ABHYASMITRA.IN

COMPREHENSIVE UNIVERSITY STUDY NOTES

BLOCKCHAIN TECHNOLOGY

Unit I — Introduction to Blockchain & Cryptographic Foundations

---

Topics Covered in this Unit

Cryptography in Blockchain: Symmetric Key vs. Asymmetric Key Cryptography (Public-Private Pairs)
Elliptic Curve Cryptography (ECC): Mathematical Foundations, Discrete Logarithm & secp256k1 Curve
Cryptographic Hash Functions: Collision Resistance, Pre-Image Resistance, Avalanche Effect & SHA-256
Digital Signature Algorithm (DSA / ECDSA): Key Generation, Signing Math & Verification Process
Merkle Trees (Hash Trees): Binary Tree Architecture, Merkle Root Calculation & SPV Merkle Proofs
Blockchain Fundamentals: History & Evolution, Limitations of Centralized Systems & SPoF
Decentralized Systems: Distributed Ledger Technology (DLT), P2P Networks & Trustless Consensus
Architectural Layers of Blockchain: Infrastructure, Network, Consensus, Data, Execution & Application
Types of Blockchain: Public (Permissionless), Private (Permissioned) & Consortium (Federated)
Master Comparison Table across Blockchain Types & High-Yield University Exam Question Bank

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & IT Students

COURSE SYLLABUS & MAPPING

[UNIT I SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit I. Verify with your university curriculum mapping.

Unit I Syllabus Breakdown:

Unit I - Introduction to Blockchain and Cryptographic Foundations: Cryptography: Symmetric Key Cryptography and Asymmetric Key Cryptography, Elliptic Curve Cryptography (ECC), Cryptographic Hash Functions: SHA256, Digital Signature Algorithm (DSA), Merkle Trees Blockchain Basics: History, Limitation of Centralized System, Decentralized Systems, Layers in Blockchain Types of Blockchain: Public, Private and Consortium.

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

1.1 Cryptographic Foundations of Blockchain

Definition: Cryptography in Distributed Ledgers

Cryptography is the mathematical foundation of blockchain technology, ensuring Data Immutability, User Authentication, Transaction Integrity, and Non-Repudiation across a trustless, decentralized peer-to-peer network without relying on a central intermediary authority.

Symmetric vs. Asymmetric Key Cryptography

Comparison Dimension	Symmetric Key Cryptography (e.g., AES)	Asymmetric Key Cryptography (Public-Key)
Key Architecture	Uses a single, shared secret key for both encryption and decryption.	Uses a mathematically linked key pair: Public Key (shared freely) & Private Key (kept strictly secret).
Primary Role in Blockchain	Used for local wallet database encryption (e.g., encrypting keystore JSON files).	Used for Identity, Digital Signatures (ECDSA), and Address Generation in Bitcoin/Ethereum.
Key Distribution Problem	Severe: securely transmitting the shared key over untrusted networks is vulnerable.	Zero key exchange risk: public keys are broadcast publicly; private keys never leave local devices.
Computational Speed	Extremely fast and lightweight (hardware-accelerated).	Computationally heavier due to complex modular mathematics and elliptic curve point scalar operations.

Elliptic Curve Cryptography (ECC) & The secp256k1 Curve

Elliptic Curve Cryptography (ECC) is an asymmetric public-key cryptographic scheme based on the algebraic structure of elliptic curves over finite fields. Both Bitcoin and Ethereum utilize the standardized curve secp256k1.

- **Elliptic Curve Equation (Weierstrass Form):** y^2 = (x^3 + 7) mod p (where p = 2^256 - 2^32 - 977 is a massive 256-bit prime).
- **The Elliptic Curve Discrete Logarithm Problem (ECDLP):** Given a predefined generator base point G on the curve and a private scalar key d in [1, n-1], computing the Public Key point Q via scalar point multiplication is easy: Q = d * G. However, given Q and G, it is computationally infeasible to recover the private key d! (Requires O(sqrt(n)) = 2^128 operations, impossible with modern supercomputers).

- **Key Advantage of ECC over RSA:** A 256-bit ECC private key delivers the same cryptographic security strength as a 3072-bit RSA key while consuming drastically lower memory, bandwidth, and computational power.

1.2 Cryptographic Hash Functions & Digital Signatures

Five Essential Properties of Cryptographic Hash Functions

A cryptographic hash function H: {0, 1} -> {0, 1}^k maps arbitrary-length input data to a fixed-size digest:*

1. Deterministic: The identical input message m always produces the exact same hash output H(m).

2. Fast Computation: Efficiently computable for any given input.

3. Pre-Image Resistance (One-Way): Given hash h, it is computationally impossible to find m such that H(m) = h.

4. Second Pre-Image Resistance (Weak Collision Resistance): Given m_1, it is impossible to find m_2 != m_1 such that H(m_1) = H(m_2).

5. Collision Resistance (Strong Collision Resistance): Impossible to find any arbitrary pair (m_1, m_2) with H(m_1) = H(m_2).

6. Avalanche Effect: Changing a single bit in the input causes a drastic, unpredictable change (> 50% bits) in output digest.

The SHA-256 Algorithm

Secure Hash Algorithm 256-bit (SHA-256) is a NIST standard cryptographic hash function based on the Merkle-Damgard construction. It processes padded input in 512-bit message blocks through 64 rounds of non-linear logical bitwise functions (Ch, Maj, Sigma_0, Sigma_1) to produce a 256-bit (32-byte / 64-hex character) output hash.

Digital Signature Algorithm (ECDSA in Blockchain)

A Digital Signature is a mathematical mechanism that binds an author's private key to a digital transaction message, guaranteeing Authentication, Data Integrity, and Non-Repudiation.

- **1. Signing Process (Sender Alice):** Alice computes transaction hash e = SHA256(m), chooses a random ephemeral integer k, and calculates signature pair (r, s) using her private key d: r = (k * G)_x mod n, and s = k^(-1) * (e + d * r) mod n.

- **2. Verification Process (Network Nodes):** Any validator node verifies validity using Alice's Public Key Q without ever seeing her private key d: calculates w = s^(-1) mod n, u_1 = e * w mod n, u_2 = r * w mod n, and checks if (u_1 * G + u_2 * Q)_x mod n == r. If true, the transaction is authentic!

1.3 Merkle Trees & SPV Verification

Definition: Merkle Tree (Ralph Merkle, 1979)

A Merkle Tree is a complete binary tree of cryptographic hashes where every Leaf Node represents the hash of an individual transaction: H_i = SHA256(Tx_i), and every Non-Leaf Internal Node represents the cryptographic hash of its concatenated child node hashes: H_{AB} = SHA256(H_A || H_B). The top root node is the Merkle Root.

BINARY MERKLE TREE ARCHITECTURE IN A BLOCK

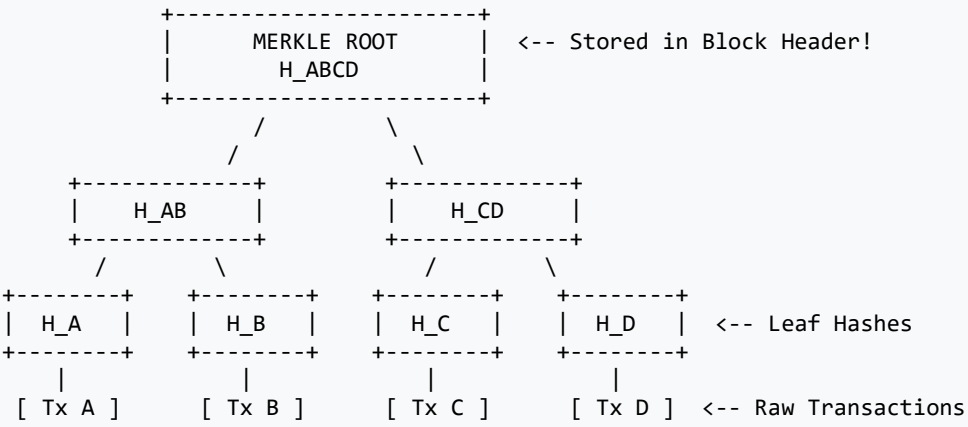

Figure: Hierarchical cryptographic hashing structure allowing O(log N) transaction verification

Simplified Payment Verification (SPV) & Merkle Proofs

- **The Problem:** Full blockchain nodes store hundreds of gigabytes of historical ledger data. Mobile and lightweight IoT wallets (SPV clients) cannot download the entire blockchain.
- **The Solution (Merkle Proofs):** An SPV client downloads ONLY the 80-byte block headers (containing the Merkle Root). To verify if transaction Tx_C is included in a block with N transactions, the client requests a Merkle Audit Path containing only log_2(N) sibling hashes (e.g., H_D and H_AB).
- **Verification Complexity:** Verifying a transaction in a block of 4,096 transactions requires only log_2(4096) = 12 hash operations (O(log N)) instead of downloading all 4,096 transactions (O(N))!

1.4 Blockchain Fundamentals & Architectural Layers

1. Limitations of Centralized Systems vs. Decentralization

Comparison Parameter	Centralized Systems (Traditional Banks / Cloud)	Decentralized Blockchain Systems (DLT)
Point of Failure	Single Point of Failure (SPoF); server outages crash the entire network.	Zero SPoF; thousands of distributed P2P nodes maintain redundant copies.
Trust Model	Requires blind trust in central intermediary (Bank, Government, AWS).	Trustless: consensus enforced mathematically by deterministic protocol code.
Data Immutability	Database administrators can alter, rollback, or censor records.	Immutable: once confirmed, cryptographically linked blocks cannot be altered.
Transparency & Audit	Opaque; private internal ledgers inaccessible to the public.	100% transparent; all transactions are publicly auditable in real time.

2. The Six Architectural Layers of Blockchain

- **1. Application & Presentation Layer:** User-facing DApps, decentralized exchanges (DEX), wallet software interfaces (MetaMask), and Web3 browsers.
- **2. Execution / Smart Contract Layer:** Turing-complete virtual machines (Ethereum Virtual Machine EVM, Wasm, Chaincode) executing programmable contract bytecode.
- **3. Data Layer:** Underlying physical data structures: cryptographically linked blocks, Merkle Patricia Trees, UTXO state pools, digital signatures, and asymmetric key pairs.
- **4. Network / P2P Layer:** Peer-to-peer gossip communication protocols, node discovery, transaction propagation, and mempool synchronization across distributed nodes.
- **5. Consensus Layer:** The foundational consensus engine governing decentralized agreement (Proof of Work, Proof of Stake, PBFT, Raft) and fork resolution rules.
- **6. Infrastructure / Hardware Layer:** Physical mining rigs, validator server nodes, CPU/GPU/ASIC chips, and data center network backbones.

1.5 Types of Blockchain: Public, Private & Consortium

1. Public (Permissionless) Blockchain

Public blockchains are completely open, decentralized networks where anyone can join, read transactions, submit transactions, and participate in consensus without seeking permission from any entity. Examples: Bitcoin, Ethereum, Litecoin, Solana.

2. Private (Permissioned) Blockchain

Private blockchains are closed, centralized networks operated and strictly governed by a single organization or enterprise. Read and write permissions are tightly restricted to authenticated identity holders. Examples: Hyperledger Fabric, MultiChain.

3. Consortium (Federated) Blockchain

Consortium blockchains are semi-decentralized networks governed collectively by a pre-selected group of multiple enterprise organizations (e.g., a consortium of 10 international banks). Examples: R3 Corda, Quorum, B3i.

Master Comparison Table: Public vs. Private vs. Consortium

Comparison Dimension	Public Blockchain (Permissionless)	Private Blockchain (Permissioned)	Consortium Blockchain (Federated)
Access & Participation	Completely Open to the public; anonymous.	Strictly Restricted to single organization.	Restricted to pre-selected consortium members.
Consensus Mechanism	PoW, PoS (Economic incentives required).	Crash Fault Tolerance (Raft, Paxos, PBFT).	BFT, Raft, IBFT (No mining incentives needed).
Transaction Throughput	Low (7 to 50 TPS; high latency).	Very High (1,000 to 10,000+ TPS; instant).	High (500 to 5,000 TPS; near-instant).
Immutability & Security	Nearly impossible to tamper (requires 51%).	Can be modified if central admin decides.	Can be modified if consortium majority colludes.
Energy Consumption	High (PoW) to Moderate (PoS).	Negligible (Zero mining energy wasted).	Negligible (Lightweight consensus).
Primary Use Cases	Cryptocurrencies, DeFi, NFTs, Public Web3.	Internal supply chain, corporate audit, EHR.	Inter-bank cross-border settlement, trade finance.

1.6 High-Yield University Exam Question Bank

Part A: Short Answer Questions (2 to 5 Marks)

Q1. What is an Elliptic Curve and why is ECC preferred over RSA in blockchain? [3-4 Marks]

- **Answer:** ECC is based on elliptic curves y^2 = x^3 + ax + b mod p and the Discrete Logarithm Problem (ECDLP: Q = d*G is one-way). ECC is preferred because a 256-bit ECC key provides identical security to a 3072-bit RSA key, saving bandwidth, storage, and processing power on resource-constrained blockchain nodes.

Q2. Explain the Merkle Tree and how it enables SPV (Light Node) verification. [4 Marks]

- **Answer:** A Merkle Tree is a binary hash tree summarizing all transactions into a single top-level Merkle Root. Light nodes (SPV) verify transaction inclusion in O(log N) time using only a Merkle proof path of sibling hashes without downloading full blocks.

Q3. Differentiate between Public and Private blockchains. [3 Marks]

- **Answer:** Public blockchains are open, decentralized, permissionless networks (Bitcoin, Ethereum) using PoW/PoS with lower throughput. Private blockchains are closed, permissioned networks controlled by a single entity (Hyperledger) with high throughput and instant BFT finality.

Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Explain the cryptographic primitives of blockchain: Symmetric vs Asymmetric cryptography, SHA-256 properties, and ECDSA digital signatures. [12 Marks]**
- **Q5. Describe the structure of a Merkle Tree with neat diagrams. Explain step-by-step how a Merkle Proof is constructed and verified. [10 Marks]**
- **Q6. Detail the Six Architectural Layers of Blockchain from hardware to application layer with protocols and functions. [12 Marks]**
- **Q7. Compare Public, Private, and Consortium blockchains across governance, consensus, throughput, scalability, and enterprise use cases. [12 Marks]**

1.7 Unit Summary & Key Takeaways

- Blockchain combines distributed P2P networking, cryptographic hash chains, and consensus algorithms to create tamper-proof decentralized ledgers.
- Asymmetric cryptography (ECC secp256k1) generates public-private key pairs, while ECDSA digital signatures guarantee authentication and non-repudiation.
- SHA-256 cryptographic hashing provides pre-image resistance, collision resistance, and the avalanche effect.

- Merkle Trees organize transaction hashes hierarchically, enabling efficient O(log N) SPV Merkle verification.
- Blockchain architecture comprises 6 layers: Application, Execution/Contracts, Data, Network, Consensus, and Infrastructure.
- Blockchain networks are classified as Public (permissionless), Private (permissioned enterprise), or Consortium (federated multi-organization).