BlockChain

Introduction

- Introduction to BlockChain
- Centralized v/s Decentralised Systeam
- Layers of Blockchain
- Advantages of Blockchain

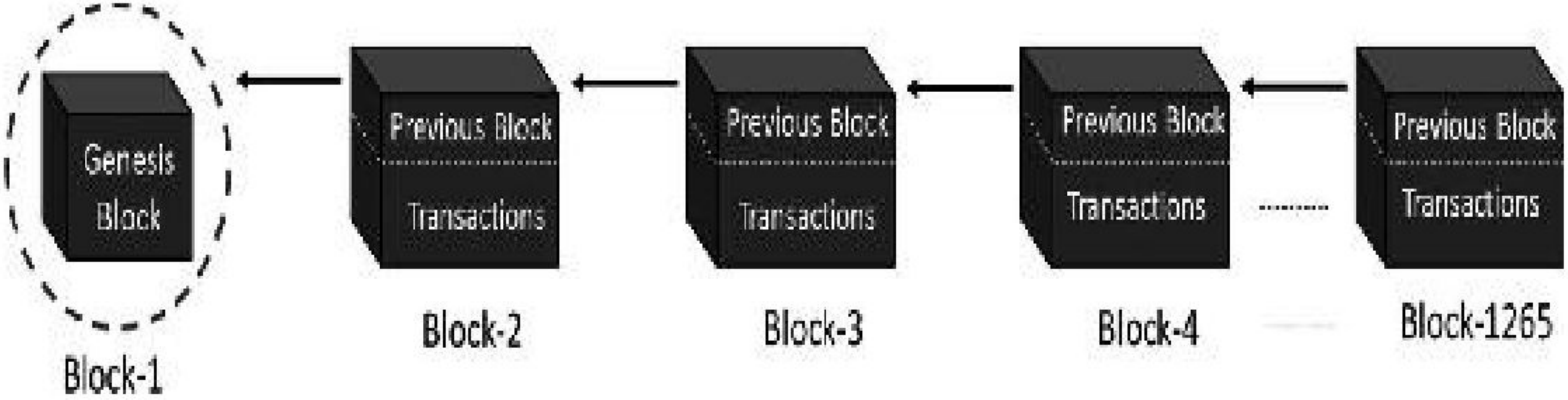
Blockchain Foundation

- Cryptography
  - -Symmetric Key Cryptography
  - -Asymmetric Key Cryptography
- Hash Function
- Game Theory
  - -Nash Equilibriam
- Prisoner’s Dilemma
- Byzantine Generals problem
- Zero sum Games
- Trees-Merkle Trees

Introduction

- Blockchain is a peer-to-peer system of transacting values with no trusted third parties in between.
- It is a shared, decentralized, and open ledger of transactions. This ledger database is replicated across a large number of nodes.
- This ledger database is an append-only database and cannot be changed or altered. It means that every entry is a permanent entry. Any new entry on it gets reflected on all copies of the databases hosted on different nodes.
- There is no need for trusted third parties to serve as intermediaries to verify, secure, and settle the transactions.
- It is another layer on top of the Internet and can coexist with other Internet technologies. Just the way TCP/IP was designed to achieve an open system, blockchain technology was designed to enable true decentralization.
- In an effort to do so, the creators of Bitcoin open-sourced it so it could inspire many decentralized applications.

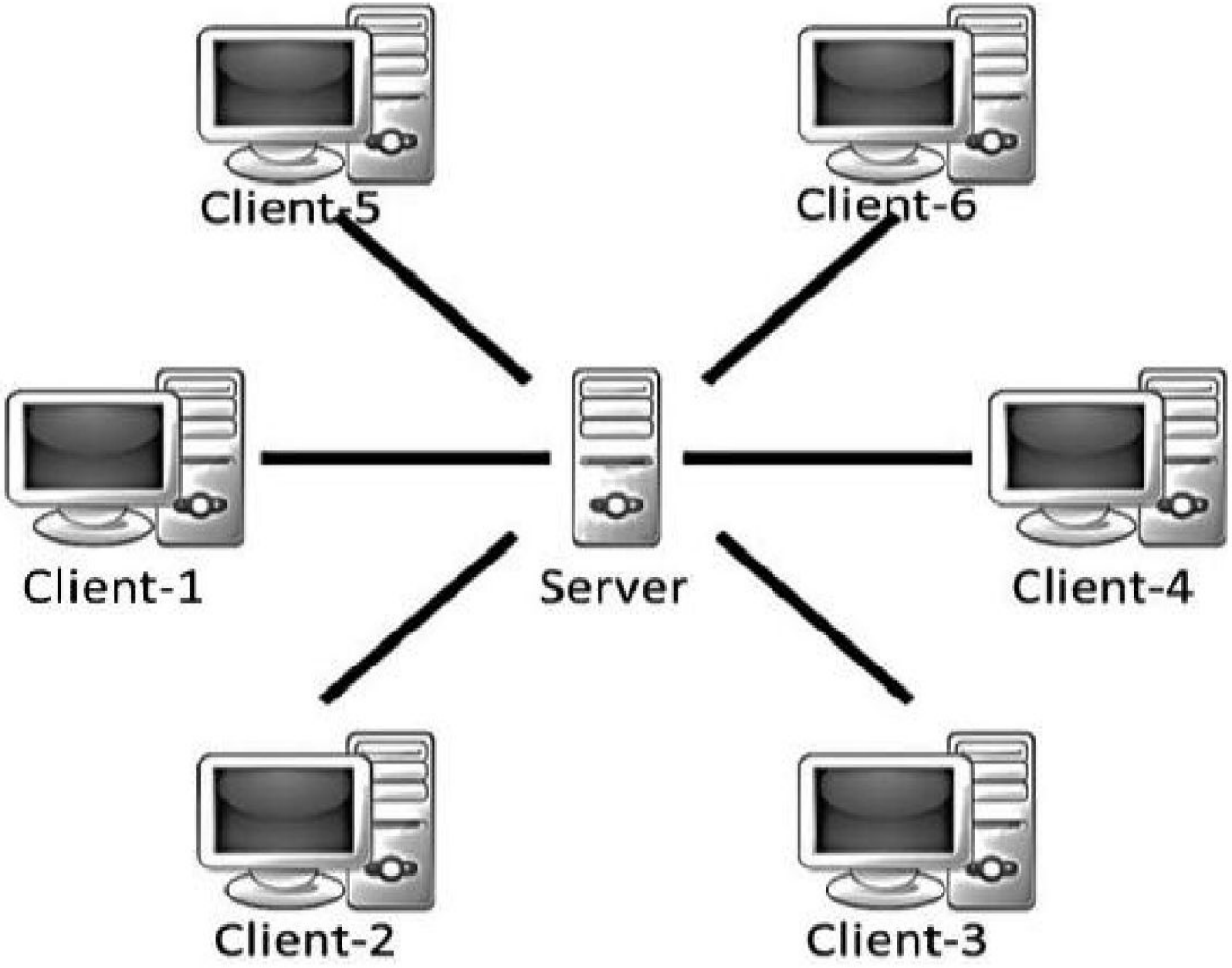
Blockchain


Genesis Block
Block-1
Previous Block
Transactions
Block-2
Previous Block
Transactions
Block-3
Previous Block
Transactions
Block-4
Previous Block
Transactions
Block-1265

CENTRALIZED SYSTEMS

As the name suggests, a centralized system has a centralized control with all administrative authority. Such systems are easy to design, maintain, impose trust, and govern, but suffer from many inherent limitations, as follows:

- They have a central point of failure, so are less stable.
- They are more vulnerable to attack and hence less secured.
- Centralization of power can lead to unethical operations.
- Scalability is difficult most of the time



A typical centralized system may appear as shown in Figure

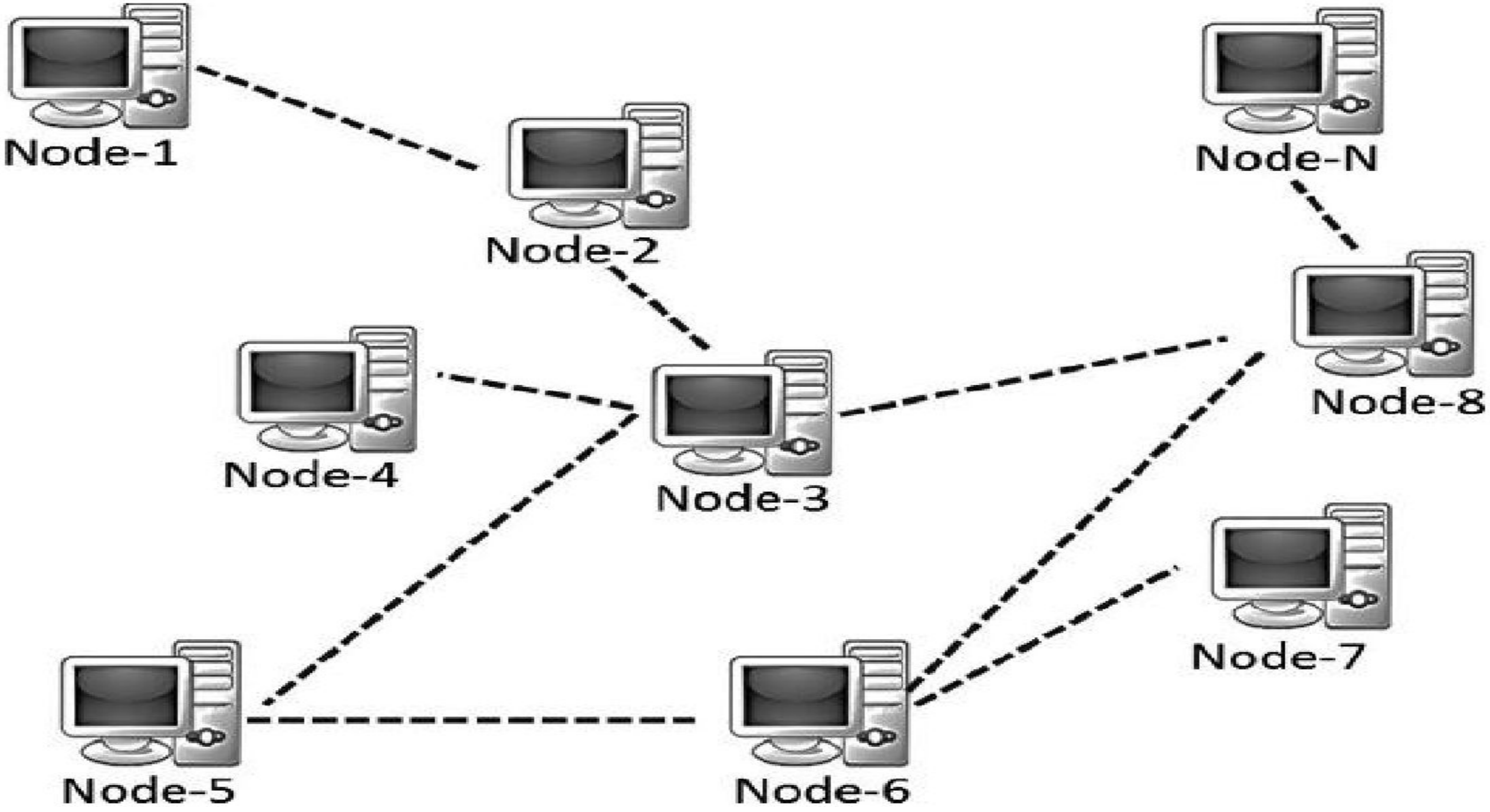
DECENTRALIZED SYSTEMS

As the name suggests, a decentralized system does not have a centralized control and every node has equal authority. Such systems are difficult to design, maintain, govern, or impose trust. However, they do not suffer from the limitations of conventional centralized systems. Decentralized systems offer the following advantages:

- They do not have a central point of failure, so more stable and fault tolerant

- Attack resistant, as no central point to easily attack and hence more secured

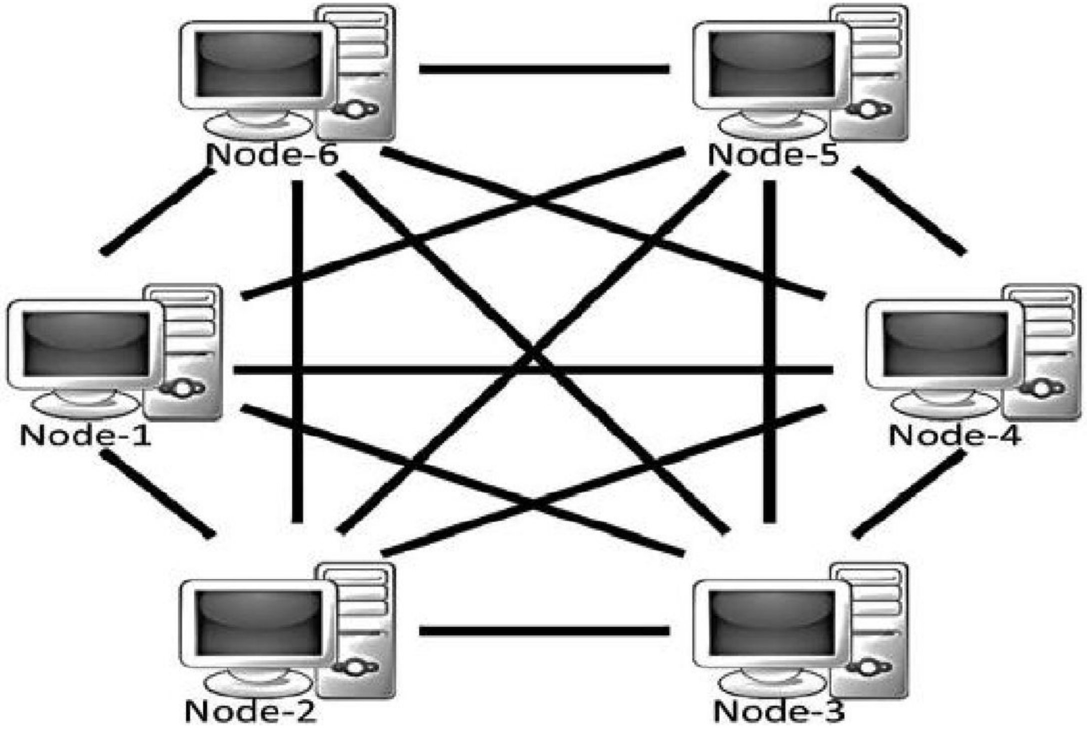
- Symmetric system with equal authority to all, so less scope of unethical operations and usually democratic in nature



A typical Decentralized system may appear as shown in Figure

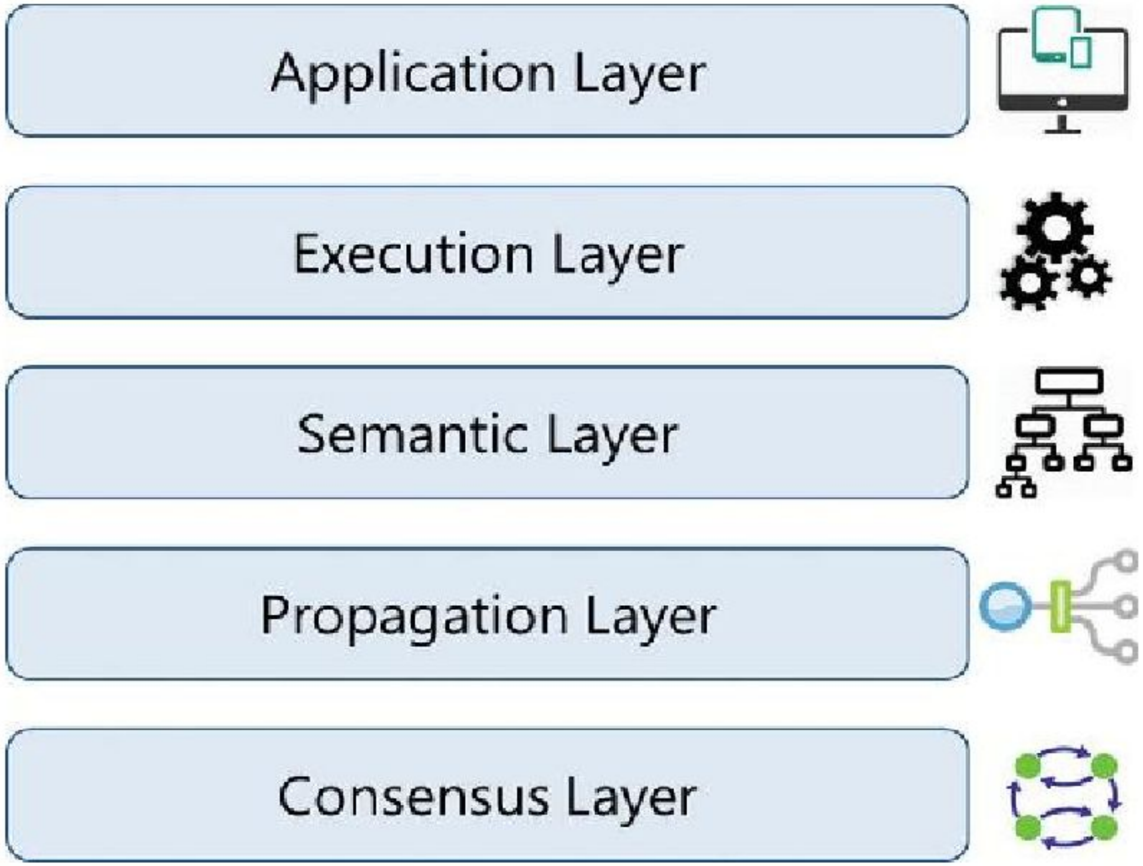
Peer to Peer Systeam

Please note that a distributed system can also be decentralized. An example would be blockchain! However, unlike common distributed systems, the task is not subdivided and delegated to nodes, as there is no master who would do that in blockchain. The contributing nodes do not work on a portion of the work, rather, the interested nodes (or the ones chosen at random) perform the entire work.



A typical Peer-to-Peer system may appear as shown in Figure

Layers of Blockchain



Application Layer

Execution Layer

Semantic Layer

Propagation Layer

Consensus Layer

-The purpose of Layers of Blockchain is not really to standardize blockchain technology, but to build a better understanding. Please keep in mind that all these layers are present on all the nodes.

APPLICATION LAYER

- This is the layer where you code up the desired functionalities and make an application out of it for the end users.
- It usually involves a traditional tech stack for software development such as client-side programming constructs, scripting, APIs, development frameworks, etc.
- In other words, this concept is to ensure that the heavy lifting is done at the application layer, or bulky storage requirements are taken care of off the chain so that the core blockchain is light and effective and the network traffic is not too much

EXECUTION LAYER

- The Execution Layer is where the executions of instructions ordered by the Application Layer take place on all the nodes in a blockchain network.
- The instructions could be simple instructions or a set of multiple instructions in the form of a smart contract
- All the nodes in a blockchain network have to execute the programs/scripts independently.
- Deterministic execution of programs/scripts on the same set of inputs and conditions always produces the same output on all the nodes, which helps avoid inconsistencies. .

SEMANTIC LAYER

- The Semantic Layer is a logical layer because there is an orderliness in the transactions and blocks.
- A transaction, whether valid or invalid, has a set of instructions that gets through the Execution Layer but gets validated in the Semantic Layer.
- It is the semantic layer that defines how the blocks are linked with each other. Every block in a blockchain contains the hash of the previous block, all the way to the genesis block.
- Though the final state of the blockchain is achieved by the contributions from all the layers, the linking of blocks with each other needs to be defined in this layer.

PROPAGATION LAYER

- The previous layers were more of an individual phenomenon: not much coordination with other nodes in the system.
- The Propagation Layer is the peer-to-peer communication layer that allows the nodes to discover each other, and talk and sync with each other with respect to the current state of the network.
- When a transaction is made, we know that it gets broadcast to the entire network.
- Similarly, when a node wants to propose a valid block, it gets immediately propagated to the entire network so that other nodes could build on it, considering it as the latest block.
- So, transaction/block propagation in the network is defined in this layer, which ensures stability of the whole network.

CONSENSUS LAYER

- The Consensus Layer is usually the base layer for most of the blockchain systems.
- The primary purpose of this layer is to get all the nodes to agree on one consistent state of the ledger.
- There could be different ways of achieving consensus among the nodes, depending on the use case. Safety and security of the blockchain is ascertained in this layer.
- In Bitcoin or Ethereum, the consensus is achieved through proper incentive techniques called “mining.”
- For a public blockchain to be self-sustainable, there has to be some sort of incentivization mechanisms that not only helps in keeping the network alive, but also enforces consensus .
- Bitcoin and Ethereum use a Proof of Work (PoW) consensus mechanism to randomly select a node that can propose a block.

Cryptography

Cryptography has been around for more than two thousand years now. It is the science of keeping things confidential using encryption techniques. Cryptography has following features:

- **Confidentiality** : Only the intended or authorized recipient can understand the message. It can also be referred to as privacy or secrecy.
- **Data Integrity** : Data cannot be forged or modified by an adversary intentionally or by unintended/accidental errors. Though data integrity cannot prevent the alteration of data, it can provide a means of detecting whether the data was modified.
- **Authentication** : The authenticity of the sender is assured and verifiable by the receiver.
- **Non-repudiation** : The sender, after sending a message, cannot deny later that they sent the message. This means that an entity (a person or a system) cannot refuse the ownership of a previous commitment or an action.

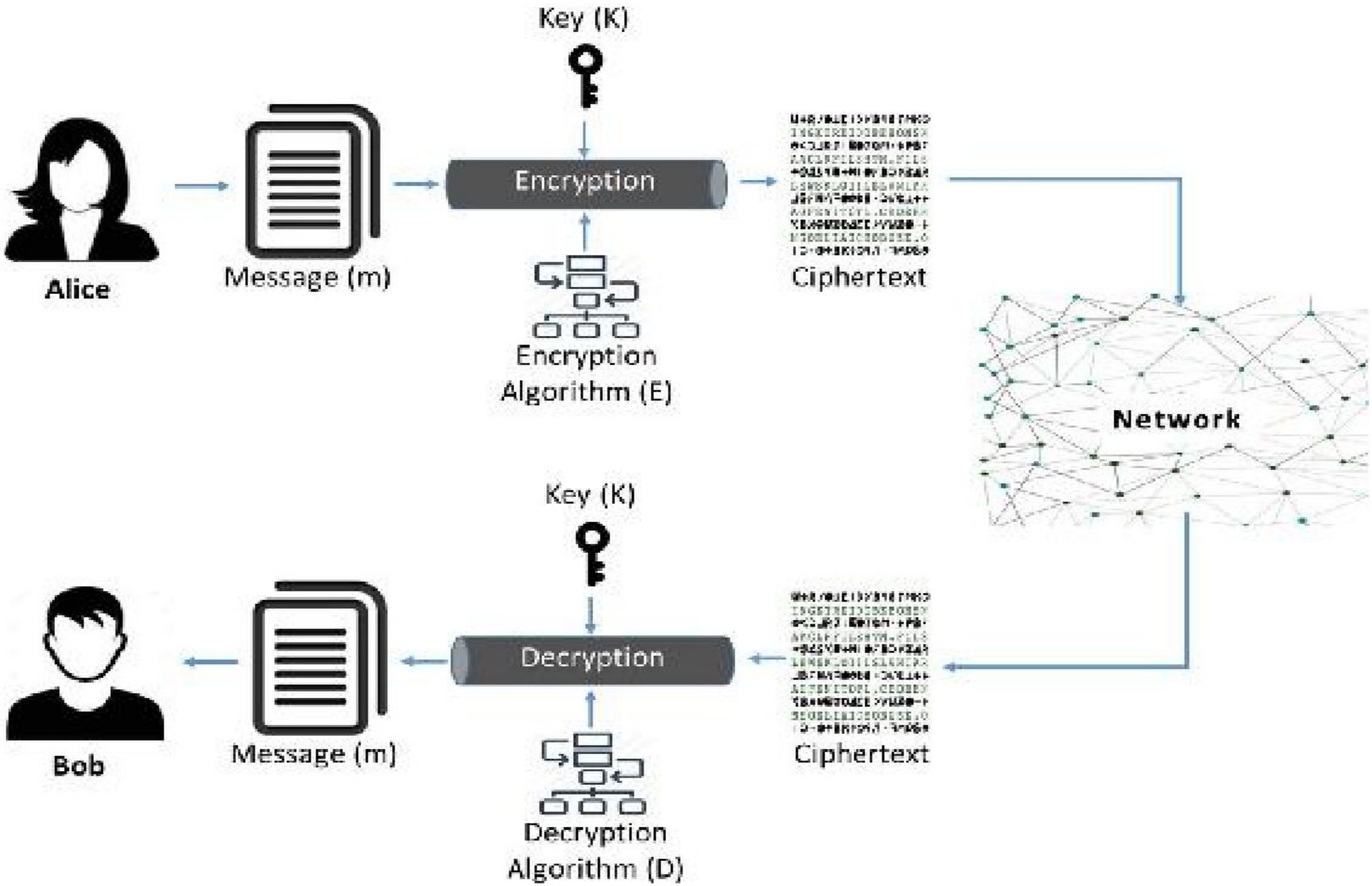


Fig: How Cryptography works in general

SYMMETRIC KEY CRYPTOGRAPHY

- In the previous section we looked at how Alice can encrypt a message and send the ciphertext to Bob.
- Bob can then decrypt the ciphertext to get the original message. If the same key is used for both encryption and decryption, it is called symmetric key cryptography.
- This means that both Alice and Bob have to agree on a key (k) called “shared secret” before they exchange the ciphertext. So, the process is as follows:

Alice—the Sender:

- Encrypt the plaintext message m using encryption algorithm E and key k to prepare the ciphertext c
- c = E(k, m)
- Send the ciphertext c to Bob

Bob—the Receiver:

- Decrypt the ciphertext c using decryption algorithm D and the same key k to get the plaintext m
- m = D(k, c)

- Kerckhoff’s Principle and XOR Function
- Stream Ciphers vs. Block Cipher
- One-Time Pad
- Data Encryption Standard

Challenges in Symmetric Key Cryptography

There are some limitations in symmetric key cryptography. A few of them are listed as follows:

1. The key must be shared by the sender and receiver before any communication.
2. It requires a secured key establishment mechanism in place. The sender and receiver must trust each other, as they use the same symmetric key. If a receiver is hacked by an attacker or the receiver deliberately shared the key with someone else, the system gets compromised.
3. A large network of, say, n nodes require key n(n–1)/2 key pairs to be managed.
4. It is advisable to keep changing the key for each communication session.

HASH FUNCTIONS

A cryptographic hash function is a one-way function that converts input data of arbitrary length and produces a fixed-length output. The output is usually termed “hash value” or “message digest.”