□□□□□□□□□□

CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 March 2025

Sub:	Blockchain Technology			Sub Code:	BCS613A	Branch:	CSE
Date:	26.03.2025	Duration:	90 minutes	Max Marks:	50	Sem / Sec:	VI / A, B, C

	Answer any FIVE FULL Questions	MARKS	CO	RBT
			OBE	
1	a)Explain different types of blockchain	5	CO1	L2
	b)Mention benefits and limitations of blockchain	5	CO1	L1
2	a)Explain the methods of achieving decentralization	5	CO2	L2
	b)Describe the various decentralized organizations in blockchain	5	CO2	L2
3	a)Explain Elliptic curve cryptography.	7	CO2	L2
	b)What are smart contracts?	3	CO2	L1
4	a)Explain Merkle tree with the help of a neat diagram. Explain the use of Merkle tree in blockchain	6	CO2	L2
	b)List the major differences between symmetric and asymmetric cryptography	4	CO2	L1
5	a)What is the Byzantine generals problem, and how is it relevant to blockchain design	7	CO1	L2
	b)If your blockchain network has 6 Byzantine nodes, what is the minimum number of nodes that are required to ensure Byzantine fault tolerance using PBFT protocol?	3	CO1	L2
6	Explain SHA 256 algorithm	10	CO2	L2

Solution

1. a) Explain different types of blockchain

TYPES OF BLOCKCHAIN

I. Public blockchains

- As the name suggests, these blockchains are open to the public and anyone can participate as a node in the decision-making process.
- Users may or may not be rewarded for their participation. These ledgers are not owned by anyone and are publicly open for anyone to participate in.
- All users of the permission-less ledger maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism in order to reach a decision about the eventual state of the ledger.
- These blockchains are also known as permission-less ledgers.

II. Private blockchains

- Private blockchains as the name implies are private and are open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves.

III. Semi-private blockchains

- Here part of the blockchain is private and part of it is public. The private part is controlled by a group of individuals whereas the public part is open for participation by anyone.

IV. Sidechains

- More precisely known as pegged sidechains, this is a concept whereby coins can be moved from one blockchain to another and moved back. Common uses include the creation of new altcoins (alternative cryptocurrencies) whereby coins are burnt as a proof of adequate stake.
- There are two types of sidechain. The example provided above for burning coins is applicable to a one-way pegged sidechain. The second type is called a two-way pegged sidechain, which allows the movement of coins from the main chain to the sidechain and back to the main chain when required.

V. Permissioned ledger:

A permissioned ledger is a blockchain whereby the participants of the network are known and already trusted.

- Permissioned ledgers do not need to use a distributed consensus mechanism, instead an agreement protocol can be used to maintain a shared version of truth about the state of the records on the blockchain.
- There is also no requirement for a permissioned blockchain to be private as it can be a public blockchain but with regulated access control.

VI. Distributed ledger

- As the name suggests, this ledger is distributed among its participants and spread across multiple sites or organizations. This type can either be private or public. The key idea is that, unlike many other blockchains, the records are stored contiguously instead of sorted into blocks. This concept is used in Ripple.

VII. Shared ledger

- This is generic term that is used to describe any application or database that is shared by the public or a consortium.
- Fully private and proprietary blockchains These blockchains perhaps have no mainstream application as they deviate from the core idea of decentralization in blockchain technology.
- Nonetheless in specific private settings within an organization there might be a need to share data and provide some level of guarantee of the authenticity of the data. These blockchains could be useful in that scenario. For example, for collaboration and sharing data between various government departments.

VIII. Tokenized blockchains

- These blockchains are standard blockchains that generate cryptocurrency as a result of a consensus process via mining or via initial distribution.

IX. Token less blockchains

- These are probably not real blockchains because they lack the basic unit of transfer of value but are still valuable in situations where there is no need to transfer value between nodes and only sharing some data among various already trusted parties is required.
- Consensus is the backbone of a blockchain and provides

decentralization of control as a result through an optional process known as mining. The choice of consensus algorithm is also governed by the type of blockchain in use. Not all consensus mechanisms are suitable for all types of blockchains.

- For example, in public permission-less blockchains it would make sense to use PoW instead of some basic agreement mechanism that perhaps is based on proof of authority. Therefore it is essential to choose a consensus algorithm appropriately for a blockchain project

1. b))Mention benefits and limitations of blockchain

BENEFITS OF BLOCKCHAIN

- Decentralization: This is a core concept and benefit of blockchain. There is no need for a trusted third party or intermediary to validate transactions; instead a consensus mechanism is used to agree on the validity of transactions

Transparency and trust: As blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent and as a result trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion should be restricted.

- Immutability: Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not truly immutable but, due to the fact that changing data is extremely difficult and almost impossible, this is seen as a benefit to maintaining an immutable ledger of transactions.
- High availability: As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on each and every node, the system becomes highly available. Even if nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available.
- Highly secure: All transactions on a blockchain are cryptographically secured and provide integrity.
- Simplification of current paradigms: The current model in many industries such as finance or health is rather disorganized, wherein multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. But as a blockchain can serve as a single shared ledger among interested parties, this can result in simplifying this model

by reducing the complexity of managing the separate systems maintained by each entity.

- Faster dealings: In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by allowing the quicker settlement of trades as it does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed upon data is already available on a shared ledger between financial organizations.

- Cost saving: As no third party or clearing houses are required in the blockchain model, this can massively eliminate overhead costs in the form of fees that are paid to clearing houses or trusted third parties.

CHALLENGES AND LIMITATIONS OF BLOCKCHAIN TECHNOLOGY

As with any technology there are challenges that need to be addressed in order to make a system more robust, useful, and accessible. Blockchain technology is no exception; in fact a lot of effort is being made in Academia and Industry to overcome the challenges posed by blockchain technology.

A selection of the most sensitive challenges are presented as follows:

❖ Scalability
❖ Adaptability
❖ Regulation Relatively immature technology Privacy
❖ Privacy

Even though blockchain technology has revolutionized many industries, it still faces several challenges and limitations that hinder its widespread adoption.

Scalability: It is one of the biggest drawbacks of blockchain technology as it cannot be scaled due to the fixed size of the block for storing information. The block size is 1 MB due to which it can hold only a couple of transactions on a single block.

Immaturity: Blockchain is only a couple-year-old technology so people do not have much confidence in it, they are not ready to invest in it several applications of blockchain are doing great in different industries but still it needs to win the confidence of even more people to be recognized for its complete utilization.

Energy Consuming: For verifying any transaction, a lot of energy is used so it becomes a problem according to the survey it is considered that 0.3 percent of the world's electricity had been used by 2018 in the verification of transactions done using blockchain technology.

Time-Consuming: To add the next block in the chain miners, need to compute nonce values many times so this is a time-consuming process and needs to be speed up to be used for industrial purposes.

Legal Formalities: In some countries, the use of blockchain technology applications is banned like cryptocurrency due to some environmental issues they are not promoting to use of blockchain technology in the commercial sector.

Storage: Blockchain databases are stored on all the nodes of the network creates an issue with the storage, increasing the number of transactions will require more storage.

Regulations: Blockchain faces challenges with some financial institutions. Other aspects of technology will be required in order to adopt blockchain in a wider aspect.

2. a) Explain the methods of achieving decentralization

Methods of decentralization

There are two methods that can be used to achieve decentralization.

1. Disintermediation
2. Contest-driven decentralization

1. Disintermediation:

- This can be explained with the help of an example. Imagine you want to send money to your friend in another country.
- You go to a bank that will transfer your money to the bank in the country of your choice for a fee. In this case, the bank keeps a central database that is updated, confirming that you have sent the money.
- With blockchain technology, it is possible to send this money directly to your friend without the need for a bank. All you need is the address of your friend on the blockchain. This way, the intermediary is no longer required and decentralization is achieved by disintermediation.
- However, it is debatable how practical decentralization is in the financial sector by disintermediation due to heavy regulatory and compliance requirements. Nevertheless, this model can be used not only in finance but also in many other different industries.
- In the health industry, where patients, instead of relying on a trusted third party (such as the hospital record system) can be in full control of their own identity and their data that they can share directly with only those entities that they trust.
- As a general solution, blockchain can serve as a decentralized health record management system where health records can be exchanged securely and directly between different entities (hospitals, pharmaceutical companies, patients) globally without any central authority.

2. Contest-driven decentralization

- In the method involving competition, different service providers compete with each other in order to be selected for the provision of services by the system. This paradigm does not achieve complete decentralization. However, to a certain degree, it ensures that an intermediary or service provider is not monopolizing the service.
- In the context of blockchain technology, a system can be envisioned in which smart contracts can choose an external data provider from a large number of providers based on their reputation, previous score, reviews, and quality of service. This method will not result in full decentralization, but it allows smart contracts to make a free choice based on the criteria just mentioned. This way, an environment of competition is cultivated among service providers where they compete with each other to become the data provider of choice.
- In the following diagram, varying levels of decentralization are shown. On the left side, the conventional approach is shown where a central system is in control; on the right side, complete disintermediation is achieved, as intermediaries are entirely removed. Competing intermediaries or service providers are shown in the center.
- At that level, intermediaries or service providers are selected based on reputation or voting, thus achieving partial decentralization.

b) Describe the various decentralized organizations in blockchain

Decentralized organizations DOs are software programs that run on a blockchain and are based on the idea of actual organizations with people and protocols.

- Once a DO is added to the blockchain in the form of a smart contract or a set of smart contracts, it becomes decentralized and parties interact with each other based on the code defined within the DO software.

Decentralized autonomous organizations DAOs

- Decentralized autonomous organization (DAO) is also a computer program that runs on top of a blockchain, and embedded within it are governance and business logic rules. DAOs and DOs are fundamentally the same thing. The main difference, however, is that DAOs are autonomous, which means that they are fully automated and contain artificially intelligent logic. DOs, on the other hand, lack this feature and rely on human input to execute business logic.
- Ethereum blockchain led the way with the introduction of DAOs. In a DAO, the code is considered the governing entity rather than people or paper contracts. However, a human curator maintains this code and acts as a proposal evaluator for the community. DAOs are capable of hiring external contractors if enough

input is received from the token holders (participants)

- The most famous DAO project is The DAO, which raised $168 million in its crowdfunding phase. The DAO project was designed to be a venture capital fund aimed at providing a decentralized business model with no single entity as owner. Unfortunately, this project was hacked due to a bug in the DAO code, and millions of dollars' worth of ether currency (ETH) was siphoned out of the project and into a child DAO created by hackers. A major network change (hard fork) was required on the Ethereum blockchain to reverse the impact of the hack and initiate the recovery of the funds. This incident opened up the debate on the security, quality, and need for thorough testing of the code in smart contracts in order to ensure their integrity and adequate control. There are other projects underway, especially in academia, that are seeking to formalize smart contract coding and testing

Decentralized autonomous corporations DACs

- Decentralized autonomous corporations (DACs) are similar to DAOs in concept, though considered to be a subset of them. The definitions of DACs and DAOs may sometimes overlap, but the general distinction is that DAOs are usually considered to be nonprofit, whereas DACs can earn a profit via shares offered to the participants and to whom they can pay dividends. DACs can run a business automatically without human intervention based on the logic programmed into them.

Decentralized autonomous societies (DASes)

- Decentralized autonomous societies (DASes) are a concept whereby an entire society can function on a blockchain with the help of multiple, complex smart contracts and a combination of DAOs and decentralized applications (DApps) running autonomously.
- This model does not necessarily translate to a free-for-all approach, nor is it based on an entirely libertarian ideology; instead, many services that a government commonly offers can be delivered via blockchains, such as government identity card systems, passports, and records of deeds, marriages, and births.
- Another theory is that, if a government is corrupt and central systems do not provide the levels of trust that a society needs, then that society can start its own virtual one on a blockchain that is driven by decentralized consensus and transparency. This concept might look like a libertarian's or cypherpunk's dream, but it is entirely possible on a blockchain.

DECENTRALIZED APPLICATIONS(DAPPS)

DApps at a fundamental level are software programs that execute using either of the following methods. They are categorized as Type 1, Type 2, or Type 3 DApps:

1.Type 1: Run on their own dedicated blockchain, for example, standard smart contract based DApps running on Ethereum.

- If required, they make use of a native token, for example, ETH on Ethereum blockchain. For example, Ethlance is a DApp that makes use of ETH to provide a job market.

2.Type 2: Use an existing established blockchain. that is, make use of Type 1 blockchain and bear custom protocols and tokens, for example, smart contract based tokenization DApps running Ethereum blockchain.

- An example is DAI, which is built on top of Ethereum blockchain, but contains its own stable coins and mechanism of distribution and control.
- Another example is Golem, which has its own token GNT and a transaction framework built on top of Ethereum blockchain to provide a decentralized marketplace for computing power where users share their computing power with each other in a peer-to-peer network.
- A prime example of Type 2 DApps is the OMNI network, which is a software layer built on top of Bitcoin to support trading of custom digital assets and digital currencies. More information on the OMNI network can be found at https://www.omnilayer. Org

3. Type 3: Use the protocols of Type 2 DApps; for example, the SAFE Network uses the OMNI network protocol.

- Another example to understand the difference between different types of DApps is the USDT token (Tethers).
- The original USDT uses the OMNI layer (a Type 2 DApp) on top of the Bitcoin network. USDT is also available on Ethereum using ERC20 tokens.
- This example shows that a USDT can be considered a Type 3 DApp, where the OMNI layer protocol (a Type 2 DApp) is used, which is itself built on Bitcoin (a Type 1 DApp).
- Also, from an Ethereum point of view USDT can also be considered a Type 3 DApp in that it makes use of the Type 1 DApp Ethereum blockchain using the ERC 20 standard, which was built to operate on Ethereum.

3. a)Explain Elliptic curve cryptography.

Elliptic curves:

The elliptic curves algorithm is based on the discrete logarithm problem discussed previously, but in the context of elliptic curves. An elliptic curve is an algebraic cubic curve over a field, which can be defined by an equation, as shown here.

The curve is non-singular, which means that it has no cusps or self-intersections. It has two variables a and b, as well as a point of infinity:

$y2 = x3+ax+b3$ Here, a and b are integers whose values are elements of the field on which the elliptic curve is defined.

Elliptic curves can be defined over reals, rational numbers, complex numbers, or finite fields.

For cryptographic purposes, an elliptic curve over prime finite fields is used instead of real numbers. Additionally, the prime should be greater than 3. Different curves can be generated by varying the values of a and/or b.

The most prominently used cryptosystems based on elliptic curves are the Elliptic Curve Digital Signatures Algorithm (ECDSA) and the Elliptic Curve Diffie-Hellman (ECDH) key exchange.

Elliptic curve cryptography

•ECC is based on the discrete logarithm problem founded upon elliptic curves over finite fields (Galois fields).

•The main benefit of ECC over other types of public key algorithms is that it requires a smaller key size, while providing the same level of security as, for example, RSA. Two notable schemes that originate from ECC are ECDH for key exchange and ECDSA for digital signatures. ECC can also be used for encryption, but it is not usually used for this purpose in practice. Instead, key exchange and digital signatures are used more commonly. As ECC needs less space to operate, it is becoming very popular on embedded platforms and in systems where storage resources are limited. By comparison, the same level of security can be achieved with ECC when only using 256-bit operands as compared to 3,072 bits in RSA.

•Maths behind ECC

To understand ECC, a basic introduction to the underlying mathematics is necessary. An elliptic curve is basically a type of polynomial equation known as the Weierstrass equation, which generates a curve over a finite field. The most commonly used field is where all arithmetic operations are performed modulo a prime number p. Elliptic curve groups consist of points on the curve over a finite field. An elliptic curve is defined in the following equation: The following equation:

Y2 =x3+Ax+B mod P

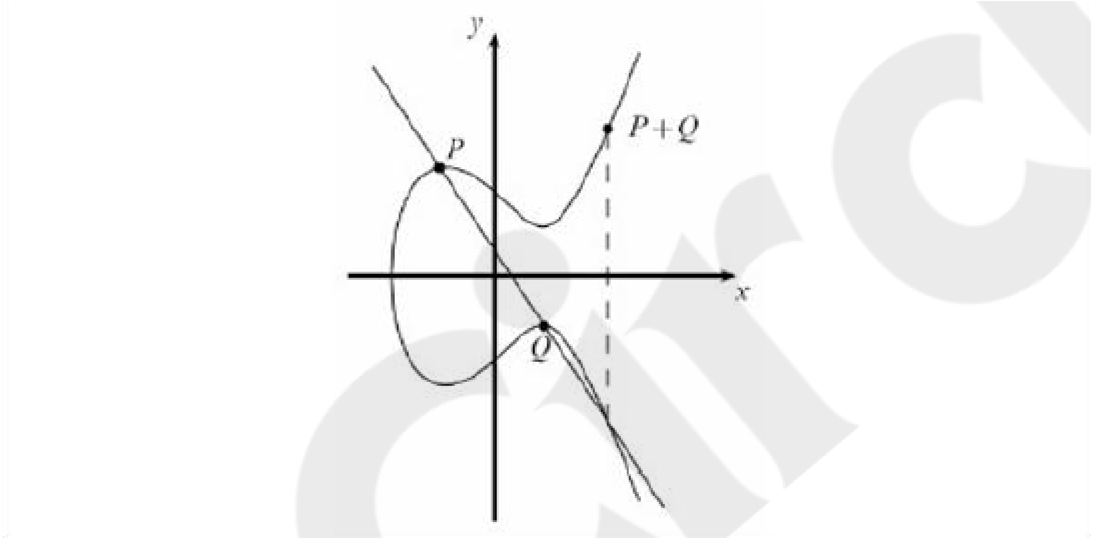
Here, A and B belong to a finite field Zp or Fp (prime finite field), along with a special value called the point of infinity. The point of infinity ∞ is used to provide identity operations for points on the curve

Basic Group operations on elliptic curves are point addition and point doubling. Point addition is a process where two different points are added, and point doubling means that the same point is added to itself.

Point addition

•Point addition is shown in the following diagram. This is a geometric representation of point addition on elliptic curves.

•In this method, a diagonal line is drawn through the curve that intersects the curve at two points shown below P and Q, which yields a third point between the curve and the line.



The group operation denoted by the + sign for addition yields the following equation: P + Q = R

In this case, two points are added to compute the coordinates of the third point on the curve: P + Q = R

More precisely, this means that coordinates are added, as shown in the following equation: $(x1 , y1) + (x2 , y2) = (x3 , y3)$

The equation of point addition is as follows: $X3 = s2 – x1 – x2 \bmod p$

$y3 = s (x1 – x3)-y1 \bmod p$

b)What are smart contracts?

Smart contracts

•A smart contract is a software program that usually runs on a blockchain. Smart contracts do not necessarily need a blockchain to run; however, due to the security benefits that blockchain technology provides, blockchain has become a standard decentralized execution platform for smart contracts.

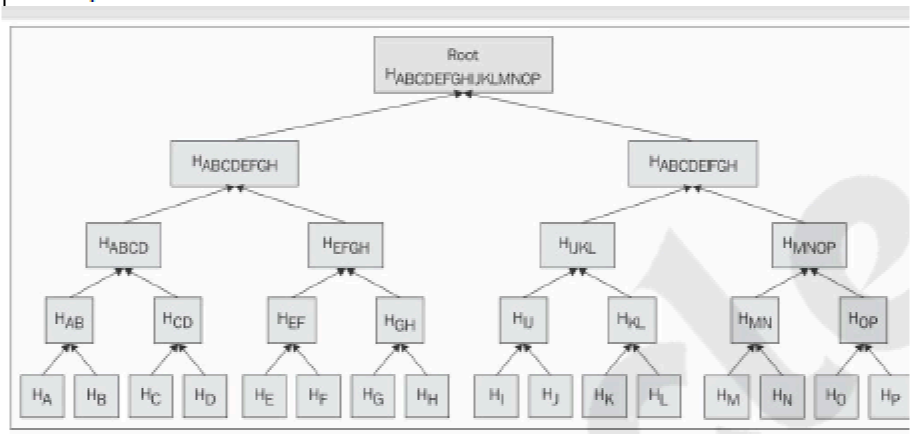
•A smart contract usually contains some business logic and a limited amount of data. The business logic is executed if specific criteria are met. Actors or participants in the blockchain use these smart contracts, or they run autonomously on behalf of the network participants

4. a) Explain Merkle tree with the help of a neat diagram. Explain the use of Merkle tree in blockchain

Merkle trees

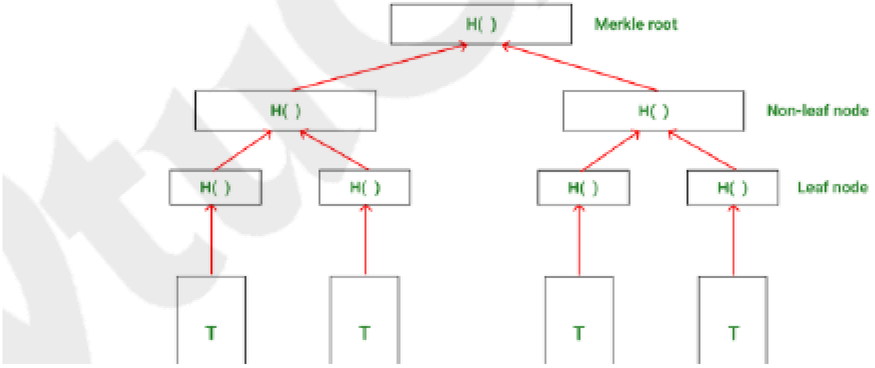
The concept of Merkle tree was introduced by Ralph Merkle. A visualization of Merkle tree is shown here, which makes it easy to understand. Merkle trees allow secure and efficient verification of large data sets. It is a binary tree in which first, the inputs are placed at the leaves (node with no children), and then values of pairs of child nodes are hashed together in order to produce a value for the parent node (internal node) until a single hash value known as Merkle root is achieved.

An example of a Merkle tree:



A Merkle tree

Merkle Tree Structure:



1. A blockchain can potentially have thousands of blocks with thousands of transactions in each block. Therefore, memory space and computing power are two main challenges.

2. It would be optimal to use as little data as possible for verifying transactions, which can reduce CPU processing and provide better security, and this is exactly what Merkle trees offer.

3. In a Merkle tree, transactions are grouped into pairs. The hash is computed for each pair and this is stored in the parent node. Now the parent nodes are grouped into pairs and their hash is stored one level up in the tree. This continues till the root of the tree. The different types of nodes in a Merkle tree are:

•Root node: The root of the Merkle tree is known as the Merkle root and this Merkle root is stored in the header of the block.

•Leaf node: The leaf nodes contain the hash values of transaction data. Each transaction in the block has its data hashed and then this hash value (also known as transaction ID) is stored in leaf nodes.

•Non-leaf node: The non-leaf nodes contain the hash value of their respective children. These are also called intermediate nodes because they contain the intermediate hash values and the hash process continues till the root of the tree.

4. Bitcoin uses the SHA-256 hash function to hash transaction data continuously till the Merkle root is obtained.

5. Further, a Merkle tree is binary in nature. This means that the number of leaf nodes needs to be even for the Merkle tree to be constructed properly. In case there is an odd number of leaf nodes, the tree duplicates the last hash and makes the number of leaf nodes even.

How do Merkle trees Work?

•A Merkle tree is constructed from the leaf nodes level all the way up to the Merkle root level by grouping nodes in pairs and calculating the hash of each pair of nodes in that particular level. This hash value is propagated to the next level. This is a bottom-to-up type of construction where the hash values are flowing from down to up direction.

•Hence, by comparing the Merkle tree structure to a regular binary tree data structure, one can observe that Merkle trees are actually inverted down.[refer above merkle tree example]

b)List the major differences between symmetric and asymmetric cryptography

Symmetric cryptography

There are two types of symmetric ciphers: stream ciphers and block ciphers. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are typical examples of block ciphers, whereas RC4 and A5 are commonly used stream ciphers.

Stream ciphers Stream ciphers are encryption algorithms that apply encryption algorithms on a bit-by-bit basis (one bit at a time) to plaintext using a keystream.

•There are two types of stream ciphers: synchronous stream ciphers and asynchronous stream ciphers.

•Synchronous stream ciphers are those where the keystream is dependent only on the key.

•Asynchronous stream ciphers have a keystream that is also dependent on the encrypted data.

•In stream ciphers, encryption and decryption are the same function because they are simple modulo 2 additions or XOR operations. The fundamental requirement in stream ciphers is the security and randomness of keystreams. Various techniques ranging from PRNGs to true hardware RNGs have been developed to generate random numbers, and it is vital that all key generators be cryptographically secure.

Block ciphers

•Block ciphers are encryption algorithms that break up the text to be encrypted (plaintext) into blocks of a fixed length and apply the encryption block by block. Block ciphers are generally built using a design strategy known as a Feistel cipher. Recent block ciphers such as AES (Rijndael) have been built using a combination of substitution and permutation called a Substitution-Permutation Network (SPN)

Feistel ciphers are based on the Feistel network, which is a structure developed by Horst Feistel. This structure is based on the idea of combining multiple rounds of repeated operations to achieve desirable cryptographic properties known as confusion and diffusion. Feistel networks operate by dividing data into two blocks (left and right) and processing these blocks via keyed round functions in iterations to provide sufficient pseudorandom permutations.

•Confusion adds complexity to the relationship between the encrypted text and plaintext. This is achieved by substitution. In practice, A in plaintext is replaced by X in encrypted text. In modern cryptographic algorithms, substitution is performed using lookup tables called S-boxes.

ASYMMETRIC CRYPTOGRAPHY:

the concepts and practical aspects of public key cryptography, also called asymmetric cryptography or asymmetric key cryptography

MATHEMATICS:

1. Modular arithmetic Also known as clock arithmetic, numbers in modular arithmetic wrap around when they reach a certain fixed number. This fixed number is a positive number called modulus (sometimes abbreviated to mod), and all operations are performed concerning this fixed number.

•Modular arithmetic is analogous to a 12-hour clock; there are numbers from 1 to 12. When 12 is reached, the numbers start from 1 again. Imagine that the time is 9:00 now; 4 hours from now, it will be 1:00 because the numbers wrap around at 12 and start from 1 again.

•In normal addition, this would be 9 + 4 = 13, but that is not the case on a 12-hour clock; it is 1:00. In other words, this type of arithmetic deals with the remainders after the division operation. For example, 50 mod 11 is 6 because 50 / 11 leaves a remainder of 6

Asymmetric Cryptography

•Asymmetric cryptography refers to a type of cryptography where the key that is used to encrypt the data is different from the key that is used to decrypt the data. These keys are called private and public keys, respectively, which why asymmetric cryptography is also known as public key cryptography. It uses both public and private keys to encrypt and decrypt data, respectively.

5) a) What is the Byzantine generals problem, and how is it relevant to blockchain design

BYZANTINE GENERAL PROBLEM

History:

Before discussing consensus in distributed systems, events in history are presented that are precursors to the development of successful and practical consensus mechanisms.

In September 1962, Paul Baran introduced the idea of cryptographic signatures with his paper On distributed communications networks. This is the paper where the concept of decentralized networks was also introduced for the very first time.

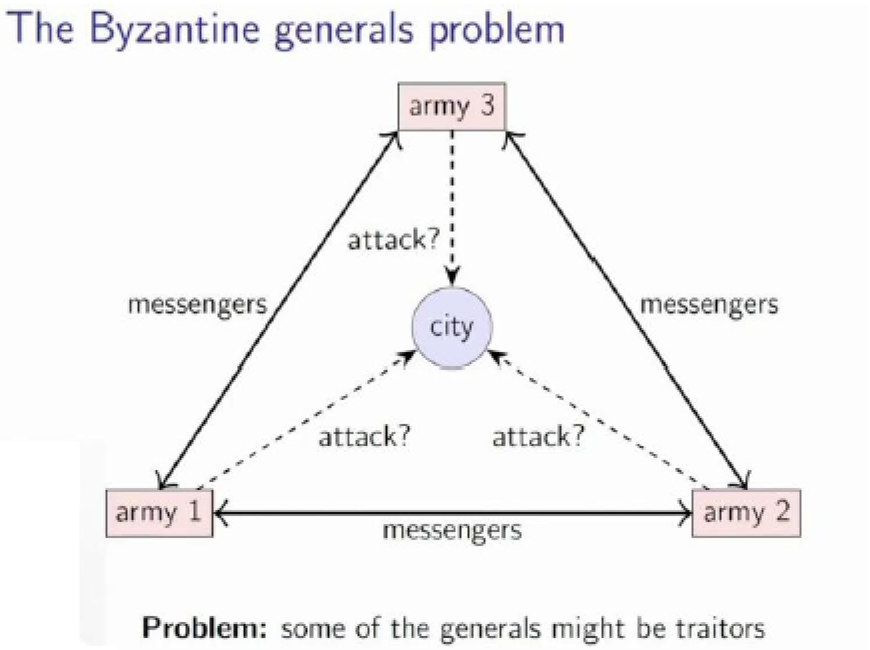
What is Byzantine General's Problem?

In 1982, The Byzantine General's Problem was invented by Leslie Lamport, Robert Shostak, and Marshall Pease. Byzantine Generals Problem is an impossibility result which means that the solution to this problem has not been found yet as well as helps us to understand the importance of blockchain. It is basically a game theory problem that provides a description of the extent to which decentralized parties experience difficulties in reaching consensus without any trusted central parties.

The Byzantine army is divided into many battalions in this classic problem called the Byzantine General's problem, with each division led by a general.

The generals connect via messenger in order to agree to a joint plan of action in which all battalions coordinate and attack from all sides in order to achieve success.

It is probable that traitors will try to sabotage their plan by intercepting or changing the messages. As a result, the purpose of this challenge is for all of the faithful commanders to reach an agreement without the imposters tampering with their plans



The Byzantine generals problem

Problem: some of the generals might be traitors

Byzantine Fault Tolerance (BFT) This problem was solved in 1999 by Castro and Liskov who presented the Practical Byzantine Fault Tolerance (PBFT) algorithm. Later on in 2009, the first practical implementation was made with the invention of bitcoin where the Proof of Work (PoW) algorithm was developed as a mechanism to achieve consensus

-

The Byzantine Fault Tolerance was developed as inspiration in order to address the Byzantine General's Problem. The Byzantine General's Problem, a logical thought experiment where multiple generals must attack a city, is where the idea for BFT originated.

-

Byzantine Fault Tolerance is one of the core characteristics of developing trustworthy blockchain rules or features is tolerance.

-

When two-thirds of the network can agree or reach a consensus and the system still continues to operate properly, it is said to have BFT.

-

Blockchain networks' most popular consensus protocols, such as proof-of-work, proof-of-stake, and proof-of-authority, all have some BFT characteristics.

In order to create a decentralized network, the BFT is essential.

-

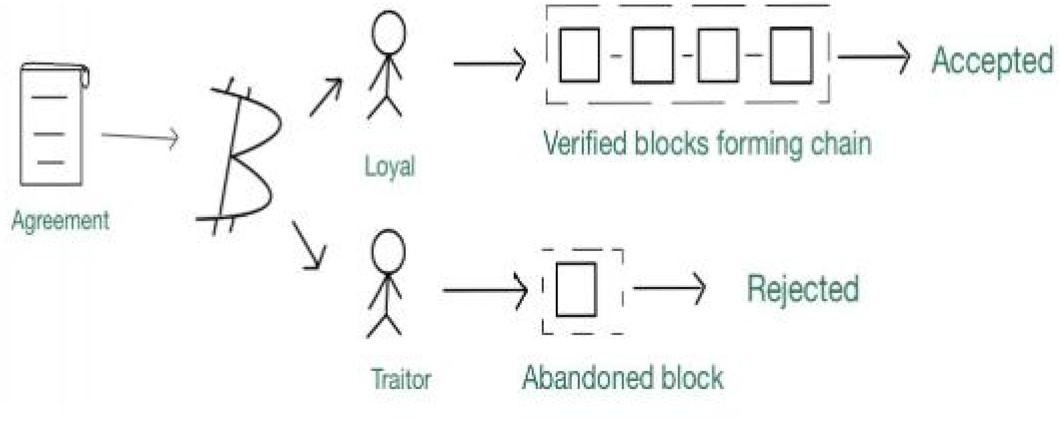
The consensus method determines the precise network structure. For instance, BFT has a leader as well as peers who can and cannot validate.

-

In order to maintain the sequence of the Blockchain SC transactions and the consistency of the global state through local transaction replay, consensus messages must pass between the relevant peers.

-

More inventive approaches to designing BFT systems will be found and put into practice as more individuals and companies investigate distributed and decentralized systems. Systems that use BFT are also employed in sectors outside of blockchains, such as nuclear power, space exploration, and aviation.



b)If your blockchain network has 6 Byzantine nodes, what is the minimum number of nodes that are required to ensure Byzantine fault tolerance using PBFT protocol?

To determine the **minimum number of nodes required** to tolerate **6 Byzantine (faulty) nodes** using the **Practical Byzantine Fault Tolerance (PBFT)** protocol, we use the formula:

$n >= 3f+1$

Where:

- n= total number of nodes
- f = number of Byzantine faulty nodes

Given:

- Given $f=6$

Plug into the formula:

$n \geq 3(6) + 1 = 18 + 1 = 19$

At least 19 nodes are required to tolerate 6 Byzantine nodes using the PBFT protocol.

6) Explain SHA 256 algorithm

Secure Hash Algorithms The following list describes the most common Secure Hash Algorithms (SHAs):

- SHA-0: This is a 160-bit function introduced by the U.S. National Institute of Standards and Technology (NIST) in 1993.
- SHA-1: SHA-1 was introduced in 1995 by NIST as a replacement for SHA-0. This is also a 160-bit hash function. SHA-1 is used commonly in SSL and TLS implementations. It should be noted that SHA-1 is now considered insecure, and it is being deprecated by certificate authorities. Its usage is discouraged in any new implementations.

-
SHA-2: This category includes four functions defined by the number of bits of the hash: SHA-224, SHA-256, SHA-384, and SHA-512. SHA-3: This is the latest family of SHA functions. SHA3-224, SHA3-256, SHA3-384, and SHA3-512 are members of this family. SHA3 is a NIST-standardized version of Keccak.

-
Keccak uses a new approach called sponge construction instead of the commonly used Merkle Damgard transformation

DESIGN OF SECURE HASH ALGORITHMS (SHA)

Design of SHA-256

-

SHA-256 has an input message size limit of 264 - 1 bits. The block size is 512 bits, and it has a word size of 32 bits. The output is a 256-bit digest. The compression function processes a 512-bit message block and a 256-bit intermediate hash value.

There are two main components of this function: the compression function and a message schedule.

-

The algorithm works as follows, in nine steps:

Pre-processing

1. Padding of the message is used to adjust the length of a block to 512 bits if it is smaller than the required block size of 512 bits.

2. Parsing the message into message blocks, which ensures that the message and its padding is divided into equal blocks of 512 bits.

3. Setting up the initial hash value, which consists of the eight 32-bit words obtained by taking the first 32 bits of the fractional parts of the square roots of the first eight prime numbers. These initial values are fixed and chosen to initialize the process. They provide a level of confidence that no backdoor exists in the algorithm.

Hash computation

4. Each message block is then processed in a sequence, and it requires 64 rounds to compute the full hash output. Each round uses slightly different constants to ensure that no two rounds are the same.

5. The message schedule is prepared.

6. Eight working variables are initialized.

7. The compression function runs 64 times.

8. The intermediate hash value is calculated.

9. Finally, after repeating steps 5 through 8 until all blocks (chunks of data) in the input message are processed, the output hash is produced by concatenating intermediate hash values

SHA-256 is a Merkle Damgard construction that takes the input message and divides it into equal blocks (chunks of data) of 512 bits. Initial values (or initial hash values) or the initialization vector are composed of eight 32bit words (256 bits) that are fed into the compression function with the first message. Subsequent blocks are fed into the compression function until all blocks are processed and finally, the output hash is produced. The compression function of SHA-256 is shown in the following diagram

At a high level, SHA-256 can be visualized in the following diagram:

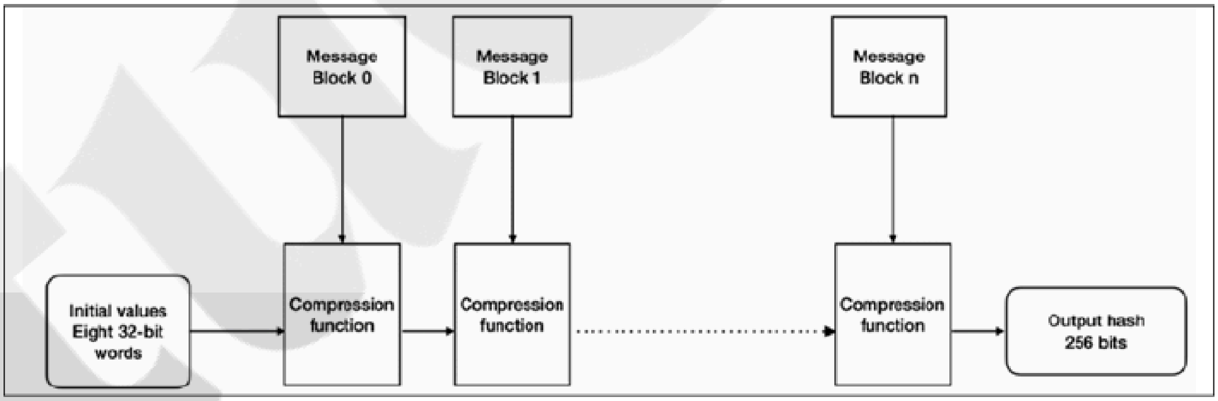


Figure 3.4: SHA-256 high level overview

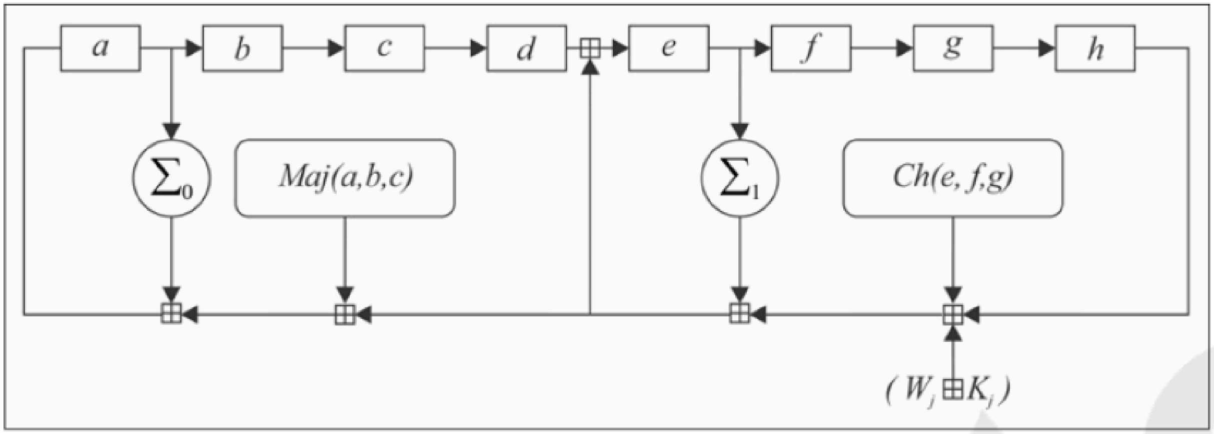


Figure 3.5: One round of an SHA-256 compression function