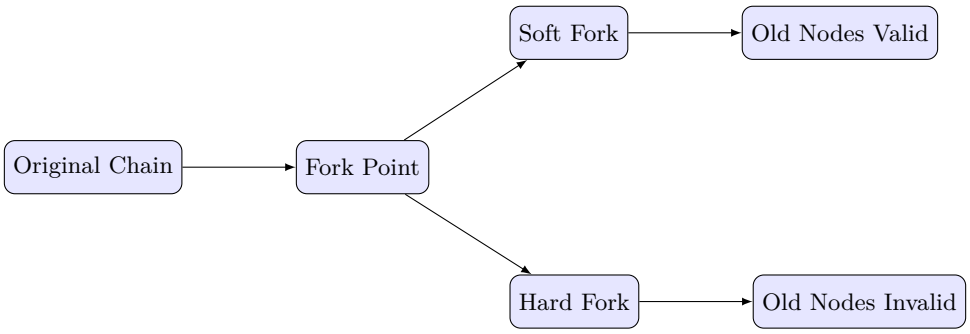

Figure 11. Soft vs Hard Fork

- **Impact**: Forks can create new cryptocurrencies
- **Examples**: Bitcoin Cash (hard fork), Ethereum updates (soft forks)

Mnemonic

Fork = Future Options, Rules Kept

Question 3(c) [7 marks]

What is Bitcoin Mining? Explain working, difficulty and benefits of Bitcoin mining in detail.

Bitcoin Mining Definition: Process of adding new transactions to Bitcoin blockchain by solving computational puzzles.

Mining Process:

1. **Collection**: Gather pending transactions from mempool
2. **Block Creation**: Form new block including transactions
3. **Puzzle Solving**: Find correct nonce through trial and error
4. **Verification**: Network checks solution and validates block
5. **Addition**: Add block to chain as permanent record
6. **Reward**: Miner gets Bitcoin (Currently 6.25 BTC)

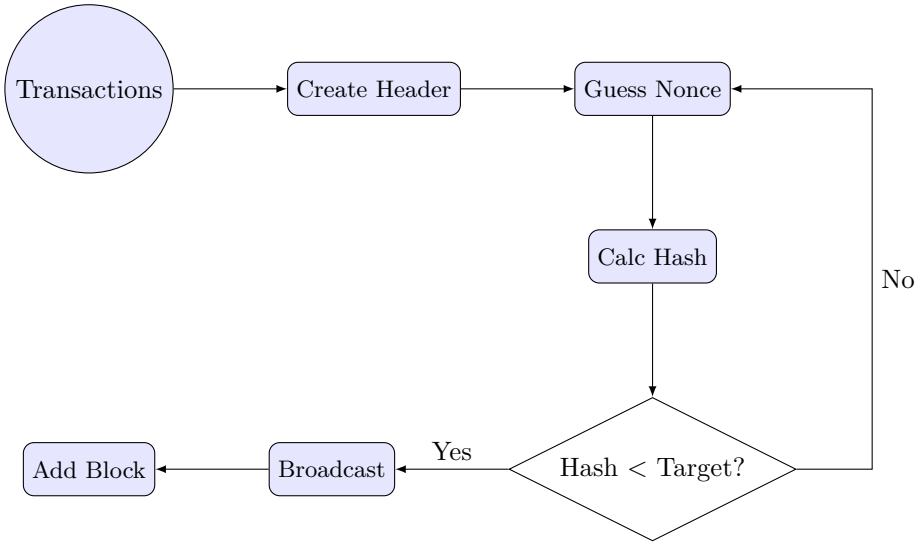

Figure 12. Bitcoin Mining Workflow

Difficulty Adjustment:

Aspect	Mechanism
Target Time	10 minutes per block
Adjustment Period	Every 2016 blocks (2 weeks)
Auto-Regulation	Increases if blocks too fast
Purpose	Maintain consistent block time

Benefits of Mining:

- **Financial Reward**: Earn Bitcoin for successful mining
- **Network Security**: More miners = more secure network
- **Transaction Processing**: Enables Bitcoin transfers
- **Decentralization**: No central authority needed

Mnemonic

Mining = Money, Math, Maintenance

OR

Question 3(a) [3 marks]

Differentiate Soft fork and Hard fork.

Fork Comparison:

Aspect	Soft Fork	Hard Fork
Compatibility	Backward compatible	Not backward compatible
Rules	Makes rules stricter	Changes rules completely
Node Updates	Optional for old nodes	Mandatory for all nodes
Chain Split	No permanent split	Can create permanent split
Consensus	Easier to implement	Requires majority agreement
Examples	SegWit (Bitcoin)	Bitcoin Cash, Ethereum Classic

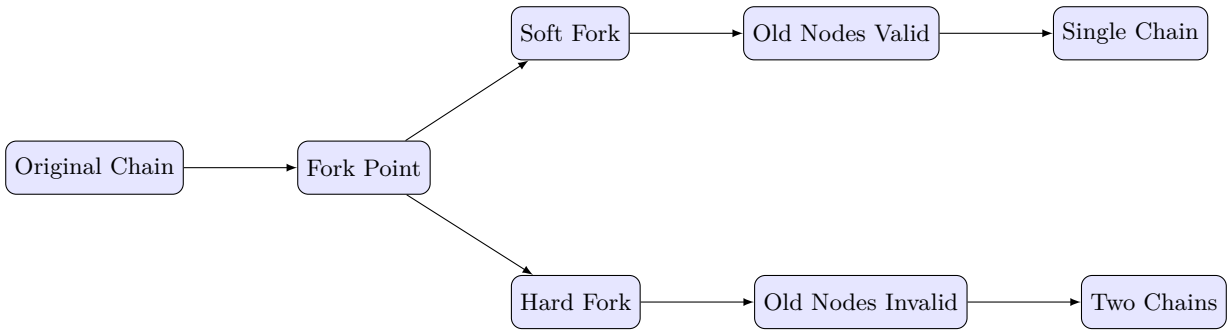

Figure 13. Soft Fork vs Hard Fork Outcome

- **Risk**: Hard forks can split community and create competing currencies
- **Safety**: Soft forks are generally safer and less disruptive

Mnemonic

Soft = Same Direction, Hard = Huge Difference

OR

Question 3(b) [4 marks]

What is the importance of Finality in the World of Blockchain?

Finality Definition: The guarantee that once a transaction is confirmed, it cannot be reversed or altered.

Importance:

Aspect	Importance
Trust	Users confident transactions are permanent
Business Use	Companies can rely on completed transactions
Legal Certainty	Courts can enforce blockchain records
Settlement	Financial institutions can clear payments

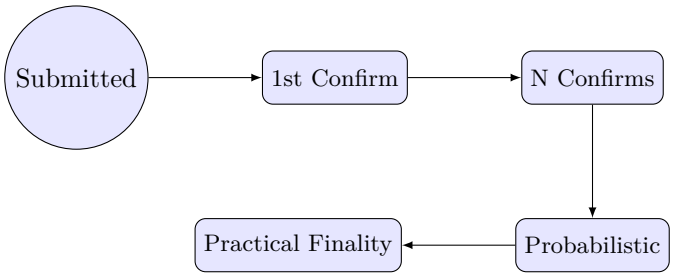

Figure 14. Consensus and Finality Process

- **Bitcoin**: 6 confirmations generally considered final
- **Ethereum**: Moving toward faster finality with Proof of Stake

Mnemonic

Final = Forever, Important = Irreversible

OR

Question 3(c) [7 marks]

What is a 51% attack in Blockchain? Explain in brief.

51% Attack Definition: When a single entity controls more than 50% of network's mining power or validators, allowing them to manipulate the blockchain.

Attack Mechanism:

1. **Control**: Gain >50% mining power to dominate network
2. **Double Spend**: Create secret chain to prepare alternative history
3. **Execute**: Release longer chain so network accepts fake version
4. **Profit**: Spend coins twice to steal from victims

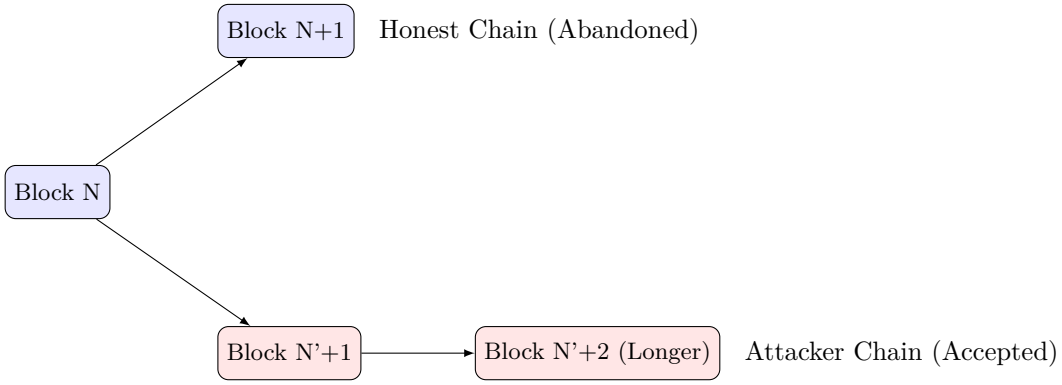

Figure 15. 51% Attack: Longest Chain Rule Abuse

Prevention Methods:

Method	How It Helps
Decentralization	Spread mining across many participants
High Hash Rate	Make attack economically unfeasible
Proof of Stake	Attackers lose their staked coins

Mnemonic

51% = Majority Mischief, Control = Chaos

Question 4(a) [3 marks]

Describe various types of Hyperledger projects.

Hyperledger Project Types:

Project	Purpose	Use Case
Fabric	Modular blockchain platform	Enterprise applications
Sawtooth	Scalable blockchain suite	Supply chain, IoT
Iroha	Mobile-focused blockchain	Identity management
Indy	Digital identity platform	Self-sovereign identity

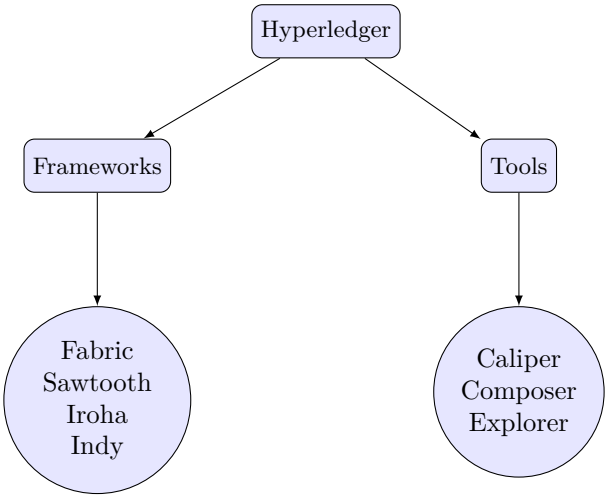

Figure 16. Hyperledger Ecosystem

Mnemonic

Hyper = High Performance, Ledger = Large Enterprise

Question 4(b) [4 marks]

Differentiate between Blockchain and Bitcoin.

Comprehensive Comparison:

Aspect	Blockchain	Bitcoin
Definition	Technology/Platform	Digital Currency
Scope	Broader concept	Specific application
Purpose	Record keeping system	Peer-to-peer payments
Applications	Many industries	Primarily financial
Flexibility	Can be customized	Fixed protocol

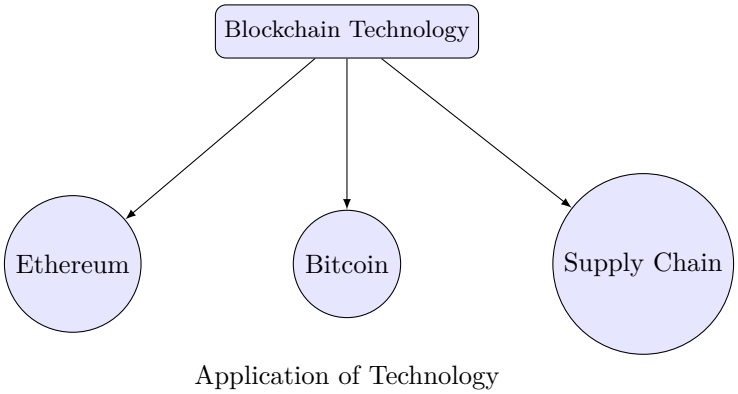

Figure 17. Blockchain (Platform) vs Bitcoin (App)

Mnemonic

Blockchain = Building Block, Bitcoin = Specific Brick

Question 4(c) [7 marks]

Write a short note on: Merkle Tree

Merkle Tree Definition: A binary tree structure where each leaf represents a transaction hash, and each internal node contains the hash of its children.

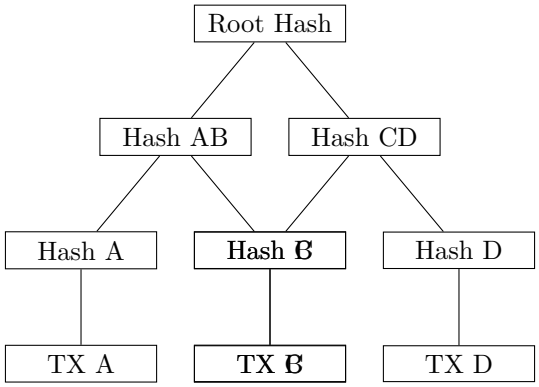

Figure 18. Merkle Tree Structure

Benefits:

- **Efficiency**: Quick verification without downloading all data
- **Security**: Any change detected immediately
- **Scalability**: Verification time stays constant

Verification Process:

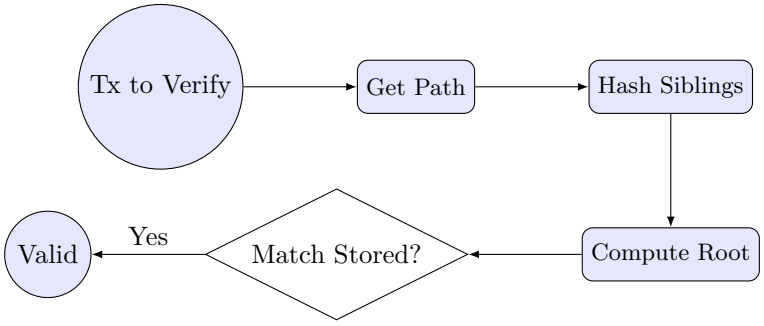

Figure 19. Merkle Verification Flow

Mnemonic

Merkle = Many Made One, Tree = Trustworthy

OR

Question 4(a) [3 marks]

Discuss briefly about Hash pointer and how it is used in Merkle tree.

Hash Pointer Definition: A data structure containing both the location of data and cryptographic hash of that data.

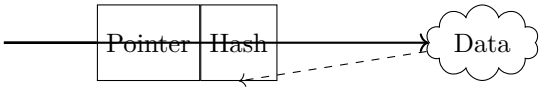

Figure 20. Hash Pointer Concept

Usage in Merkle Tree:

Level	Hash Pointer Function
Leaf Level	Points to transaction, contains transaction hash
Internal Nodes	Points to children, contains combined hash
Root	Points to tree structure, contains overall hash

- **Verification**: Can detect any change in tree structure
- **Navigation**: Allows efficient traversal of tree

Mnemonic

Hash Pointer = Location + Verification

OR

Question 4(b) [4 marks]

What is Hashing in Blockchain? How it is useful in Bitcoin?

Hashing Definition: Mathematical function that converts input data into fixed-size string of characters.

Bitcoin Usage:

Use Case	Purpose
Block Linking	Each block contains hash of previous block
Mining	Find hash meeting difficulty requirement
Transaction IDs	Unique identifier for each transaction
Merkle Root	Summarize all transactions in block

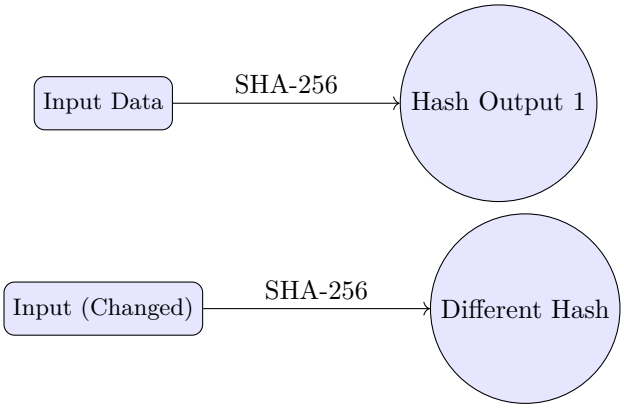

Figure 21. Avalanche Effect in Hashing

Mnemonic

Hash = Fingerprint, Bitcoin = Built on Hashing

OR

Question 4(c) [7 marks]

Explain classic Byzantine generals problem and Practical Byzantine Fault Tolerance in detail.

Byzantine Generals Problem: A classic problem about achieving consensus in distributed systems with unreliable participants.

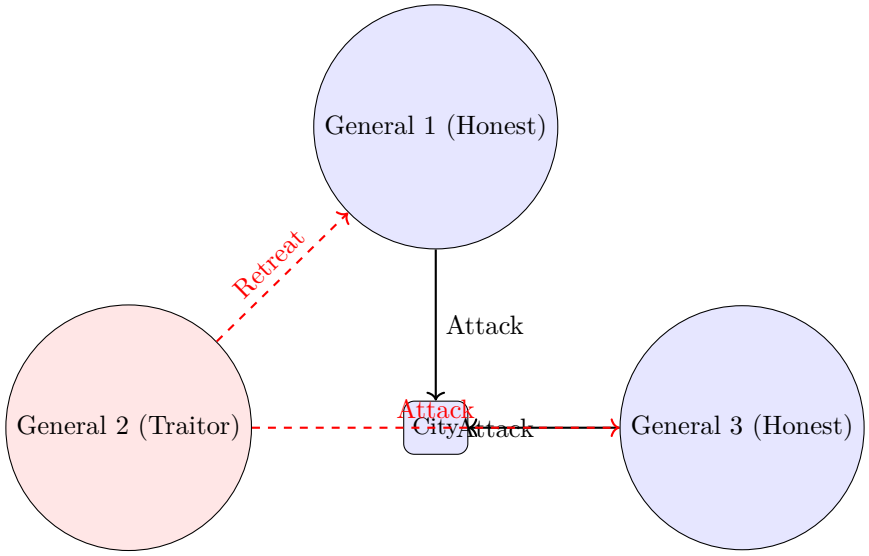

Figure 22. Byzantine Generals Traitor Scenario

Practical Byzantine Fault Tolerance (pBFT):

Phase	Action	Purpose
Pre-prepare	Leader broadcasts proposal	Initiate consensus round
Prepare	Nodes validate	Ensure proposal is seen by all
Commit	Nodes commit	Finalize consensus

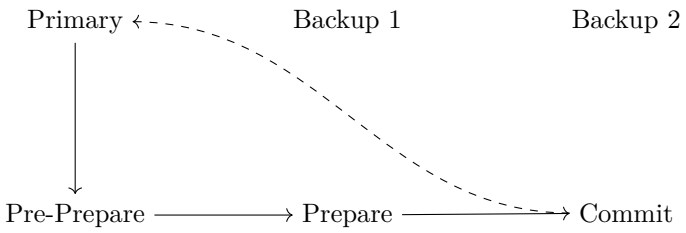

Figure 23. pBFT Consensus Phases

- **Advantage**: Fast finality, good for permissioned networks
- **Limitation**: High communication overhead O(n squared)

Mnemonic

Byzantine = Bad actors, pBFT = Practical Fix

Question 5(a) [3 marks]

List and explain cryptocurrency wallets in blockchain.

Wallet Types:

Wallet Type	Description
Hardware	Physical device storing keys (High Security)
Software	Application on computer/phone
Paper	Keys printed on paper
Web	Online wallet service

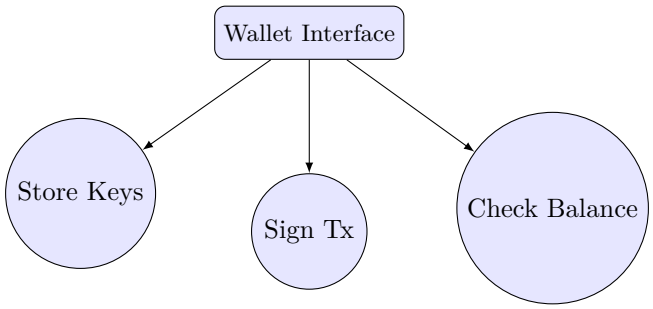

Figure 24. Wallet Functions

Mnemonic

Wallet = Key Keeper, Not Coin Container

Question 5(b) [4 marks]

Write advantages and disadvantages of ERC-20 token.

ERC-20 Token Definition: Standard protocol for creating tokens on Ethereum blockchain.

Aspect	Advantage	Disadvantage
Standardization	Work same way everywhere	Limited customization
Interoperability	Compatible with wallets	-
Development	Easy to create	Easy to create scams
Cost	-	Gas fees can be high

Mnemonic

ERC-20 = Easy and Expensive

Question 5(c) [7 marks]

What are dApps used for? Explain advantages and disadvantages of dApps.

dApps Definition: Decentralized Applications that run on blockchain networks without central authority.

Usage Categories: DeFi, Gaming, Social Media, Marketplaces, Governance.

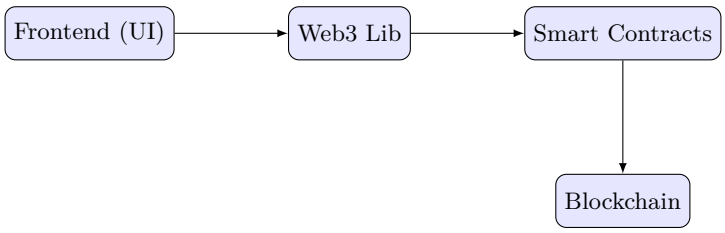

Figure 25. dApp Architecture

Advantages vs Disadvantages:

Advantages	Disadvantages
Censorship Resistance	Poor User Experience
Transparency	Scalability Issues
No Downtime	High Costs (Gas)
User Ownership	Technical Complexity

Mnemonic

dApps = Decentralized but Difficult

OR

Question 5(a) [3 marks]

Explain tokenized and token less Blockchain in detail.

Aspect	Tokenized	Token-less
Definition	With native crypto	No native crypto
Purpose	Incentive	Record keeping
Example	Bitcoin, Ethereum	Hyperledger Fabric
Access	Public (usually)	Private (Permissioned)

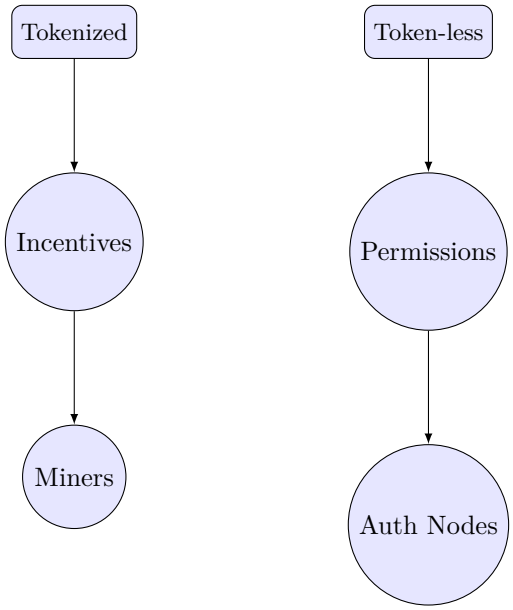

Figure 26. Incentive Models

Mnemonic

Token = Public Participation, Token-less = Private Permission

OR

Question 5(b) [4 marks]

Write advantages and disadvantages of Hyperledger.

Advantages:

- **Enterprise Focus**: Designed for business use cases
- **Privacy**: Confidential transactions possible
- **Performance**: Higher transaction throughput
- **Permissioned**: Control who can participate

Disadvantages:

- **Centralization**: Less decentralized than public blockchains
- **Complexity**: Requires technical expertise
- **No Token Economy**: Cannot leverage crypto incentives

Mnemonic

Hyperledger = High Performance, Low Publicity

OR

Question 5(c) [7 marks]

Explain Smart contract. Write various applications of smart contract.

Smart Contract Definition: Self-executing contracts with terms directly written into code, automatically enforced on blockchain.

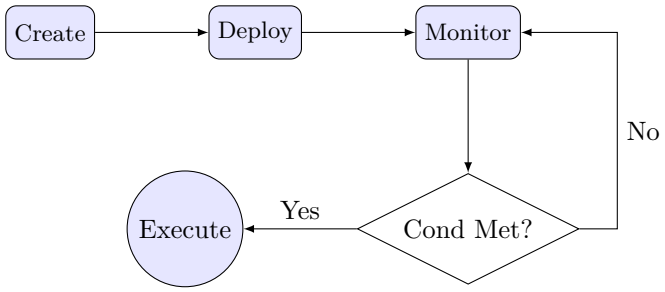

Figure 27. Smart Contract Workflow

Applications:

Industry	Application
Finance	Automated loans (DeFi)
Real Estate	Property transfers without agents
Supply Chain	Automated tracking and payments
Insurance	Automatic claim payouts

Mnemonic

Smart Contract = Self-executing, Solves Problems