# Unit-2

Bitcoin's blockchain technology enables decentralized digital currency through cryptographic security, peer-to-peer networking, and consensus mechanisms. Below is a structured analysis of Bitcoin's core components and processes.

## Creation of Coins

New bitcoins are generated exclusively through **mining**, a computational process where nodes (miners) compete to solve cryptographic puzzles. Key aspects include:

- **Fixed emission**:
  🔢 **Fixed Supply (21 Million Coins)** Bitcoin has a hard limit: there will *never* be more than **21 million bitcoins** in existence. This limit is built right into its code to avoid inflation—sort of like having a fixed amount of gold in the world.
  ⛏️ **Block Rewards and Halving** When new transactions are added to the blockchain, miners get rewarded with some newly created bitcoins. But here's the twist: **every 210,000 blocks** (which takes about **4 years**), that reward gets **cut in half**. This is called a **halving**.

  So, the reward started at **50 bitcoins per block** in 2009, then dropped to 25, 12.5, 6.25, and so on. Eventually, the rewards get so tiny they basically stop—and no new bitcoins are created after that. This keeps the supply limited and valuable.

- ### ⛏️ Bitcoin Mining Process

  1. **Transaction Validation** Miners gather unconfirmed transactions from the mempool and check:
     - Are the digital signatures valid?
     - Are the inputs unspent?
     - Is the sender’s balance sufficient?
  2. **Block Assembly** Valid transactions are bundled into a candidate block. The miner also includes:
     - A reference to the previous block’s hash
     - A timestamp
     - A **Merkle root** (a single hash summarizing all transactions)
     - A **nonce** (a number that miners tweak to find a valid hash)
  3. **Proof-of-Work Puzzle** The miner hashes the block header using **SHA-256**. The goal is to find a hash that’s **below the network’s difficulty target**.
     - This requires trying **trillions of nonce values**.
     - The process is computationally intensive and random.
  4. **Success and Reward** The first miner to find a valid hash:

        - Broadcasts the block to the network
        - Receives the **block reward** (currently 6.25 BTC)
        - Collects **transaction fees** from the included transactions

### 🔐 SHA-256 Hash Function in Mining

- **SHA-256** (Secure Hash Algorithm 256-bit) takes any input and produces a **fixed 256-bit output**.
- Even a tiny change in input (like the nonce) gives a completely different hash—this is called the **avalanche effect**.
- Bitcoin uses **double SHA-256** for added security:

  ```
  hash = SHA256(SHA256(block_header))
  ```

This hashing process ensures that mining is **fair, secure, and tamper-resistant**. It's what makes Bitcoin's decentralized consensus possible.

## Payments and Double Spending

### 💸 How Bitcoin Prevents Double-Spending

Double-spending is when someone tries to spend the same bitcoin in more than one place—kind of like trying to copy a digital file and pass it off as unique money. Bitcoin prevents this using two powerful mechanisms:

#### 1. Blockchain Immutability

Once a transaction is confirmed and included in a block, it's locked into the blockchain using **cryptographic hash links**. Changing even a tiny detail in one block would break the link for all blocks that come after it. To fake a transaction, someone would need to re-mine all subsequent blocks **faster than the entire global Bitcoin network**—which is practically impossible.

#### 2. Consensus Verification

Every transaction is:

- **Broadcast** to the entire network,
- **Validated** independently by multiple nodes, and
- **Confirmed** by miners solving cryptographic puzzles (Proof of Work).

After **6 confirmations** (i.e., 6 more blocks added after the one holding the transaction), the network treats the transaction as final—**like carving it in digital stone**.

## Bitcoin Scripts

### 🧮 What Is Bitcoin Script?

**Bitcoin Script** is a simple, *non-Turing complete* programming language used to control how bitcoins are spent. Instead of traditional programming logic, it works like a **stack-based interpreter**, similar to how a calculator processes inputs.

### 🔐 1. Locking Script (a.k.a. ScriptPubKey)

This is written by the **sender** and attached to the output of a Bitcoin transaction. It specifies the **conditions that must be met** to unlock and spend the funds.

Example:

```
OP_DUP OP_HASH160 <recipient_pubkey_hash> OP_EQUALVERIFY OP_CHECKSIG
```

This is a typical Pay-to-Public-Key-Hash (P2PKH) locking script—essentially saying: *"To spend this, you must prove you own the private key corresponding to this public key hash."*

### 🔒 2. Unlocking Script (a.k.a. ScriptSig)

This is provided by the **spender** when they want to use those funds. It contains the data needed to satisfy the locking script.

Example:

```
<signature> <public_key>
```

### 🔁 3. Script Execution

When a transaction is being verified, **both scripts are combined and executed by nodes**:

```
<signature> <public_key> OP_DUP OP_HASH160 <pubkey_hash> OP_EQUALVERIFY OP_CHECKSIG
```

Nodes process this like a checklist:

- Is the public key’s hash correct?
- Does the signature match?
- If **yes** to all—✅ transaction is valid!
- If **no**—⛔ rejected.

### 📌 Why It’s Powerful

- Supports **multi-signature wallets** (e.g., 2-of-3 signatures needed).
- Enables **time locks** (funds can’t be spent before a specific time).
- Can support **smart contract-like logic** (though limited).

### 🌐 Bitcoin P2P Network Explained

Bitcoin doesn’t rely on banks or central servers. Instead, it runs on a **peer-to-peer (P2P) network**—think of it like a global web of thousands of equal players all keeping watch.

### 🧱 1. Flat Topology: Everyone’s Equal

- Each **node** (a computer running the Bitcoin software) is an *equal participant*.
- No boss, no hierarchy. Every node can:
  - Validate transactions and blocks
  - Relay data to other nodes
  - Maintain a complete copy of the blockchain
- This promotes **decentralization** and fault tolerance—if some nodes fail or go offline, the network keeps humming.

### 🔌 2. What Nodes Actually Do

- Listen for new transactions and verify them.
- Store the history of all Bitcoin transactions ever made.
- Help **propagate blocks** and **enforce consensus rules**.

Think of it like a neighborhood where everyone has a full copy of the community records—and they compare notes to make sure no one cheats.

### ⚙️ 3. Extended Network (Stratum, Pools, Wallets)

- Some specialized components extend the core P2P layer:
  - **Mining pools** (via protocols like Stratum) aggregate hashing power to increase reward chances.
  - **Lightweight wallets** (like Electrum) rely on full nodes but allow users to transact securely without storing the entire blockchain.

## 🔁 Bitcoin Transaction Workflow (Simplified)

1. **Initiation** The sender (say, Alice) enters the recipient's Bitcoin address and the amount to send. This is like filling out a digital check.
2. **Signing** Alice uses her **private key** to sign the transaction. This proves she owns the bitcoins and authorizes the transfer—like putting a secure digital signature on the check.
3. **Broadcasting** The signed transaction is sent out to the **Bitcoin peer-to-peer network**, where thousands of nodes receive and relay it. Think of it as shouting the transaction across a global room of accountants.
4. **Validation** Miners and full nodes check:
   - Is the signature valid?
   - Does Alice have enough unspent bitcoins (UTXOs)?
   - Has this bitcoin already been spent?

   If everything checks out, the transaction is added to the **mempool** (a waiting area for valid transactions).
5. **Confirmation** A miner includes the transaction in a new block and solves a cryptographic puzzle (Proof of Work). Once the block is added to the blockchain:

    - The transaction is **confirmed**.
    - After **6 confirmations**, it’s considered irreversible.

## ⛏️ Block Mining in Bitcoin: Step-by-Step

1. 🧾 **Transaction Selection** Miners scan the **mempool** (a waiting area for unconfirmed transactions) and pick those with the **highest fees**—since they get to keep those fees as part of their reward.
2. 🌲 **Merkle Tree Construction** All selected transactions are **hashed** and organized into a **Merkle tree**.
    - This structure allows for efficient and secure verification of transactions.
    - The **Merkle root** (top hash) summarizes all transactions and is included in the block header.
3. 🔐 **Proof-of-Work (PoW)** Miners now race to solve a **cryptographic puzzle**:
    - They tweak a value called a **nonce** and hash the block header.
    - The goal is to find a hash **below a target difficulty**.
    - This process takes **trillions of attempts**—hence the energy cost.
4. 📢 **Block Broadcast** The first miner to find a valid hash:
    - **Broadcasts** the new block to the network.
    - Other nodes **verify** it and, if valid, add it to their copy of the blockchain.

It’s like a global lottery where the ticket is computational effort—and the prize is both **newly minted bitcoins** and **transaction fees**.

## 🚀 Block Propagation and Block Relay in Bitcoin

### 📡 Block Propagation

This is the process of spreading a newly mined block across the **peer-to-peer (P2P) network**. Once a miner finds a valid block:

- It **broadcasts** the block to its connected peers.
- Those peers **validate** it and pass it along to their peers.
- This continues until the block reaches the entire network.

To make this fast and efficient, Bitcoin uses protocols like:

- **Compact Blocks (BIP 152)**: Instead of sending the full block, nodes send short transaction identifiers. Since most nodes already have the transactions in their mempool, they can reconstruct the block with minimal data transfer—**reducing latency and bandwidth**.

### ⚡ Block Relay Networks

These are **specialized high-speed networks** designed to **accelerate block propagation** even further. One example is:

- **FIBRE (Fast Internet Bitcoin Relay Engine)**: Developed by Bitcoin Core contributors, it uses **cut-through routing** and **UDP-based transmission** to relay blocks faster than the standard P2P method.

Why this matters:

- **Faster propagation** = **fewer orphaned blocks** (blocks that get rejected because another one reached the network first).
- It also **improves decentralization** by giving smaller miners a better chance to compete.

### 🧠 Distributed Consensus in Bitcoin

In a decentralized system like Bitcoin, there’s no central authority to decide which transactions are valid. So how does everyone agree on a single, shared version of the truth?

Bitcoin uses a clever combo of two key ideas:

#### ⛏️ 1. Proof-of-Work (PoW)

Miners compete to solve a **cryptographic puzzle**.

- Solving it takes **real computational effort** (and electricity), so it’s costly to cheat.
- The first miner to solve it gets to **add the next block** to the blockchain and earn a reward.
- This process acts like a **vote**—miners “vote” with their computing power.

> The more hash power you have, the more influence you have—but you still have to play by the rules.

#### 🔗 2. Longest Chain Rule

All nodes in the network follow a simple rule: > **“Trust the chain with the most accumulated work.”**

- If two versions of the blockchain exist, nodes choose the one with the **most total PoW** behind it.
- This ensures that **everyone eventually agrees** on a single version of history—even if temporary forks occur.

Together, these mechanisms make Bitcoin’s consensus:

- **Decentralized** (no central authority),
- **Robust** (resistant to tampering),
- And **self-correcting** (forks resolve naturally).

## Attacks on PoW and Monopoly Concerns

#### ⚠️ 1. 51% Attacks

If a single miner or mining pool controls **more than 50% of the total network hash rate**, they could:

- **Double-spend** coins by reversing transactions.

- **Censor** transactions by excluding them from blocks.
- **Fork** the chain and rewrite recent history.

This isn't just theoretical—there have been real incidents on smaller blockchains. For example, the GHash.io pool once briefly approached 51% of Bitcoin's hash rate in 2014, sparking major concern.

**Mitigation:**

- **Decentralized mining**: Encouraging a wide distribution of miners across the globe.
- **Economic disincentives**: A successful attack would likely crash Bitcoin's value, hurting the attacker too.

### 🏗️ 2. Centralization Risks

Even without crossing the 51% threshold, **mining pools** can concentrate power:

- A few large pools dominate Bitcoin mining.
- This undermines the decentralized ethos of blockchain.

**Solutions in progress:**

- **Pool hopping**: Miners switch pools to avoid centralization.
- **BetterHash protocol**: Lets individual miners choose which transactions to include, reducing pool operator control.
- **Relay networks** like FIBRE help smaller miners compete by speeding up block propagation.

Alternative Consensus Mechanisms

### 🔐 Proof of Stake (PoS)

- **How it works**: Validators are chosen to create new blocks based on how many coins they've staked (locked up as collateral).
- **Pros**: Energy-efficient, faster than Proof of Work (PoW), and encourages long-term commitment.
- **Cons**: Vulnerable to the *"nothing-at-stake"* problem—validators might support multiple chains during a fork since there's little cost to doing so.

### 🔥 Proof of Burn (PoB)

- **How it works**: Participants "burn" (permanently destroy) coins by sending them to an unspendable address. This act earns them the right to mine or validate blocks.
- **Pros**: Reduces total supply, mimicking scarcity like Bitcoin's halving.
- **Cons**: Destruction of value can be controversial, and it's less secure than PoW or PoS due to lower participation incentives.

### ⏱️ Proof of Elapsed Time (PoET)

- **How it works**: Each validator waits for a randomly assigned time. The one whose timer expires first gets to propose the next block.
- **Pros**: Energy-efficient and fair (randomized selection).
- **Cons**: Relies on **trusted hardware** (like Intel's SGX), which introduces centralization and trust assumptions.

## 🧑💻 Life of a Bitcoin Miner: From Setup to Survival

### ⚙️ 1. Setup

- Miners invest in **ASIC hardware**—specialized machines built solely for mining.
- They often **join mining pools** to combine computing power and increase chances of earning rewards.
- Mining software is installed to connect the hardware to the Bitcoin network and manage operations.

> Think of this as setting up your own digital gold-digging rig—except it hums with fans and eats electricity.

### 🔄 2. Operation

- ASICs run **24/7**, verifying transactions and racing to solve cryptographic puzzles.
- Miners must **monitor performance**, **cooling**, and **network difficulty**, which adjusts every ~2 weeks to keep block times steady.
- Downtime = lost revenue, so maintenance and uptime are critical.

> It's like running a high-stakes server farm where every second counts.

### 💸 3. Economics

- Income comes from:
  - **Block rewards** (currently 6.25 BTC per block, halving every ~4 years),
  - **Transaction fees** from included transactions.
- Expenses include:
  - **Electricity** (often 60–70% of total cost),
  - **Hardware depreciation**,
  - **Cooling and infrastructure**.
- If mining becomes unprofitable (e.g., due to high electricity or low BTC price), miners may **shut down or upgrade**.

> It's a constant balancing act between cost, competition, and crypto market swings.

## 🧱 Bitcoin Mining Difficulty: What It Means

**Mining difficulty** is like the dial that controls how hard it is to find the next block in the Bitcoin blockchain. It’s not fixed—it **adjusts every 2,016 blocks**, which is roughly **every 2 weeks**, to keep block creation steady at **one block every 10 minutes**.

### 🔁 Why It Adjusts

- If blocks are being mined **too quickly** (because more miners joined or hardware got faster), the network **increases the difficulty**.
- If blocks are coming in **too slowly** (because miners dropped off), it **lowers the difficulty**.

This keeps the rhythm of the blockchain consistent—like a metronome for decentralized finance.

### ⚙️ Impact of Higher Difficulty

- **More computational power** is needed to solve the cryptographic puzzle.
- It becomes **harder and more expensive** to mine a block.
- This **protects the network** from spam attacks and malicious actors, since attacking the chain would require enormous resources.

> Think of it like a self-adjusting safe: the more people try to crack it, the more complex the lock becomes.

## Mining Pools

- #### 🧑‍🤝‍🧑 What Are Mining Pools?

  Mining pools are **collaborative groups of miners** who combine their computational power (hash rate) to improve their chances of successfully mining a block. Instead of competing individually, they **work together** and **share the rewards**.

  #### 🧷 Key Concepts

  #### 1. Collaboration & Reward Sharing

  - Each miner contributes hash power to the pool.
  - When the pool successfully mines a block, the **block reward and transaction fees** are split among participants.
  - The split is **proportional** to each miner’s contribution—measured in “shares.”

  > Think of it like a team of treasure hunters splitting the gold based on how much digging each person did.

  #### 2. Pool Protocols (e.g., Stratum)

  - **Stratum** is the most widely used protocol for communication between miners and the pool.
  - It allows:
    - Efficient job assignment
    - Real-time updates

        - Secure submission of shares
    - Other protocols may include **getwork** (older) or **BetterHash** (more decentralized).

### ⚖️ Why Use a Pool?

- **More consistent payouts**: Solo mining is like buying a lottery ticket—you might win big, but it's rare. Pools offer **smaller, steady earnings**.
- **Lower variance**: Reduces the unpredictability of income.
- **Accessibility**: Even miners with modest hardware can earn rewards.