

Q.1 Write short notes on synchronous and asynchronous byzantine agreement

Synchronous Byzantine Agreement:

- **Synchronous Byzantine Agreement** refers to a class of consensus algorithms designed to achieve agreement among a network of participants despite the presence of Byzantine faults.
- In a synchronous setting, there is a known upper bound on message delays and process execution times, and all processes in the system are expected to follow this timing assumption.
- Synchronous Byzantine Agreement protocols typically work in rounds, where participants exchange messages and perform computations within each round.
- The protocols aim to ensure that all correct participants agree on a common value, even in the presence of Byzantine faults, where some participants may behave arbitrarily.
- Synchronous Byzantine Agreement protocols often utilize techniques such as digital signatures, redundancy, and voting schemes to achieve consensus and tolerate Byzantine faults.
- The protocols rely on the assumption that the majority of participants are correct and that faulty participants cannot control the majority of the network.

Asynchronous Byzantine Agreement:

- **Asynchronous Byzantine Agreement** refers to a class of consensus algorithms designed to achieve agreement among a network of participants in an asynchronous environment.
- In an asynchronous setting, there are no assumptions about message delays or process execution times, and participants can exhibit arbitrary delays or failures.
- Asynchronous Byzantine Agreement protocols face additional challenges compared to synchronous protocols due to the lack of timing assumptions and the possibility of long delays or network partitions.
- These protocols often use techniques such as redundancy, error detection, and consensus rounds to overcome the challenges of asynchrony and Byzantine faults.
- Achieving asynchronous Byzantine Agreement typically requires additional rounds of communication and computation compared to synchronous protocols.
- Asynchronous Byzantine Agreement protocols often focus on achieving safety (agreement on a common value) and liveness (eventual termination and progress) properties, even in the presence of Byzantine faults and asynchrony.

Q. 2 What do you understand by the byzantine general problem

ANS . 2)

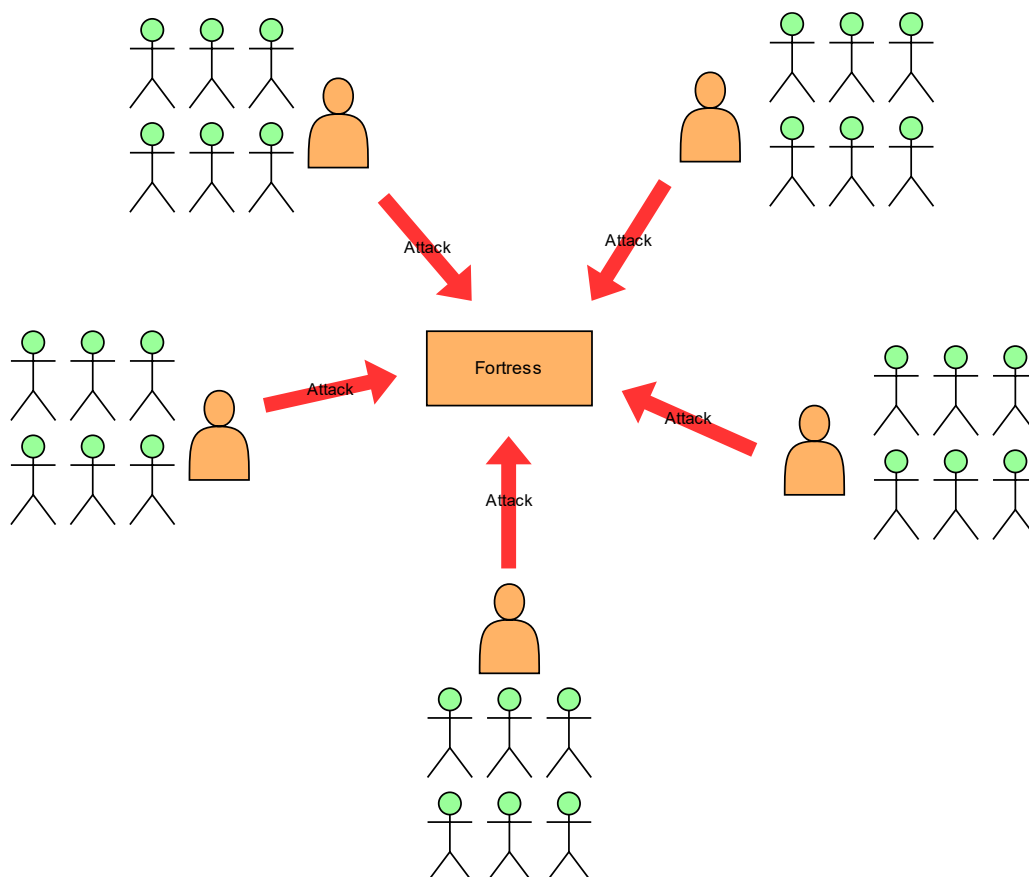
The Byzantine Generals' problem refers to the game theory problem. A group of generals attacks a fortress; every general has an army and surrounds a fort from one side. Every general has a preference about whether to attack or retreat. It has to be a coordinated attack or retreat to incur minimum losses. Thus, a consensus is held, and the majority decision is implemented.

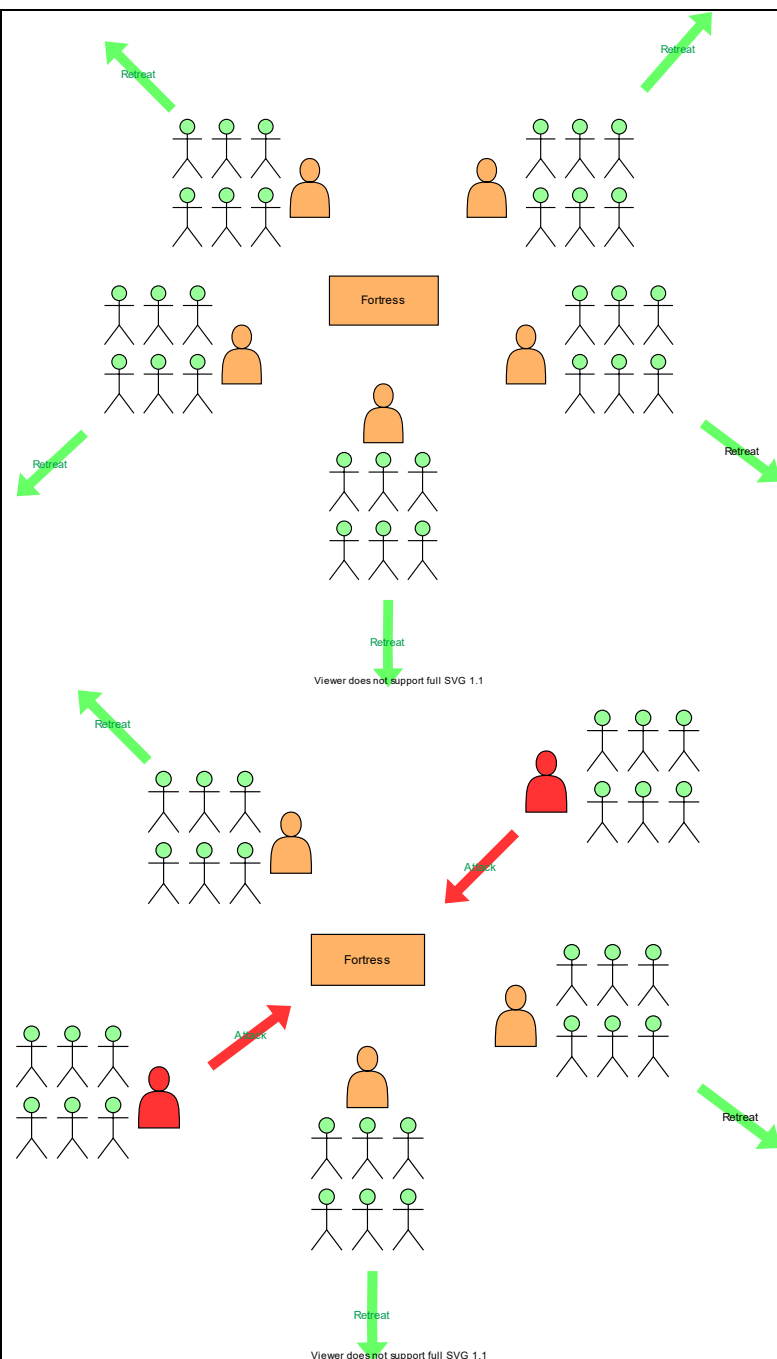
This consensus is formed after following the following steps:

1. Every general sends their own choice to all other generals.
2. After receiving the choice of all generals, every general calculates the votes in favor of attacking and retreating.
3. If the majority is in favor of retreat, then they retreat; otherwise, they attack.

Suppose there is a traitor general who sends a retreat message to half generals and an attack message to the other half generals. Then half of the generals may end up attacking while the other half will retreat, causing the army to lose.

The following slides show all the scenarios of a successful attack, a successful retreat, and an unsuccessful attack.





Application

- **Blockchain:** In blockchains, we've got a network of nodes (generals), and they have to decide whether to add a block to their journal or not. All the nodes must make the same decision (either to add the block or not) to maintain a constant state of the blockchain across all the nodes. Otherwise, every node will have a different view of the blockchain, making it hard to maintain the network.
- **Distributed systems:** In data centers, we have a lot of servers to handle user requests. All these servers have their data; if a user request results in a change in data, then all the servers need to update their data(to keep the data consistent across all the servers). If some servers(generals) fail or send the wrong information at this point, this may bring down the whole system.

Q.2 What are the different encryption schemes available

Common Encryption Algorithms

1. Triple DES

- [Triple DES](#) was designed to replace the original Data Encryption Standard (DES) algorithm, which hackers eventually learned to defeat with relative ease. At one time, Triple DES was the recommended standard and the most widely used symmetric algorithm in the industry.
- Triple DES uses three individual keys with 56 bits each. The total key length adds up to 168 bits, but experts would argue that 112-bits in key strength is more accurate. Despite slowly being phased out, Triple DES has, for the most part, been replaced by the Advanced Encryption Standard (AES).

2. AES

- The [Advanced Encryption Standard \(AES\)](#) is the algorithm trusted as the standard by the U.S. Government and numerous organizations. Although it is highly efficient in 128-bit form, AES also uses keys of 192 and 256 bits for heavy-duty encryption purposes.
- AES is largely considered impervious to all attacks, except for brute force, which attempts to decipher messages using all possible combinations in the 128, 192, or 256-bit cipher.

3. RSA Security

- [RSA](#) is a public-key encryption algorithm and the standard for encrypting data sent over the internet.
- It also happens to be one of the methods used in PGP and GPG programs.
- Unlike Triple DES, RSA is considered an asymmetric algorithm due to its use of a pair of keys.
- You've got your public key to encrypt the message and a private key to decrypt it.
- The result of RSA encryption is a huge batch of mumbo jumbo that takes attackers a lot of time and processing power to break.

4. Blowfish

- [Blowfish](#) is yet another algorithm designed to replace DES. This symmetric cipher splits messages into blocks of 64 bits and encrypts them individually.
- Blowfish is known for its tremendous speed and overall effectiveness.
- Meanwhile, vendors have taken full advantage of its free availability in the public domain.
- You'll find Blowfish in software categories ranging from e-commerce platforms for securing payments to password management tools, where it protects passwords.
- It's one of the more flexible encryption methods available.

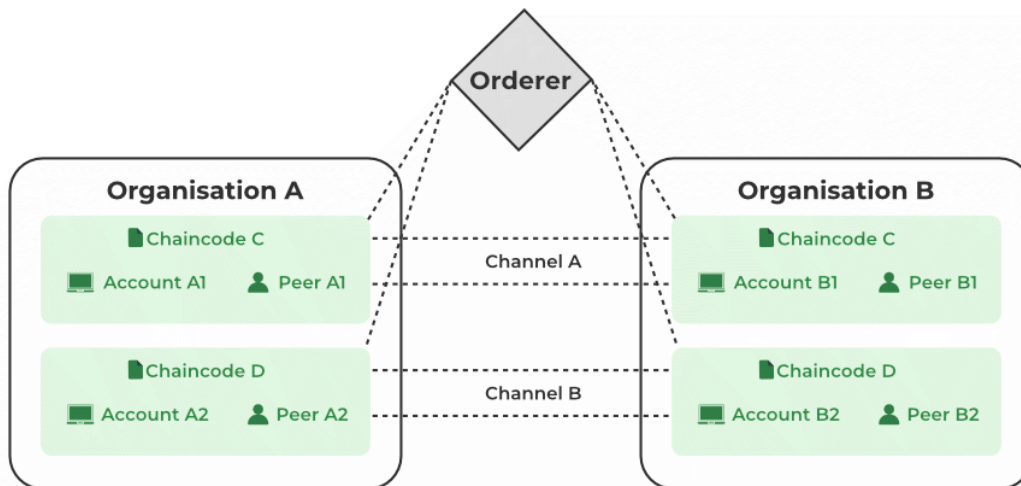
5. Twofish

- Computer security expert Bruce Schneier is the mastermind behind Blowfish and its successor [Twofish](#).
- Keys used in this algorithm may be up to 256 bits in length, and as a symmetric technique, you only need one key.
- Twofish is one of the fastest of its kind and ideal for use in hardware and software environments.
- Like Blowfish, Twofish is freely available to anyone who wants to use it.

Q.1 Discuss the elements of hyperledger

Hyperledger Fabric Infrastructure Components

One of the key design principles of Hyperledger Fabric is modularity and extensibility. The platform is composed of several components that can be customized and extended to meet the specific requirements of different applications and use cases. The following are some of the main components of Hyperledger Fabric and their roles in the platform:



1. Peer Node

This is the main component of Hyperledger Fabric. It is responsible for managing the ledger, executing smart contracts, and participating in the consensus process. A peer node in Hyperledger Fabric is a server that runs the Hyperledger Fabric software and is part of a network of peer nodes that make up a Hyperledger Fabric blockchain. Each peer node stores a copy of the ledger and participates in the consensus process to validate and endorse transactions, as well as maintain the state of the ledger. Peer nodes can also run smart contracts, known as chaincode in Hyperledger Fabric, which define the business logic of the blockchain network. There are 5 types of Peer nodes:

- Full node:
- Lightweight node:
- Supernode:
- Mining node:
- Validating node:

2. Orderer

In Hyperledger Fabric, an orderer is a component that is responsible for ensuring the delivery of transactions to the appropriate peer nodes for validation and endorsement. The orderer maintains an ordered log of all transactions that have occurred on the network and provides a communication channel for the peer nodes to reach a consensus on the order in which transactions should be processed. The orderer does not validate or endorse transactions but rather acts as a mediator to facilitate communication between the different peer nodes. There are 5 types of Orderer:

- Solo orderer:**
- Kafka orderer:**
- Raft orderer:**
- SNS orderer:**
- CouchDB orderer:**

3. Membership service provider (MSP)

In Hyperledger Fabric, a Membership Service Provider (MSP) is a component that defines the rules for identity management and authentication within a Hyperledger Fabric network. MSPs are used to verify the identity of participants in the network, such as users, applications, and peer nodes. They do this by managing the certificates and cryptographic materials that are used to identify and authenticate entities on the network. MSPs play a crucial role in ensuring the security and integrity of the Hyperledger Fabric network by ensuring that only authorized entities are able to access the network and participate in transactions. There are several types of membership service providers (MSPs) in Hyperledger, including:

- Local MSP:**
- File-based MSP:**
- Certificate Authority-based MSP:**
- Database-based MSP:**
- External Identity Provider-based MSP:**

4. Ledger

In Hyperledger Fabric, the ledger is a distributed database that records all of the transactions that occur on the network. Each peer node maintains a copy of the ledger, and the ledger is updated whenever a new transaction is endorsed and committed to the network. The ledger is composed of two parts: the world state, which stores the current state of all assets on the network, and the transaction log, which stores a record of all transactions that have occurred on the network. The ledger is used to provide an immutable record of all transactions and to ensure the consistency and integrity of the data on the network. There are several types of ledgers in Hyperledger, including:

- Fabric:**
- Sawtooth:**
- Iroha:**
- Indy:**
- Besu:**

5. Chaincode

In Hyperledger Fabric, chaincode is the term used to refer to smart contracts. Chaincode is written in Go and defines the business logic of a Hyperledger Fabric network. It specifies the rules for updating the ledger and determines which transactions are valid. When a transaction is submitted to the network, it is sent to the appropriate peer nodes for endorsement. The chaincode is then executed on the endorsing peer nodes, and the endorsed transaction is sent back to the client for ordering and finally commit to the ledger. Chaincode is an important part of the Hyperledger Fabric architecture, as it allows users to define the specific rules and functionality of their blockchain network.

In Hyperledger, chaincode refers to the smart contract code that is written in a programming language, such as Go or Java, and deployed on the blockchain network. There are two types of chaincode in Hyperledger Fabric:

- System chaincode:**
- User chaincode:**

6. Consensus Algorithm

In Hyperledger Fabric, the consensus algorithm is the mechanism by which the peer nodes in a network reach an agreement on the order and validity of transactions. The consensus algorithm is an important part of the overall architecture of a Hyperledger Fabric network, as it ensures the integrity and consistency of the ledger by ensuring that all peer nodes have a consistent view of the state of the network. Hyperledger Fabric supports several different consensus algorithms, including the Practical Byzantine Fault Tolerance (PBFT) algorithm and the Kafka-based consensus algorithm. The specific consensus algorithm used by a Hyperledger Fabric network can be configured and customized to meet the needs of the specific application.

- Proof of Work (PoW):**
- Proof of Stake (PoS):**
- Practical Byzantine Fault Tolerance (PBFT):**
- Federated Byzantine Agreement (FBA):**
- Delegated Proof of Stake (DPoS):**

7. Channels

In Hyperledger Fabric, a channel is a private “subnet” within a Hyperledger Fabric network that allows a group of participants to execute transactions and share data in a confidential manner. Each channel has its own separate ledger, and the participants on a channel can only see the transactions that are submitted to that channel. This allows different groups of participants within a Hyperledger Fabric network to have their own private, confidential interactions without revealing sensitive information to the other participants on the network. Channels provide an additional layer of security and privacy within a Hyperledger Fabric network. In Hyperledger Fabric, there are three types of channels:

- Application channels:**
- System channels:**
- Private channels:**

These components work together to provide a modular, extensible platform for building blockchain applications. By customizing and extending these components, developers can create applications that are tailored to the specific requirements of different industries and use cases.

Q.3 Differentiate between Blockchain and Bitcoin



101 Blockchains

BITCOIN VS. BLOCKCHAIN


	BITCOIN	BLOCKCHAIN
DEFINITION	The initial cryptocurrency variant	A distributed ledger for storing records of transactions
OBJECTIVE	Simplification and improvement in speed of transactions without any government restrictions	Providing an environment for peer-to-peer transactions with a low cost, secure, and safe environment
SCOPE	Limited to the role of a currency	Better adaptability to change and more support of top companies
TRADING	Only provides currency trading	Can support transfer of currencies as well as stocks, contracts, and property rights
STRATEGY	Reducing the cost of intermediaries and time of transactions	Effective responsiveness to change for catering requirements of different industries

CREATED BY 101BLOCKCHAINS.COM

[illegible]

Bitcoin vs Blockchain

Comparison Chart

Bitcoin	Blockchain
Bitcoin is a digital currency that allows you to perform online transactions anonymously.	Blockchain is a digital public ledger that holds and catalogues all the Bitcoin transactions
It is a cryptocurrency that is not governed by a central authority or a central bank.	It is a technology that uses a distributed network of computers to hold transaction records.
It is a decentralized currency that works as a medium of exchange to secure transactions.	Although a Blockchain is inherently distributed, it is not necessarily decentralized.
Bitcoins cannot work without Blockchains.	Blockchain is a database for recoding transactions which is not limited to Bitcoins. 

Q.4 Explain the transition life cycle of Blockchain

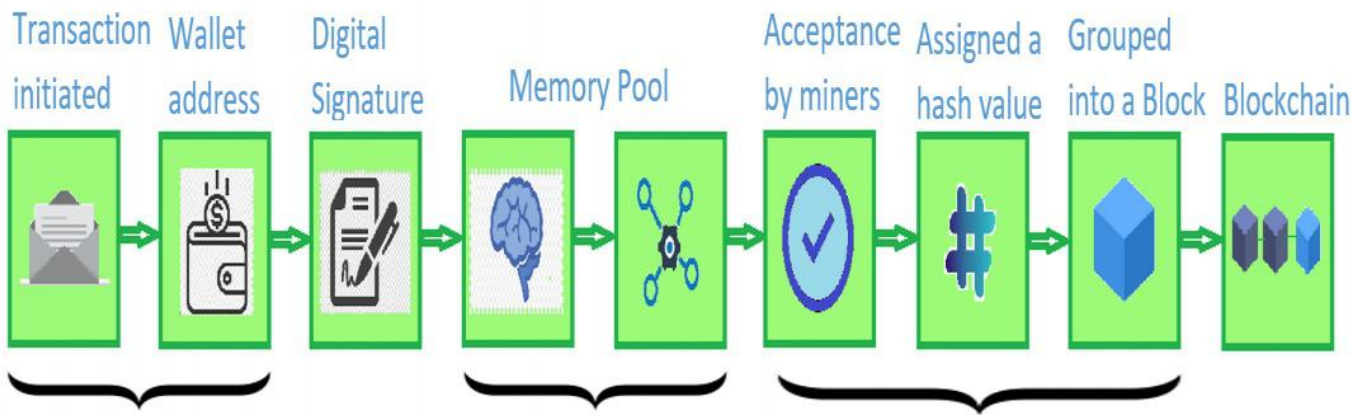
Blockchain technology is mostly about the transactions that we make digitally for ourselves. Eventually, these transactions make their way to the various blocks that become part of the Blockchain later on. So, it is important to understand the transaction life cycle in Blockchain technology. This lifecycle follows the journey of a single transaction as it makes its way through each stage in the process of joining the blockchain. Transaction in simple words is the process of sending money by the sender and the receiver receiving it. The Blockchain transaction is also quite similar, but it is made digitally.

Let us understand the various stages in a blockchain transaction life cycle with the help of an example.

Sourav and Suraj are two Bitcoin users. Sourav wants to send 1 bitcoin to Suraj.

1. First, Sourav gets Suraj's wallet address (a wallet in the blockchain is a digital wallet that allows users to manage their transactions). Using this information, he creates a new transaction for 1 bitcoins from his wallet and includes a transaction fee of 0.003 bitcoin.
2. Next, he verifies the information and sends the transaction. Each transaction that is initiated is signed by a digital signature of the sender that is basically the private key of the sender. This is done in order to make the transaction more secure and to prevent any fraud.
3. Sourav's wallet then starts the transaction signing algorithm which signs his transaction using his private key.
4. The transaction is now broadcasted to the memory pool within the network.
5. This transaction is eventually accepted by the miners. These miners, group this transaction into a block, find the Proof of Work, and assign this block a hash value to be mapped into the blockchain.
6. This block is now placed on the Blockchain.
7. As this block gains confirmation, it is accepted as a valid transaction in the network.
8. Once this transaction is accepted, Suraj finally gets his bitcoin.

The below diagram is a pictorial representation of the various stages in a transaction life cycle as discussed above.



1. A transaction is created by the sender using the wallet address of the receiver
2. Sign a transaction
3. Transaction is sent to the memory pool
4. Transaction is accepted by the miners, grouped into a block and assign a hash value to the block
5. The block now becomes a part of blockchain

Q.1 What is the importance of Signature and Confidentiality functions

Signature and confidentiality functions are crucial elements in information security and cryptography. They serve different purposes and play significant roles in ensuring the integrity and confidentiality of sensitive data.

1. Signature Function:

The signature function is primarily concerned with ensuring the integrity and authenticity of digital messages or documents. It involves using cryptographic techniques to create a unique digital signature that can be attached to a message or document. The signature provides the following key benefits:

- Integrity: A digital signature allows the recipient of a message or document to verify that it has not been altered during transit. Any modification made to the content of the message or document after the signature was created will invalidate the signature.**
- Authenticity: Digital signatures provide proof of the sender's identity. The recipient can verify the authenticity of the sender by verifying the attached signature against the sender's public key. If the signature is valid, it indicates that the message or document was indeed sent by the claimed sender.**
- Non-repudiation: Digital signatures provide non-repudiation, meaning that the sender cannot deny having sent the message or document. Since the signature is unique to the sender and is mathematically bound to the content, the sender cannot later claim that they did not send the message.**

Digital signatures are widely used in various applications, such as secure email communication, online transactions, electronic contracts, and legal documents, to ensure the integrity, authenticity, and non-repudiation of the exchanged information.

2. Confidentiality Function:

The confidentiality function focuses on protecting sensitive information from unauthorized access or disclosure. Cryptographic techniques, such as encryption, are used to transform the original data into a form that can only be understood by authorized parties. The key advantages of confidentiality measures include:

- Privacy: Encryption ensures that only authorized individuals or systems can access and understand the encrypted information. It prevents unauthorized individuals, including hackers or eavesdroppers, from gaining access to the sensitive data.**
- Data Protection: Confidentiality measures protect data from being compromised if it is intercepted during transmission or storage. Even if an attacker manages to access the encrypted data, they would not be able to decipher it without the corresponding decryption key.**
- Compliance: In many industries and jurisdictions, there are legal requirements and regulations that mandate the protection of sensitive data. Implementing confidentiality measures, such as encryption, helps organizations meet these compliance standards.**

Confidentiality measures are utilized in various scenarios, including secure communication channels, protecting stored data, safeguarding personal or financial information, and securing sensitive business data.

In summary, the signature function ensures integrity, authenticity, and non-repudiation of digital messages or documents, while the confidentiality function safeguards sensitive information from unauthorized access or disclosure. Both functions are fundamental components of information security and cryptography, helping to establish trust, protect data, and maintain the privacy of communication and sensitive data.

Q.1 Write short notes on protocols of Blockchain.

Protocols are rules which govern the functioning of a blockchain. Since Blockchains are a network of computers which operate on a peer-to-peer basis, protocols define how information is transferred between computers on the network.

Blockchains which note all the transactions of a specific crypto token need to be governed by a set of rules. These rules are essentially the heart of the blockchain. It gives an idea to the miners, stakers and the investing community about how exactly the blockchain functions. The rules also help investors identify if the crypto is worth investing in or not.

Protocols also impact network performance and security measures. These are functional building blocks of the blockchain and hence it becomes imperative for one to stay informed about the same. A detailed blockchain protocol list is mentioned below.

Five Types of Blockchain Protocols Which are Widely Accepted

1. Hyperledger

- **Hyperledger is a highly reputed protocol which powers enterprises to develop blockchain-based solutions specific to their needs. Industry giants like JP Morgan and Samsung have leveraged this technology for developing business applications. Several Hyperledger projects have graduated from an incubation stage and have caught fame. Namely: Hyperledger Besu, Hyperledger Fabric, Hyperledger Indy, Hyperledger Indy, Hyperledger Iroha, and Hyperledger Sawtooth. Each version has its possibilities.**
-
- **Hyperledger also has an extensive library which can help developers build applications. The solutions are crypto-agnostic which means the functionality of the blockchain does not depend on the price of crypto.**
- **This infrastructure provided by Hyperledger makes it one of the most sought blockchain solutions in the industry.**

2. Multichain

- It is a blockchain protocol built for communication within organisations or between organisations. Multichain provides solutions for private blockchains. Multi-chain provides an API which can be utilised for the development of blockchain solutions. It cuts down the development time by almost 80% according to the multi-chain website. Unlike public blockchains, blockchains built using [Multichain](#) offer complete control over the blockchain and hence it is most suited for organisations to deal with high-privacy financial transactions like banking.

3. Ethereum

- The Ethereum blockchain is perhaps the most explored blockchain to date. It allows the creation of Decentralised finance platforms, NFTs, and Smart contracts with various applications. Ethereum is an open-source public blockchain which runs on the Proof of Stake Consensus mechanism. [Proof of Stake](#) makes the blockchain hyper-efficient with 99.95% less energy requirement than its previous [Proof of Work](#) version.
- Decentralised Autonomous Organisations(DAO) can build [Smart contracts](#) on this blockchain. These DAOs have their constitution, voting methods, tokenomics, treasury and reward system. DAOs exist to build a profitable business or add value to the Ethereum ecosystem while ensuring that the power of decision-making is democratised to all members of the group. There is an endless list of applications which run on Ethereum. Some of the notable ones include Metamask (a crypto wallet), Brave Browser (A web browser which is giving tough competition to Google chrome) and many more.

4. Quorum

- It is an open-source solution for companies in the finance sector. It is backed by JP Morgan, one of the largest private banks in the world. It enables the use of Ethereum to build applications for specific uses. Quorum blockchain service can be run through a Microsoft azure account and can be easily deployed from the Azure marketplace. This enables enterprises to build blockchain services in the cloud.

5. Corda

- **It is an open-source project which enables interoperability, which is the power of exchanging information between various blockchains. Transactions performed using Corda are transparent and highly anonymised at the same time. While its solutions are tailored towards banking, several integrations of Corda can make it a dependable solution for any application. It is accredited by the R3 banking consortium making it a marquee name for blockchain solutions for banking.**