

MODULE 2

Blockchain technology is one of the most revolutionary innovations in the field of computer science and digital security. It solves the major problems of **trust**, **data tampering**, and **centralization** that exist in traditional systems.

Unlike conventional databases managed by a central authority, **blockchain** is a **decentralized digital ledger** that records transactions securely using **cryptography** (secret codes).

This ensures:

- **Integrity** – Data cannot be changed once entered.
- **Transparency** – Everyone in the network can view the transactions.
- **Immutability** – Data, once recorded, cannot be deleted or altered.

Each record in the blockchain is stored in a **block**, and these blocks are connected in a chain-like structure, hence the term “**Blockchain**.”

Blockchain Architecture refers to the **design and structure** of the blockchain system.

It is a **peer-to-peer (P2P) network** of computers (called **nodes**) that work together to store, verify, and share data without the need for a central authority.

Every node in the network has an identical copy of the ledger, ensuring **no single point of failure** and making the system **highly secure and reliable**.

Example:

Think of blockchain like an orchestra – every musician (node) plays their part independently, but together they create harmony (the synchronized ledger).

Key Features of Blockchain Architecture

1. Decentralization

- There is no central authority controlling the data.
- Every participant (node) has equal access and control.

- Reduces the risk of data loss and corruption.

2. Transparency

- All network participants can view transactions.
- Builds trust among users.

3. Security

- Data is secured using **cryptographic algorithms**.
- Each transaction is verified through digital signatures and hashes.

4. Immutability

- Once data is added, it cannot be changed or removed.
- Provides a permanent record of all activities.

5. Consensus Mechanism

- Ensures that all nodes agree on the validity of a transaction before adding it to the blockchain.
- Common algorithms: **Proof of Work (PoW)** and **Proof of Stake (PoS)**.

Key Components of Blockchain Architecture

1) Nodes

Nodes are the basic units of a blockchain network. They are computers connected to the network that: (Ex :group chat on WhatsApp with 10 friends.)

- Store and update the ledger
- Share information with other nodes
- Run software to interact with the blockchain

2) Transactions

Transactions are records of value transfer (like digital currency) or updates in smart contracts. Each transaction is verified and stored on the blockchain.

(Ex : Library Book Lending)

3) Decentralized Ledger

A decentralized ledger is a shared database that contains all blockchain transactions. It starts from the genesis block and continues to the current block. (Ex : Google Sheets Shared Among Team Members)

- Each block is linked using cryptographic hashes
- Blocks also contain transaction data and a timestamp, making them immutable.

4) Blocks

A block is a collection of transactions that are verified and added to the ledger. It is the fundamental building block of the blockchain. (Ex : physical ledger book in a shop)

5) Consensus Protocols

Consensus protocols are rules that guide how nodes agree on the validity of transactions.

- Examples include Proof of Work (PoW) , Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT)
- a) Proof of Work (PoW)
 - Used in Bitcoin.
 - Nodes (miners) solve complex mathematical puzzles to validate transactions.
 - The first to solve it adds the block to the blockchain and earns a reward.
 - Ensures security but requires high energy.

b) Proof of Stake (PoS)

- Used in Ethereum 2.0.
- Nodes (validators) are chosen based on the amount of cryptocurrency they hold and stake. (Ex: classroom where students count votes for a game:)
- Lower energy use compared to PoW and still ensures consensus and security.

c) Practical Byzantine Fault Tolerance (PBFT)

- Used in permissioned blockchains (private networks).
- Nodes communicate to reach agreement even if some nodes act maliciously. (Ex : group of kids racing to solve a riddle.)

- Ensures fast and reliable transaction validation.

Real-Life Example: Voting in a Committee

- Imagine a committee of 10 people deciding on a project:
 1. Everyone reviews the proposal (like nodes verifying transactions).
 2. They vote according to predefined rules (like PoW or PoS).
 3. The decision is accepted only if a majority agrees (like consensus in blockchain).
 4. Even if a few members try to cheat, the process ensures a fair outcome (similar to PBFT).
 5. (PoW = “Work hard with computers to win”, PoS = “Lock your coins and get randomly picked”)

Key Point:

- Consensus protocols are essential for trust and coordination in a decentralized network, ensuring all nodes agree on the same valid and immutable blockchain.

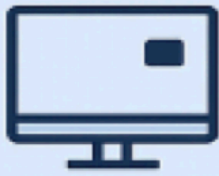
6) Miners and Validators

- Miners (PoW) use computing power to verify transactions and add them to the blockchain, earning rewards for their work
- Validators (PoS) stake their assets to verify transactions, without heavy computation
- Both ensure that only legitimate transactions are added.

7) Cryptography

Cryptography ensures security and authenticity:

- Public Key: Works like a wallet address to receive funds (Ex: Your mailbox)
- Private Key: Acts like a password to access funds and authorize actions (Ex: The key to your mailbox)
- It secures data and creates a trustless system, meaning you don't need to trust a central authority.



Nodes

Computers connected to the network that store and update the ledger



Transactions

Records of value transfer or smart contract updates



Decentralized Ledger

A shared database that stores all blockchain transactions



Blocks

A collection of transactions that are verified and added to the ledger



Consensus Protocols

Rules that guide how nodes agree on the validity of transactions



Miners and Validators

Entities responsible for verifying transactions

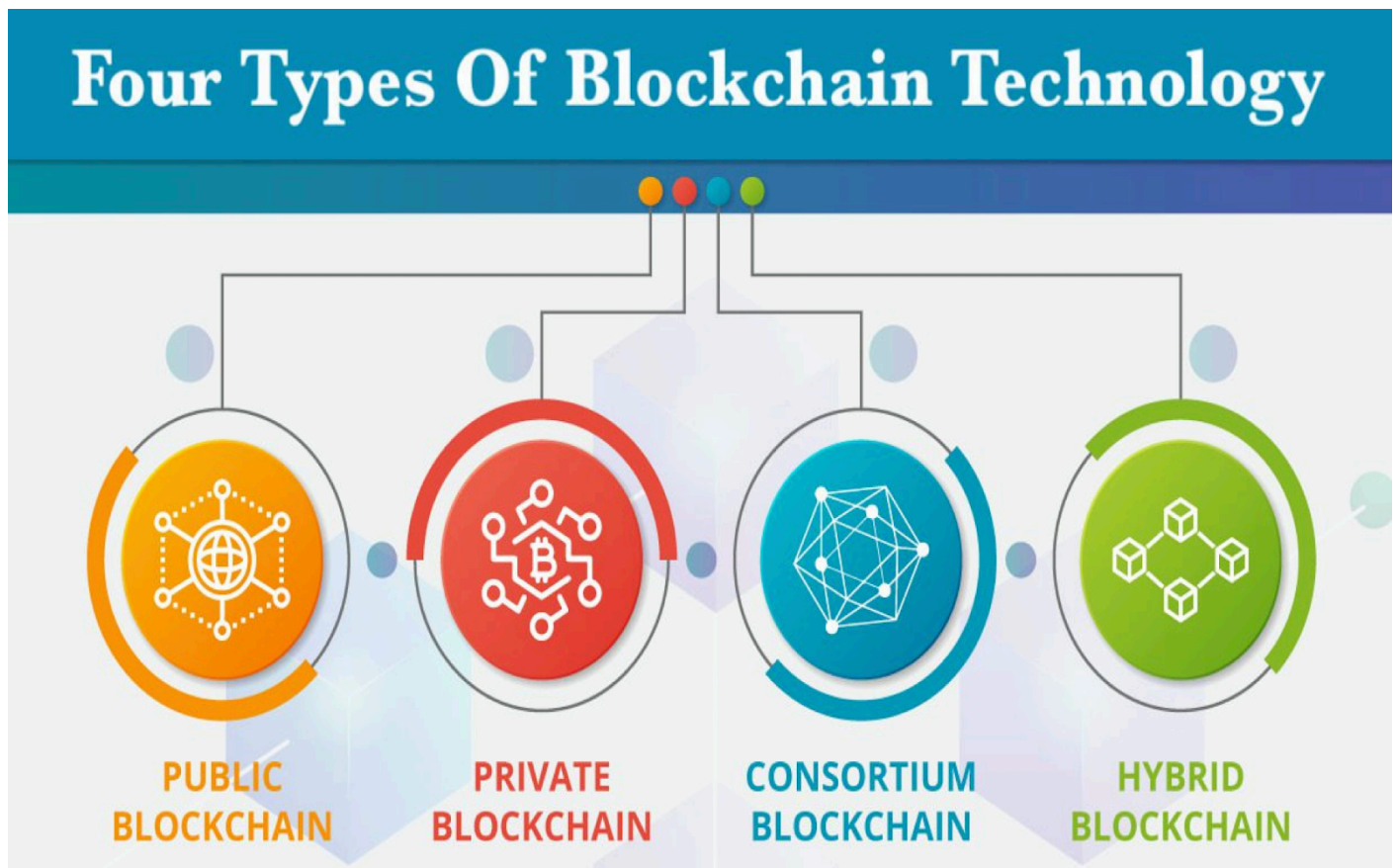


Cryptography

Ensures security and authenticity through public and private keys

Therefore,

- Blockchain architecture = decentralized, immutable, cryptographically secure system.
- Components: Block (data, hash, prev hash), Nodes, Consensus, Proof-of-Work.
- Benefits: Removes third parties, ensures trust, transparency, and security.



1. Public Blockchain

- A public blockchain is a completely open network where anyone can join without permission.
- All participants can read and write data, and transactions are validated by consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS).
- Since no single authority controls it, the network is highly decentralized and secure.
- Every transaction is transparent and visible to all, making it trustless.
- Example: Bitcoin and Ethereum are public blockchains where anyone can send or receive cryptocurrency. (Public notice board, voting system, food supply tracking)
- Use Case: Best suited for cryptocurrencies, voting systems, or public data sharing.

2. Private Blockchain

- A private blockchain is controlled by a single organization or authority.
- Only authorized members can participate, read, or validate transactions.
- Since the network size is small and controlled, transactions are faster and more efficient compared to public blockchains.
- However, because it relies on one entity, it loses the decentralization advantage. Example: Hyperledger Fabric – Used by businesses and banks for managing internal operations and secure transactions. (Banking and finance systems, Private healthcare record management)
- Use Case: Useful for internal organizational processes, banking systems, and private record management.

3. Consortium Blockchain

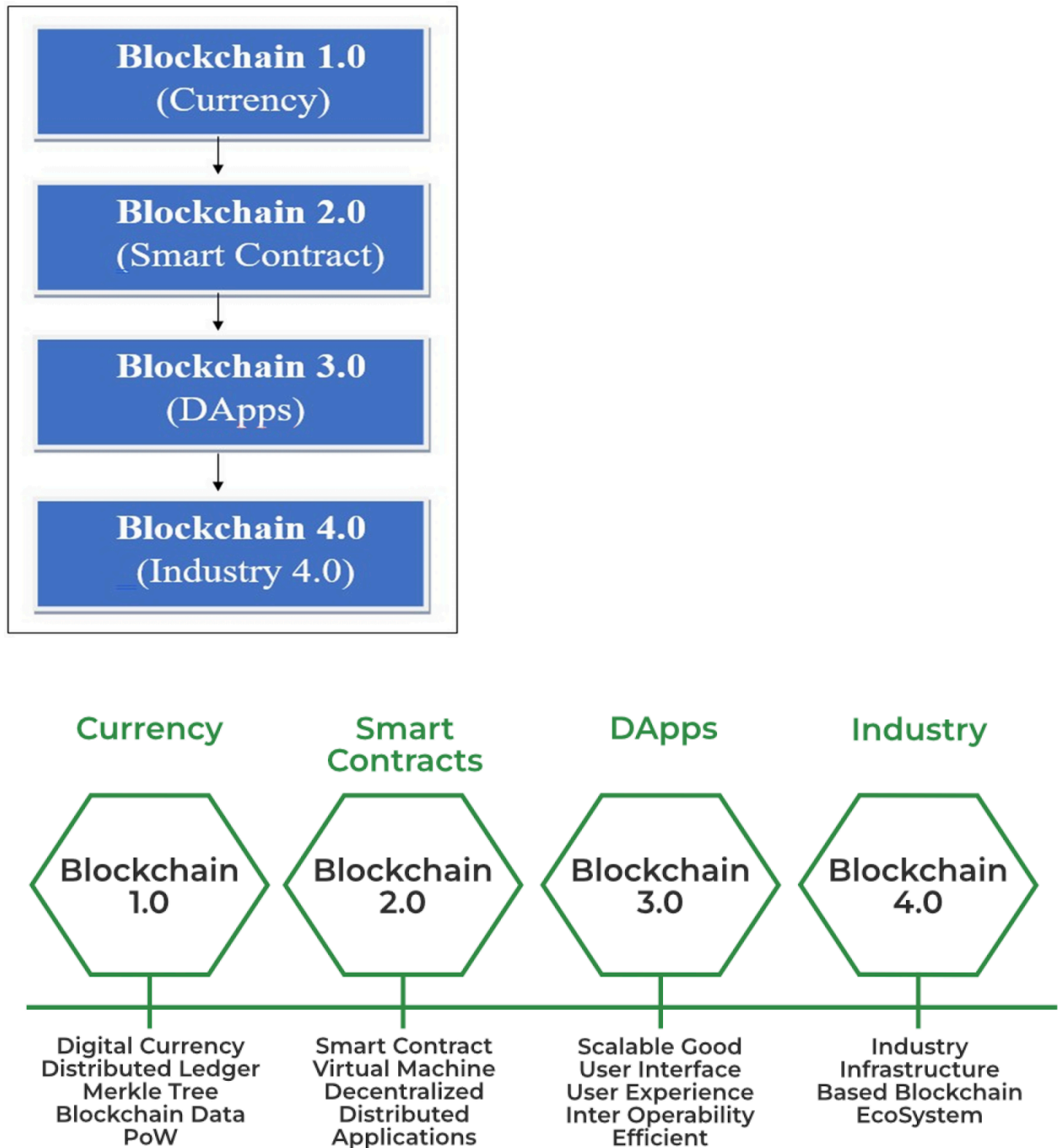
- A consortium blockchain (also called a federated blockchain) is governed by a group of organizations instead of a single entity.
- It offers more decentralization than private blockchains but not as much as public blockchains.
- Validation of transactions is done by selected members of the consortium.
- This ensures efficiency and shared responsibility while maintaining some level of trust and control.
- Example: R3 Corda used in financial institutions, Energy Web Foundation in the energy sector. (Supply chain management across companies)
- Use Case: Useful for industries where multiple companies need to collaborate, such as banking, supply chain, or healthcare.

4. Hybrid (Permissioned) Blockchain

- A hybrid blockchain combines the features of public and private blockchains.
- Some parts of the blockchain are open to the public, while others are restricted.
- This allows flexibility in balancing transparency with privacy.
- Organizations can decide which data should be public and which should remain private.
- This type is often called permissioned blockchain, since access can be customized.
- Example: IBM Food Trust.
- Use Case: Useful in government sectors, healthcare, or businesses that need both public accountability and private control.

Versions of Blockchain

There are four versions of Blockchain as depicted below :



1) Blockchain 1.0 - Bitcoin (Cryptocurrency) (Exchange of Money)

- Blockchain Version 1.0 was introduced in 2005 by Hal Finney, and it implemented Distributed Ledger Technology (DLT). This was the first practical use of blockchain for digital transactions.
- Purpose: It allows financial transactions to take place on a decentralized network without needing a central authority. These transactions are executed using Bitcoin, the first cryptocurrency.
- Permission Model: Blockchain 1.0 is permissionless, meaning anyone can join the network and perform valid transactions. There is no need for approval from a central organization.
- Main Usage: Its main application is in currency and payments, enabling peer-to-peer transfer of money without intermediaries like banks.
- Goals: The main aim was to create a transparent, publicly accessible, completely decentralized, and immutable ledger. It ensures a distributed system of transactions that is secure and visible to all participants.
- Foundation: Blockchain 1.0 is built on the idea and structure of Bitcoin. Its primary focus was on creating new cryptocurrencies and facilitating digital money transfer.
- Core Characteristics:
 - Functions as a digital, decentralized, distributed ledger.
 - Records transactions in a database shared by all nodes of the network.
 - Transactions are verified and updated by blockchain miners.
 - The ledger is maintained and monitored by everyone, with no individual ownership, ensuring transparency and trust.

2) Blockchain 2.0 - Ethereum (Evaluation of Smart Contracts)

1. Reason for Blockchain 2.0:
 - Blockchain 1.0 had some problems:
 - Mining Bitcoin was wasteful (high energy consumption).
 - Lack of scalability in the network.
 - Blockchain 2.0 was introduced to solve these issues and improve efficiency.
1. Beyond Cryptocurrency:
 - In Blockchain 2.0, the blockchain is not limited to cryptocurrencies.
 - It introduces Smart Contracts, extending blockchain functionality beyond simple transactions.

3) Smart Contracts:

- Smart contracts are like small computer programs that live in the blockchain.
- They are self-executing programs that automatically facilitate, verify, or enforce conditions defined in advance.
- They help reduce transaction costs and improve efficiency by removing intermediaries.

4) Ethereum as the Platform:

- In Blockchain 2.0, Bitcoin is replaced by Ethereum as the main platform for execution.
- Ethereum allows the processing of a high number of transactions rapidly on the public blockchain network.

5) Key Advantages:

- Automates agreements without third parties.
- Handles more complex transactions beyond simple currency transfer.
- Improves network efficiency and scalability compared to Blockchain 1.0.

3) Blockchain 3.0 (DApps)

1. Introduction:

- Blockchain 3.0 introduced Decentralized Applications (DApps) after Blockchain 2.0.
- A DApp is similar to a conventional app with a frontend written in any language, but its backend runs on a decentralized peer-to-peer network. (BitTorrent – a file-sharing platform.)

1. Decentralized Features:

- Uses decentralized storage and communication (e.g., Ethereum Swarm).
- No single owner or authority, ensuring:
 - Transparency
 - Improved security
 - Data accessibility for all
 - No censorship
 - Flexible development

Benefits of DApps:

- Zero downtime – always operational.

- Privacy and data integrity – users' data is secure.
- Trustless communication – transactions and business operations are secure without intermediaries.
- Examples of DApps: BitMessage, BitTorrent, Tor, Popcorn, etc.

* Advantages

- Transactions occur without third-party intermediaries, ensuring security of data.
- Blockchain uses cryptography to lock information securely.
- Eliminates double records, which accelerates transactions.

* Disadvantages

- Risk of human errors remains.
- Transaction costs (e.g., Bitcoin) can be high.
- Immutability – once data is inserted, it cannot be changed.
- Blockchain 4.0 is also called the Industrial or Enterprise Blockchain.
- It is designed to bring blockchain to real-world business and industrial applications.
- Focuses on creating scalable, fast, and cost-effective networks for large-scale use.
- Supports interoperability with existing enterprise systems and other blockchains.
- Provides enhanced privacy and security through permissioned networks.
- Enables applications in supply chain, healthcare, energy, IoT, and finance.
- Examples include IBM Blockchain, VeChain, and Ripple.
- Helps businesses maintain data integrity, transparency, and trust while reducing costs.

Use Cases of Blockchain

- Finance & Payments – cryptocurrencies, cross-border payments, remittances.
- Smart Contracts & DApps – automated agreements, decentralized applications, DAOs.
- Supply Chain & Logistics – product tracking, provenance, fraud prevention.
- Healthcare – secure sharing of patient records, interoperability.
- Voting & Governance – transparent, tamper-proof elections and e-governance.
- Digital Identity & Security – authentication, privacy protection.
- Energy & IoT – peer-to-peer energy trading, secure IoT communications.
- Intellectual Property – digital rights management and protection.

- Crowdfunding & Prediction Markets – transparent funding and forecasting systems.
- Insurance & Legal – smart claims processing, compliance, auditing.

| Blockchain | vs Centralized Database |
|---|---|
| There is no dependence on third parties. | Databases have central controls and administrators. |
| The data cannot be changed / deleted. | Authorized users can change / delete data. |
| Adding / removing parties; no change in system architecture is required. | Adding / removing parties; requires a change in system architecture. |
| Database management / maintenance costs are low. | Database management / maintenance costs are high. |
| High level verification is done with certificate verification. | User authentication; provided with username and password. |
| The process flow is determined without the need for changes in the system architecture. | Changing process flow requires a change in system architecture. |
| All users have «Open Ledger», where data is held. | The data are kept in a single centre. |
| It is compatible with the deed transfer process steps in the existing structure. | It is necessary to adapt the deed transfer process steps of the existing structure. |
| Users are provided to manage transactions in groups (Smart Contracts). | There is no structure like grouping transactions. |
| The blocks are stored with time stamp. | The timestamp can be added only manually. |
| Suitable where trust between parties is not required. | Central reliable authority is needed. |
| The process flow is kept together with the data in the blocks. | Process flow can be added manually with the logging mechanism. |

Introduction to crypto currencies



A cryptocurrency is a type of digital or virtual currency that uses cryptography (advanced encryption techniques) to secure transactions, control the creation of new units, and verify the transfer of assets.

Unlike traditional money, cryptocurrencies: (El savador)

- Exist only in digital form (no physical coins or notes).
- Are usually decentralized, meaning they are not issued or controlled by any central authority like a government or central bank.
- Operate on blockchain technology, which is a distributed ledger that records every transaction across a network of computers.
- This makes records transparent, tamper-proof, and secure, since no single person or authority can alter past transactions.
- Every participant in the network holds a copy of the ledger, ensuring trust and eliminating the need for a central authority.
- Because of these features, cryptocurrencies can be transferred directly between users (peer-to-peer) without the need for banks or intermediaries.

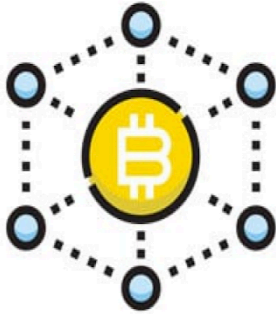
- In simple words:

Cryptocurrency is digital money, protected by cryptography, running on blockchain, and independent of banks or governments.

Cryptocurrency Market Activity

- Activity in cryptocurrency markets has increased significantly.
- Most interest is speculative → people buy cryptocurrencies mainly to make a profit, not primarily for payments.
- High volatility in prices:
 - Bitcoin: ~₹24,90,000 (mid-2021) → ~₹58,10,000 (end-2021) → ~₹29,05,000 (early-2022)
 - Ether and other cryptocurrencies showed similar price fluctuations.
- Mining & Security: Huge computing power is used to solve complex codes that protect the system from fraud or corruption.
- Despite growing interest, there is skepticism about whether cryptocurrencies can replace traditional payment methods or national currencies.

HOW ARE CRYPTO COINS CREATED?



In crypto mining, a computer solves complex numeric puzzles to verify blocks of cryptocurrency transactions and add them to the worldwide blockchain network.

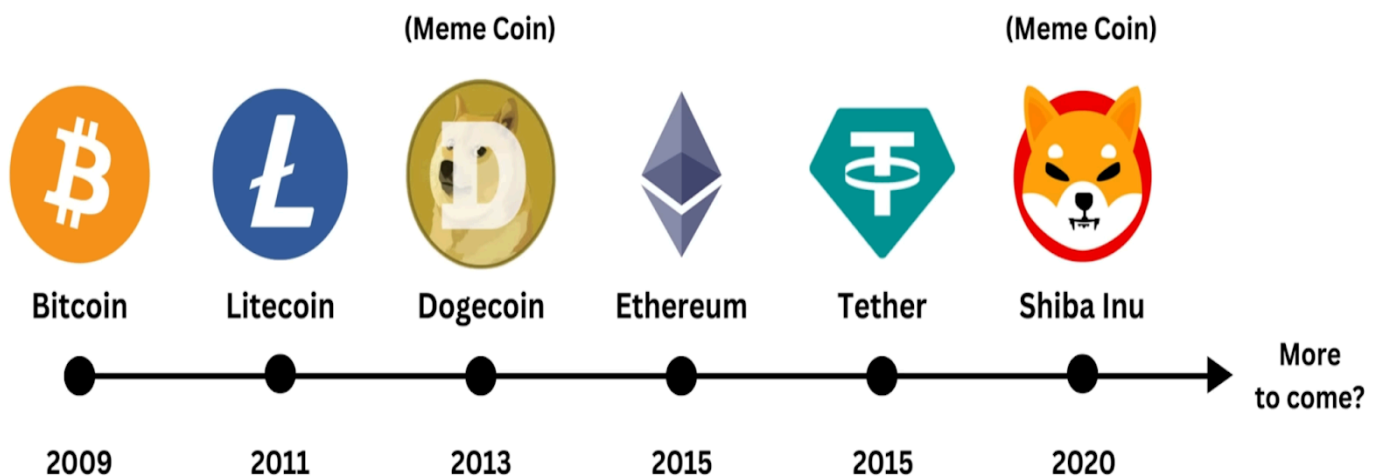


If the miner is the first to solve this puzzle before all the other computers on the network, the miner is rewarded with a cryptocurrency coin, which then enters into the worldwide supply.



In essence, miners are being offered a reward for doing the job of keeping the blockchain verified and secured, thus preventing fraud.

Cryptocurrency Coins History

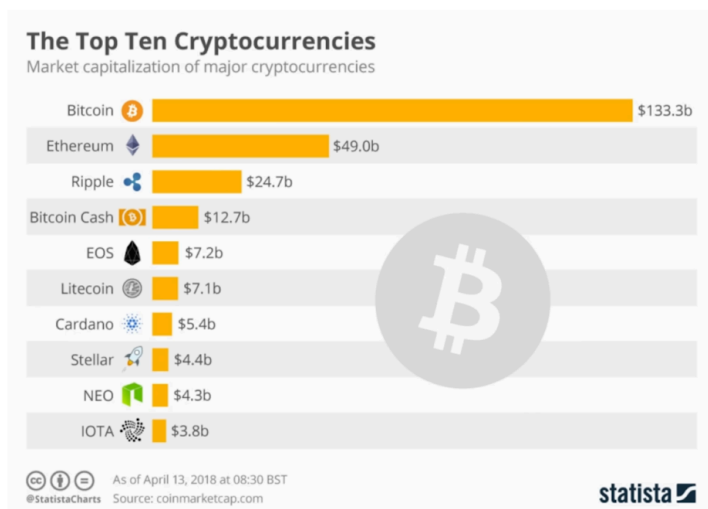


Types of Cryptocurrency

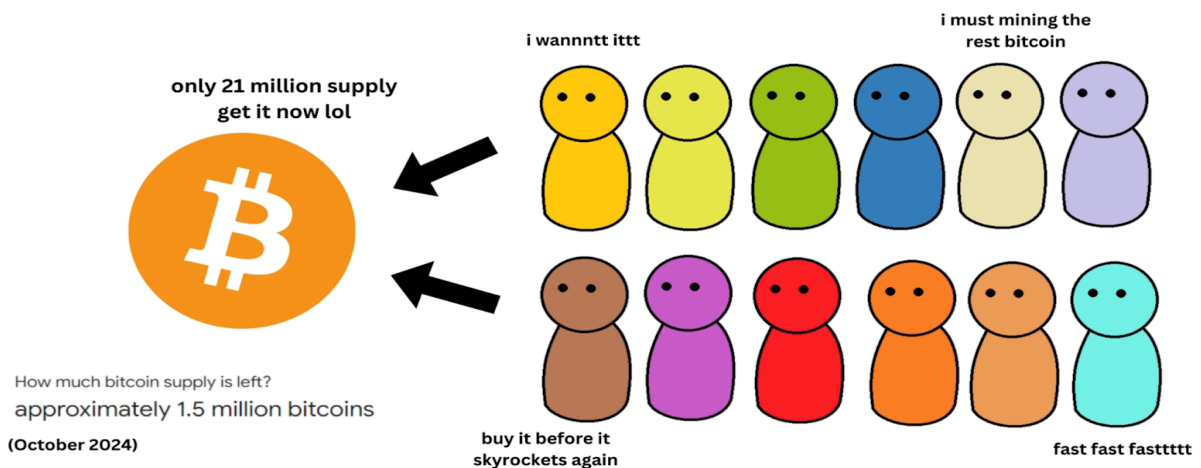
1. Bitcoin (BTC) – The Original Cryptocurrency

- Created in 2009 by Satoshi Nakamoto.
- First decentralized digital currency using blockchain technology.
- Acts as a peer-to-peer payment system without a central authority.
- Widely regarded as a store of value, sometimes called “digital gold.” (OG)
- Uses Proof of Work (PoW) to validate transactions. (Limited supply so valuable)

1. Bitcoin



1. Bitcoin



2. Altcoins (Alternative Coins)

- Cryptocurrencies other than Bitcoin.
- Designed to improve upon Bitcoin (faster transactions, lower fees, better scalability).

2. Altcoin

Alternative coins (Cryptocurrencies those **are not Bitcoin**)



Ethereum



Solana



Doge Coin



Shiba Inu

Meme Coins

a) Ethereum (ETH) / Smart Contract Platforms

- Introduced in 2015 by Vitalik Buterin.
- Goes beyond currency → supports smart contracts (self-executing agreements) and decentralized applications (DApps).
- Ether (ETH) is the cryptocurrency used to pay for computations on the Ethereum network.
- Enables the creation of Decentralized Finance (DeFi) platforms and NFTs.

b) Solana (SOL)

- Designed for speed and scalability.
 - Supports DApps, NFT platforms, and DeFi.
- Key Feature:
- Uses Proof of History (PoH) + Proof of Stake (PoS) for fast transactions.
- Speed: Handles up to 65,000 transactions per second.
- Example Use: Used in fast NFT marketplaces and Web3 apps.

c) Dogecoin (DOGE)

- Created as a fun cryptocurrency or “joke coin.”
- Gained real use for tipping, donations, and microtransactions.

Key Feature:

- Based on Litecoin’s blockchain, with unlimited supply.
- Strong community support and social media popularity.

Example Use: Donations, small online payments.

d) Shiba Inu (SHIB)

- Created as a community-driven meme token, inspired by Dogecoin.
- Aims to build its own decentralized ecosystem.

Key Feature:

- Operates on Ethereum, supports staking and NFT projects (ShibaSwap).

Nickname: “Dogecoin Killer.”

Example Use: Staking, NFT trading, and community-based projects.

Wallets

- A crypto wallet isn’t like a physical wallet that holds cash. Instead, it stores your private keys (and public address) which let you access and control your crypto on the blockchain.
- Simply put: funds (e.g., your Bitcoin or other crypto) exist on the blockchain; the wallet gives you access.
- Your public key / address is what you give someone to receive crypto. The private key is secret — whoever has it can move/transfer your crypto.

Types of Cryptocurrency Wallets

A **cryptocurrency wallet** is a digital tool that allows users to store, send, and receive cryptocurrencies such as Bitcoin, Ethereum, and others.

Instead of holding physical coins, a wallet stores **private and public keys** — unique cryptographic codes that provide access to digital assets recorded on the blockchain network.

Cryptocurrency wallets are broadly classified into **two types**, based on whether they are connected to the internet or not:

1. **Hot Wallets (Online Wallets)**
2. **Cold Wallets (Offline Wallets)**

1. Hot Wallets (Online Wallets)

Definition:

Hot wallets are those cryptocurrency wallets that remain **connected to the internet**. They are designed for quick access and convenience, allowing users to perform transactions easily and instantly.

Hot wallets are suitable for users who trade or transfer cryptocurrencies frequently. Since they are online, they can be accessed through devices such as mobile phones, computers, or web browsers. However, due to constant internet connectivity, they are **more vulnerable to security threats** like hacking, phishing, and malware attacks.

Key Characteristics:

- Always connected to the internet.
- Convenient for regular transactions.
- Suitable for beginners and active traders.
- Less secure compared to offline wallets.
- Usually free and easy to use.

Examples:

1. Mobile Wallets:

These are smartphone applications that allow users to manage and spend cryptocurrencies easily.

Examples: Trust Wallet, MetaMask (Mobile), Coinomi.

2. Web Wallets:

These wallets run on web browsers and can be accessed from any device connected to the internet.

Examples: Blockchain.com Wallet, Coinbase Wallet.

3. Exchange Wallets:

These are wallets provided by cryptocurrency exchanges for storing assets used in trading.

Examples: Binance Wallet, WazirX Wallet.

Advantages:

- Easy to set up and use.
- Provides instant access for transactions.
- Ideal for small, everyday transfers.

Disadvantages:

- High risk of online attacks and data theft.
 - Private keys may be controlled by third parties.
- Requires continuous internet connectivity.

2. Cold Wallets (Offline Wallets)

Cold wallets are cryptocurrency wallets that are **not connected to the internet**. They provide an **offline method** of storing cryptocurrencies, ensuring a high level of security against hacking and online threats.

Cold wallets are mainly used for **long-term storage** of cryptocurrencies. Since they are offline, hackers cannot access them remotely. They are preferred by investors who hold large amounts of digital currency for a long duration (commonly called “HODLers”).

Key Characteristics:

- Completely offline and secure.
- Immune to online hacking or malware attacks.
- Used for long-term or large-volume storage.
- Less convenient for frequent use.

Examples:

1. **Hardware Wallets:**

These are physical devices, like USB drives, that store private keys securely offline.

Examples: Ledger Nano S, Trezor, KeepKey.

2. **Paper Wallets:**

These are physical printouts containing the public and private keys or QR codes of a cryptocurrency wallet.

Examples: Generated using secure sites such as Bitaddress.org (used carefully).

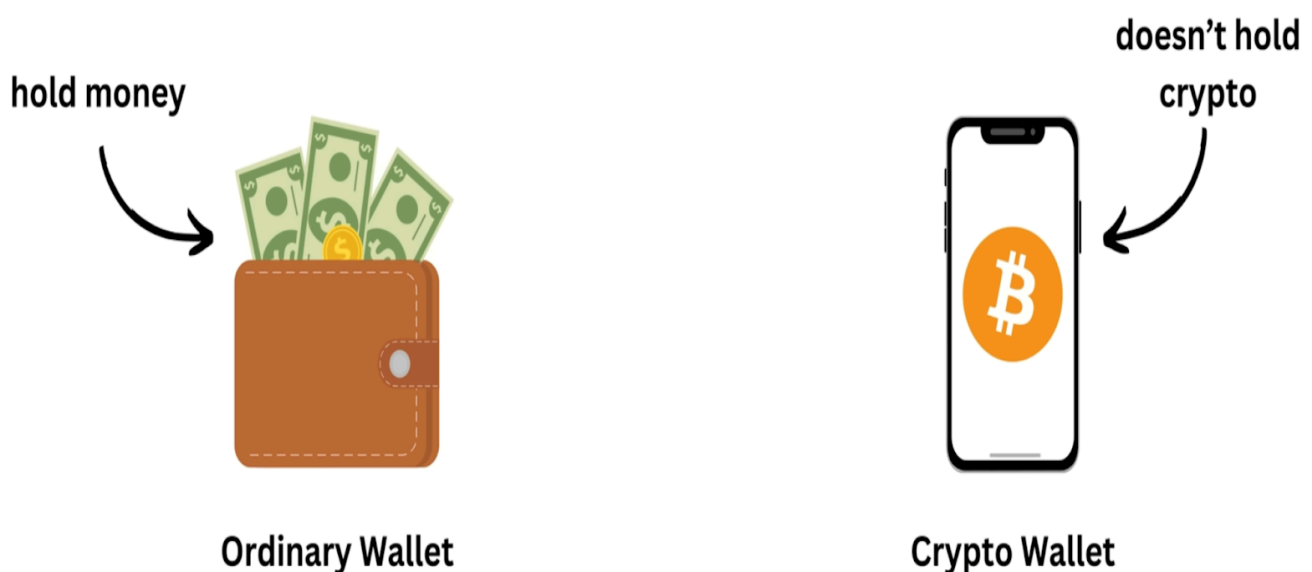
Advantages:

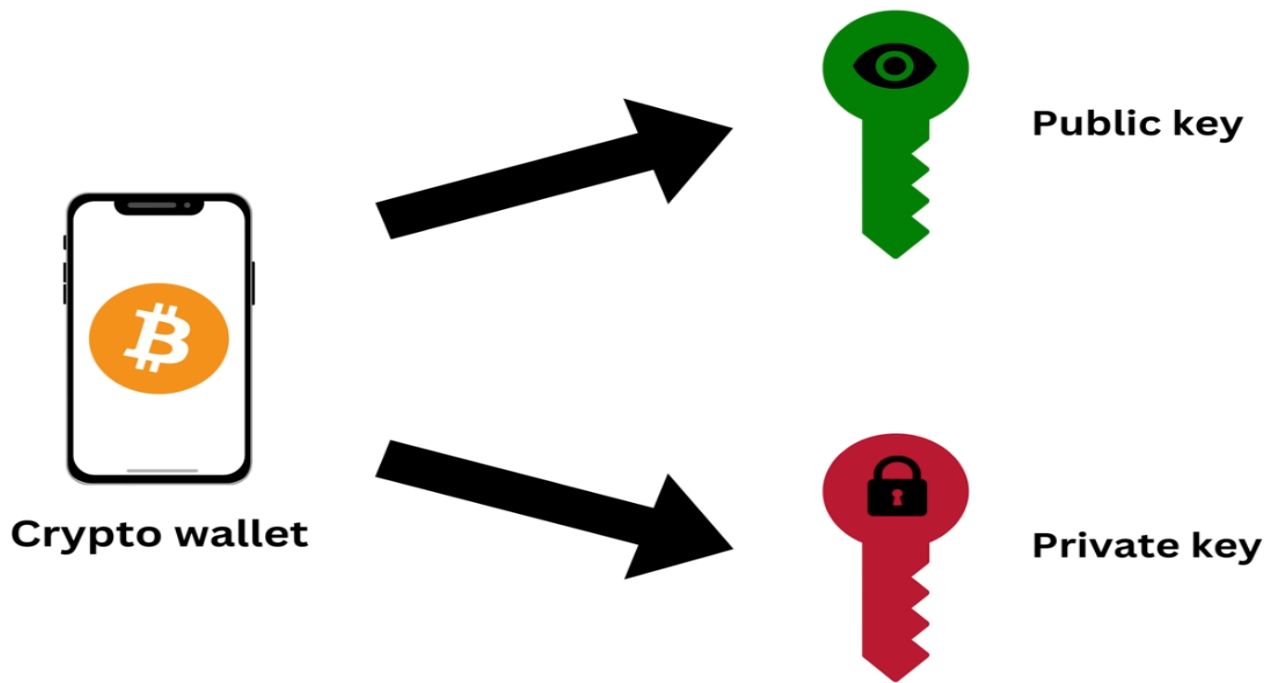
- Extremely safe from cyber threats.
- Best for long-term investment storage.
- Full control remains with the user.

Disadvantages:

- Inconvenient for frequent transactions.
- If lost or damaged, recovery is difficult.
- Physical protection (safe storage) is required.

Crypto wallet **doesn't hold any cryptocurrency**





Applications of Cryptocurrencies

1. Digital Payments

- Peer-to-Peer Transactions without banks or intermediaries.
- Low fees and faster cross-border payments (e.g., XRP for remittances).
- Useful in countries with unstable banking systems.

2. Investment & Trading

- Cryptocurrencies act as digital assets similar to stocks and commodities.
- Used for long-term investment (HODLing) or short-term trading.
- High volatility offers profit opportunities (but also high risk).

3. Decentralized Finance (DeFi)

- Borrowing, lending, and earning interest without banks.
- Platforms like Uniswap, Aave, Compound allow users to trade and earn passively.
- Provides financial services to the unbanked.

4. Smart Contracts & dApps

- Enabled by platforms like Ethereum, Solana, Cardano.
- Automates agreements (no middlemen).
- Used in insurance, real estate, supply chain, healthcare.

Cipher

A **Cipher** is a method or a set of mathematical rules used to **convert readable data (plaintext)** into **unreadable or coded data (ciphertext)** and vice versa.

It is a fundamental concept in **cryptography**, which ensures that information remains secure and confidential during transmission or storage.

A **cipher** is a system of algorithms or techniques used for **encryption** and **decryption** of data.

It transforms a message in such a way that only authorized parties can read it, while unauthorized individuals cannot understand it.

Basic Process:

There are two main operations in any cipher system:

1. Encryption:

- The process of converting **plaintext** (original readable message) into **ciphertext** (encoded or unreadable form).
- It uses a specific **key** and a **cipher algorithm**.
- Example: **HELLO** → **KHOOR** (using Caesar cipher with a shift of 3).

2. Decryption:

- The reverse process of encryption, where the **ciphertext** is converted back into **plaintext** using the correct **key**.
- Only the person who possesses the correct key can successfully decrypt the message.

Key Terminologies:

1. Plaintext:

- The original, human-readable message or data before encryption.
- Example: “HELLO WORLD”

2. Ciphertext:

- The encrypted, unreadable version of the plaintext.
- Example: “KHOOR ZRUOG” (after encryption using Caesar cipher).

3. Key:

- A piece of information (like a number, word, or code) used by the cipher to perform encryption and decryption.
- Without the correct key, decryption is impossible or very difficult.
- Example: In Caesar cipher, the key is the **number of shifts** (e.g., 3).

How Cipher Works (Simple Representation):

Plaintext + Key → [Encryption Process] → Ciphertext

Ciphertext + Key → [Decryption Process] → Plaintext

Purpose of Using Ciphers:

- To **ensure data confidentiality** (only authorized users can read it).
- To **protect sensitive information** during communication.
- To **secure passwords and digital transactions**.
- To **prevent unauthorized access** to private or government data.

Example: Caesar Cipher (Simple Substitution Cipher):

- Each letter in the plaintext is shifted by a fixed number of positions in the alphabet.
- **Example:** Shift = 3
 - Plaintext: HELLO
 - Ciphertext: KH00R

Here, each letter is replaced by the letter three positions ahead (H→K, E→H, L→O, O→R).

Classification of Ciphers:

Ciphers can be broadly classified into two main types:

1. Substitution Ciphers:

- Each character in the plaintext is replaced by another character or symbol.
- Example: Caesar Cipher, Monoalphabetic Cipher.

2. Transposition Ciphers:

- The positions of the characters are rearranged according to a specific rule, but the characters themselves remain unchanged.
- Example: Rail Fence Cipher, Columnar Transposition Cipher.

A **cipher** is the core mechanism in cryptography that helps protect data from unauthorized access.

It works by transforming readable information into coded form using **keys and algorithms**, ensuring **security, privacy, and confidentiality** in digital communication.

Would you like me to continue this section with “**Types of Ciphers (Substitution & Transposition with examples)**” as the next topic for your notes?

Cipher

A **cipher** is a method or set of mathematical rules used to **transform readable data (plaintext)** into **coded data (ciphertext)** and vice versa.

It is an important concept in **cryptography**, which ensures that information remains **secure and confidential** during communication or storage.

A **cipher** is an algorithm used for **encryption** and **decryption** of information.

It converts the original message into an unreadable form to protect it from unauthorized access.

Encryption and Decryption Process

1. Encryption:

- The process of converting **plaintext** into **ciphertext** using a cipher and a key.
- It hides the original message and makes it unreadable to anyone who does not possess the correct key.

- **Example:**

Plaintext: HELLO

Ciphertext: KHOOR (using Caesar cipher with a shift of 3)

2. Decryption:

- The process of converting **ciphertext** back into **plaintext** using the correct key.
- Only the authorized person who has the correct key can decrypt the data.

- **Example:**

Ciphertext: KHOOR

Plaintext: HELLO

Key Terminology

1. Plaintext:

The original, readable message or data before encryption.

Example: "HELLO WORLD"

2. Ciphertext:

The encrypted or unreadable version of the plaintext obtained after encryption.

Example: "KHOOR ZRUOG"

3. Key:

A secret value or code used by the cipher to perform encryption and decryption.

Without the correct key, the ciphertext cannot be properly decrypted.

Working Principle

Plaintext + Key \rightarrow [Encryption Process] \rightarrow Ciphertext

Ciphertext + Key \rightarrow [Decryption Process] \rightarrow Plaintext

This means that encryption and decryption are two opposite operations carried out using the same or related keys.

Purpose of a Cipher

- To ensure **data confidentiality** and privacy.
- To **protect sensitive information** during transmission.
- To **prevent unauthorized access** to data.
- To **secure passwords, transactions, and communications** in digital systems.

Example: Caesar Cipher

The **Caesar Cipher** is a simple substitution cipher where each letter in the plaintext is replaced by another letter that is a fixed number of positions away in the alphabet.

- **Example:**
Shift = 3
Plaintext: HELLO
Ciphertext: KHOOR

Here,

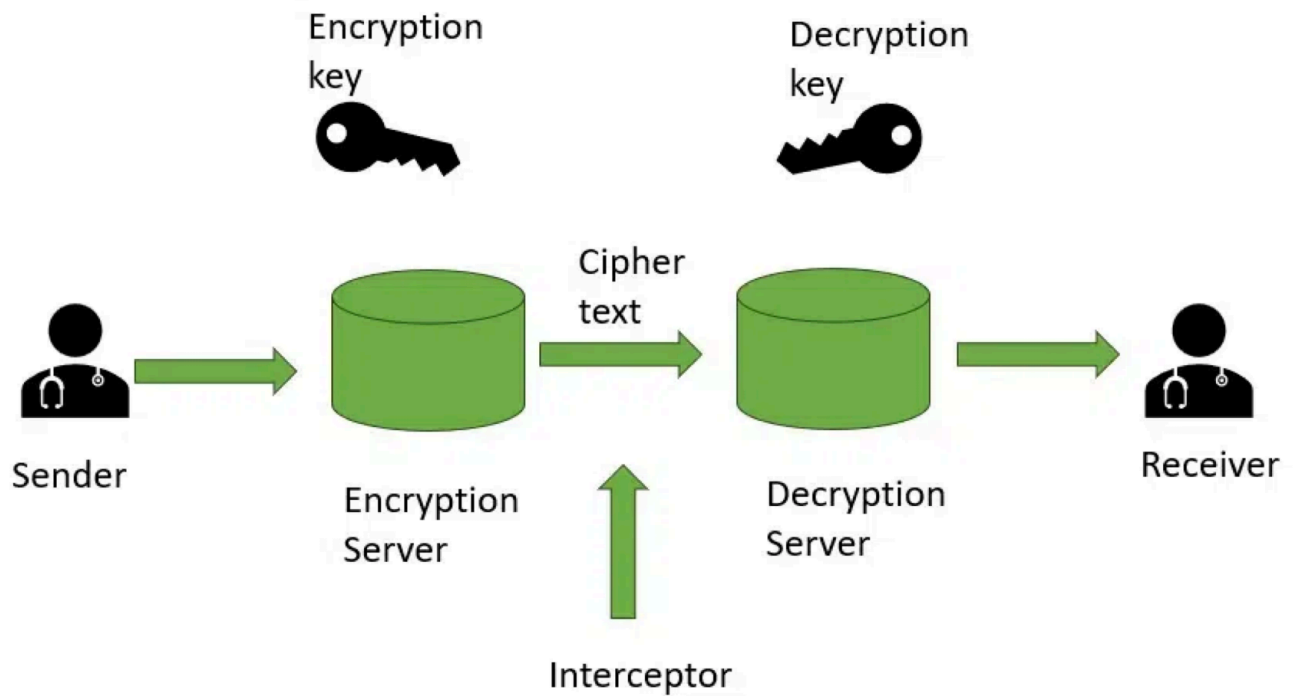
H \rightarrow K

E \rightarrow H

L \rightarrow O

O \rightarrow R

How Cipher Works



- The transformation (encryption) rearranges or alters plaintext into ciphertext under control of a key.
- Decryption requires the correct key, restoring the original message.
- Examples: traditional ciphers (Caesar, substitution, transposition) and modern cryptographic algorithms (AES, RSA).

Uses / Applications of Ciphers

- Secure communications (emails, instant messages, VoIP) so only intended recipient reads the message.
- Protecting financial transactions (online banking, e-commerce) to secure credit card info etc.
- Safeguard data storage (devices, servers, cloud). Even if hardware is lost/stolen, data stays safe if encrypted.
- Authentication: ensuring that communicating parties are who they claim to be; prevents fraudulent access.
- Digital signatures: use of encryption to ensure integrity and originality of documents/messages.

Types of Ciphers

1. Substitution Ciphers

- Replace each element of plaintext (e.g. letter) with another.

- Examples: Caesar cipher, simple substitution cipher

2. Transposition Ciphers

- Do *not* change the letters themselves, but change their positions.
- Examples: Rail Fence Cipher, Columnar Transposition Cipher.

3. Modern Ciphers

- More complex, designed for stronger security.
- Symmetric-key: same key for encryption & decryption. Example: AES.

Ciphers are fundamental to cryptography and information security. They evolved from simple ciphers (like Caesar) to complex modern ones.

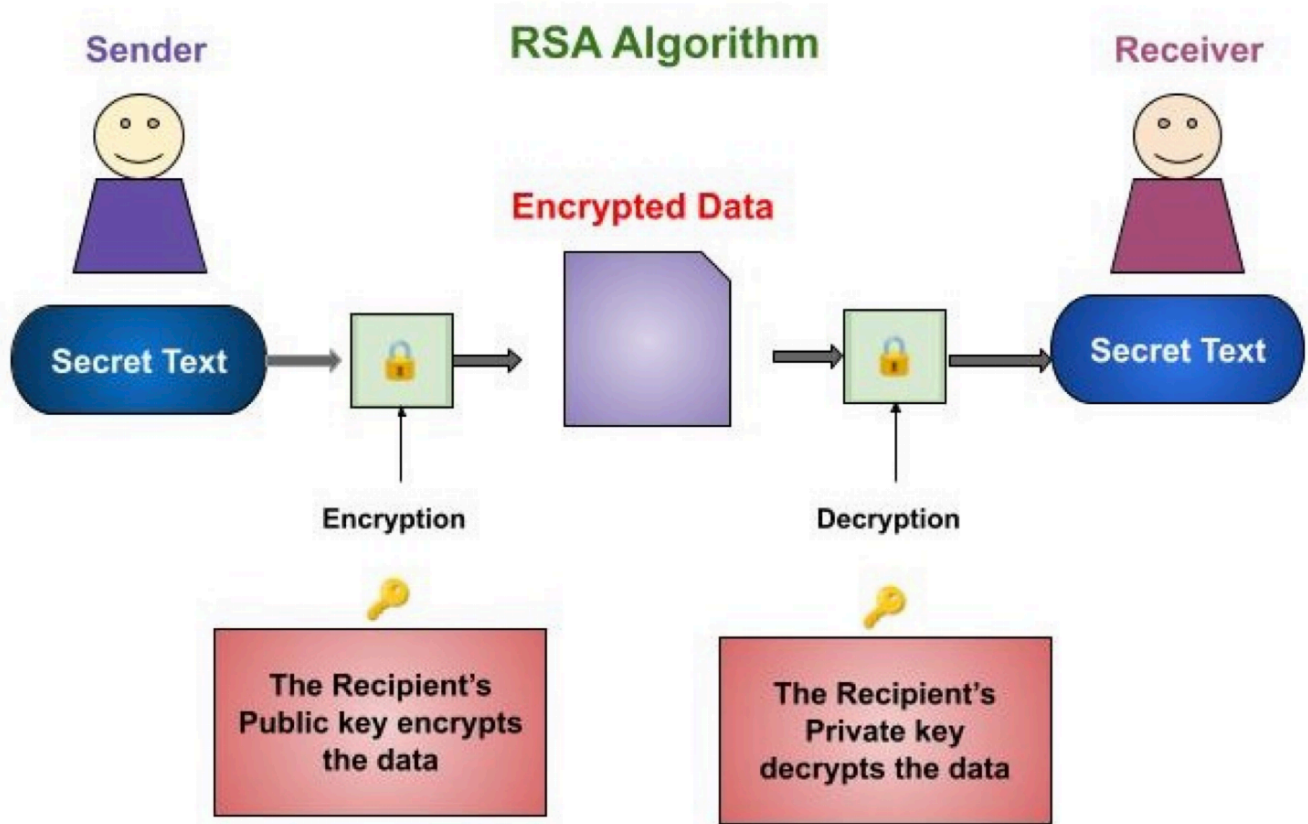
Understanding encryption, decryption, and how various ciphers work is essential for protecting data in the digital world.

RSA (Rivest–Shamir–Adleman) Algorithm

- RSA is an **asymmetric (public-key) cryptography** algorithm.
- It uses **two keys**:
 - **Public Key** → used for **encryption** (known to everyone).
 - **Private Key** → used for **decryption** (kept secret by the receiver).
- Developed in **1977** by **Ron Rivest, Adi Shamir, and Leonard Adleman**

Key Idea:

- **Public Key = Lock** (anyone can lock/encrypt the message).
- **Private Key = Key** (only the owner can unlock/decrypt the message).



How it works (Example)

1. Key Generation

- Person B generates a **Public Key** and a **Private Key**.
- Public Key is shared with everyone; Private Key is kept secret.

2. Encryption (Sender's Side – Person A)

- Person A wants to send a secure message to Person B.
- Person A **encrypts** the message using **Person B's Public Key**.

3. Decryption (Receiver's Side – Person B)

- Person B receives the encrypted message.
- Person B uses their **Private Key** to **decrypt** the message.

RSA Algorithm

1. Key Generation

- Pick two large prime numbers: **p** and **q** (keep secret).
- Compute **n** = **p** × **q** (part of both public & private keys).
- Compute Euler's totient: **Φ(n)** = (**p** − 1)(**q** − 1).
- Choose **e** (encryption key):
 - **1 < e < Φ(n)**
 - **gcd(e, Φ(n)) = 1** (must be co-prime).
- Find **d** (decryption key) such that:
 - **(d × e) mod Φ(n) = 1**
- **Public Key** = (**n**, **e**)
- **Private Key** = (**n**, **d**)

2. Encryption

- Convert message **M** into numbers (e.g., ASCII).
- Compute cipher text:
C = **M^e mod n**

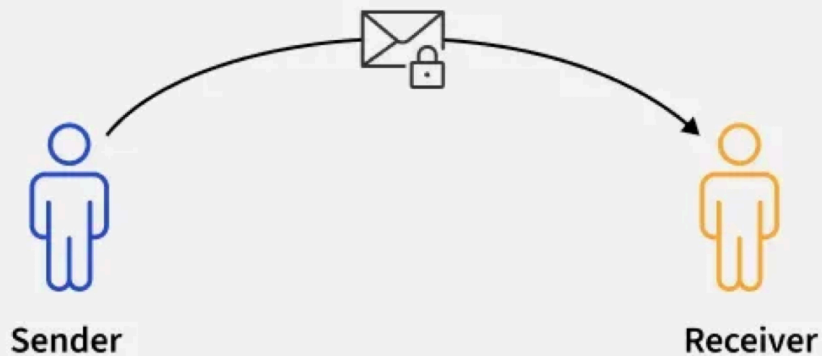
3. Decryption

- Use private key.
- Recover message:
M = **C^d mod n**

In short:

- **Public Key** → **Encrypts**
- **Private Key** → **Decrypts**

Secure communication between Sender and Receiver using RSA Algorithm



RSA Encryption and Decryption Example

01
Step

Key Generation

Choose two prime numbers: $p = 3, q = 11$

Calculate $n = p * q = 33$

Calculate Euler's Totient Function: $\Phi(33) = \Phi(3) * \Phi(11) = 2 * 10 = 20$

Choose $e = 7$, which is co-prime with 20

Calculate d as the multiplicative inverse of e (7), so $d = 3$

Public Key = $(n, e) = (33, 7)$ Private Key = $(n, d) = (33, 3)$

RSA Encryption and Decryption Example

02
Step

Sharing of Public Key

Public Key = $(n, e) = (33, 7)$ Private Key = $(n, d) = (33, 3)$



Sender

$(n, e) = (33, 7)$

The Public Key is shared with the Sender and the Private Key is kept secret with the Receiver .



Receiver

$(n, d) = (33, 3)$

RSA Encryption and Decryption Example

03
Step

Encryption - Message Conversion

Sender's Message(M) = "AC"



Numeric Conversion
(A - Z => 1 - 26)

13

Numeric Representation of "AC"

RSA Encryption and Decryption Example

04
Step**Encryption Formula:**

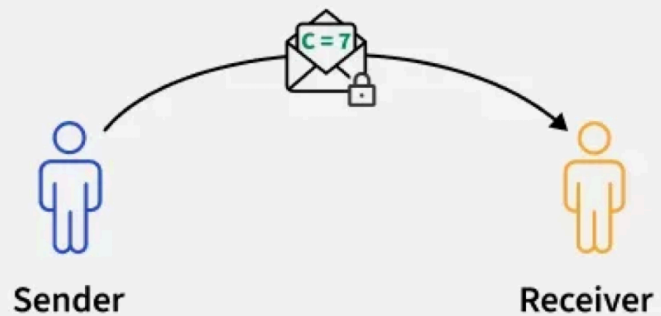
Encrypt the message using the Public Key (33, 7)

$$\text{Cipher Text } C = M^e \bmod n$$

$$C = 13^7 \bmod 33$$

$$C = 62748517 \bmod 33$$

$$C = 7$$



 RSA Encryption and Decryption Example

05
Step**Decryption Formula:**

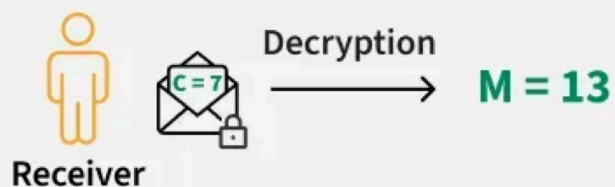
Decrypt the Cipher text using the Private Key (33, 3)

$$\text{Decrypted Text } M = C^d \bmod n$$

$$M = 7^3 \bmod 33$$

$$M = 343 \bmod 33$$

$$M = 13$$



The receiver uses decrypted text $M = 13$ to get the original message = "AC".

 RSA Encryption and Decryption Example
