PVP 20

Code: 20CS4601C

III B.Tech - II Semester – Regular / Supplementary Examinations APRIL 2024

BLOCKCHAIN TECHNOLOGY
(COMPUTER SCIENCE & ENGINEERING)

Scheme of Valuation

UNIT-I

1.a) Explain the required technologies in block chain implementation. (7 M)

----Explanation of at least 3 technologies -- 7 Marks

1.b) Discuss how Merkle Trees are used to efficiently verify the integrity of data within a block. (7 M)

----Definition and relevant explanation – 7 M

OR

2.a) Illustrate and explain block chain architecture. (7 M)

----Explanation of block chain and its components—7 M

2.b) Define consensus algorithms in the context of block chain and their importance in reaching agreement among network participants. (7 M)

---Explanation of any three algorithms—7 Marks

UNIT-II

3.a) How does block chain enable decentralization in practice? (7 M)

---Explanation on decentralization –7 Marks

3.b) Explain about contest-driven decentralization. (7 M)

---Explanation on contest-driven decentralization—7 Marks

OR

4.a) Explain in detail about centralized, decentralized and distributed systems with a neat diagram? (7 M)

---- Explanation ---7 Marks

4.b) Explain the major challenges in the decentralization of block chain technology.(7 M)

--- Any three challenges – 7 Marks

UNIT-III

5.a) **Explain about Asymmetric Cryptography in Block chain?** **(7 M)**

---- Explanation – 7 Marks

5.b) **Explain the working functionality of mining algorithm in Bitcoin.** **(7 M)**

--- Explanation with steps – 7 marks

OR

6.a) **Explain any two block ciphers with the example scenarios.** **(7 M)**

--- Explanation of any two symmetric or public key cryptographic mechanisms such as below can be awarded marks--- 7 Marks

6.b) **What are the hash rate mining systems? Explain any two mining systems.** **(7 M)**

---- Definition ---3 Marks

----- Any two mining systems—4 Marks

UNIT-IV

7.a) **Explain the life cycle of a smart contract.** **(7 M)**

--- Explanation --- 7 Marks

7.b) **What is Ethereum network? Explain the components of the Ethereum ecosystem.** **(7 M)**

---definition – 3Marks

--- explanation of any two elements/components– 4 Marks

OR

8.a) **Explain about execution environment in Ethereum virtual machine.** **(7 M)**

--- Explanation – 7 Marks

8.b) **Explain the operations of a DApp in Ethereum.** **(7 M)**

--- Explanation of DApp --- 7 Marks

UNIT-V

9.a) **Explain the fundamental components of the Hyperledger reference architecture.** **(7 M)**

--- Explantion – 4 Marks

--- Diagram – 3 Marks

9.b) **Describe about Hyperledger Frabric in Detail.** **(7 M)**

---Explanation – 7 Marks

OR

10.a) What is Hyperledger, and what distinguishes it from other block chain platforms? Explain its requirements & Design goals. (7 M)

--- Definition – 2 Marks

--- Explanation of Requirements & Design goals (Any three) – 5 Marks

10.b) How does block chain Quorum address the specific needs of industries beyond crypto- currency, such as finance or supply chain management? (7 M)

--- Explantion of any one --- 7 Marks

PVP 20

Code: 20CS4601C

III B.Tech - II Semester – Regular / Supplementary Examinations APRIL 2024

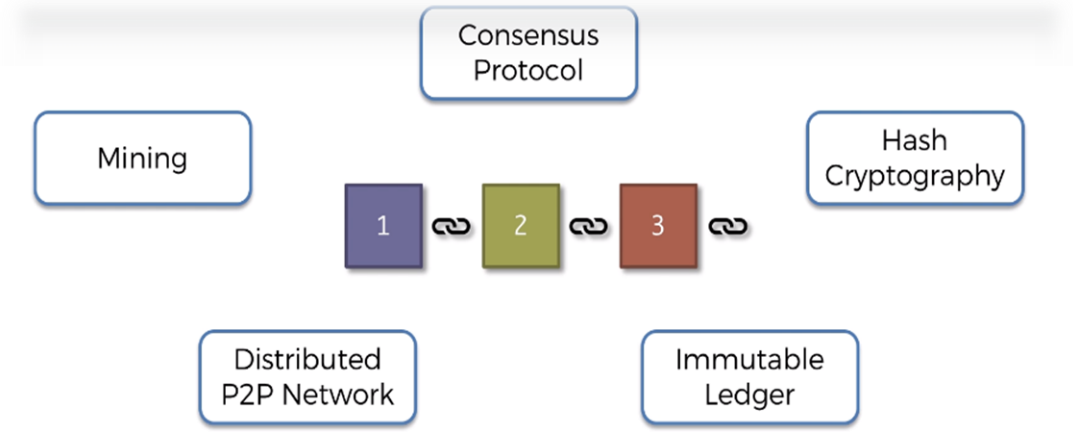
BLOCKCHAIN TECHNOLOGY
(COMPUTER SCIENCE & ENGINEERING)
Scheme of Valuation

UNIT-I

1.a) Explain the required technologies in block chain implementation. (7 M)

Sol:

Blockchain is a peer-to-peer, distributed ledger that is cryptographically-secure, append-only, immutable (extremely hard to change), and updateable only via consensus or agreement among peers. Various technologies were combined to create what now is known as blockchain. This concept can also be visualized with the help of the following diagram:



Peer-to-peer

The first keyword in the technical definition is peer-to-peer. This means that there is no central controller in the network, and all participants talk to each other directly. This property allows for cash transactions to be exchanged directly among the peers without a third-party involvement, such as by a bank.

Distributed ledger

Dissecting the technical definition further reveals that blockchain is a distributed ledger, which simply means that a ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger.

Cryptographically-secure

Next, we see that this ledger is cryptographically-secure, which means that cryptography has been used to provide security services which make this ledger secure against tampering and misuse. These services include non-repudiation, data integrity, and data origin authentication.

Append-only

Another property that we encounter is that blockchain is append-only, which means that data can only be added to the blockchain in time-ordered sequential order. This property implies that once data is added to the blockchain, it is almost impossible to change that data and can be considered practically immutable. Nonetheless, it can be changed in rare scenarios wherein collusion against the blockchain network succeeds in gaining more than 51 percent of the power.

Updateable via consensus

Finally, the most critical attribute of a blockchain is that it is updateable only via consensus. This is what gives it the power of decentralization. In this scenario, no central authority is in control of updating the ledger. Instead, any update made to the blockchain is validated against strict criteria defined by the blockchain protocol and added to the blockchain only after a consensus has been reached among all participating peers/nodes on the network. To achieve consensus, there are various consensus facilitation algorithms which ensure that all parties are in agreement about the final state of the data on the blockchain network and resolutely agree upon it to be true.

1.b) Discuss how Merkle Trees are used to efficiently verify the integrity of data within a block.(7 M)

Sol:

Merkle root is a hash of all of the nodes of a Merkle tree. Merkle trees are widely used to validate the large data structures securely and efficiently. In the blockchain world, Merkle trees are commonly used to allow efficient verification of transactions. Merkle root in a blockchain is present in the block header section of a block, which is the hash of all transactions in a block. This means that verifying only the Merkle root is required to verify all transactions present in the Merkle tree instead of verifying all transactions one by one.

2.a) Illustrate and explain block chain architecture.(7 M)

Sol:

In the case of the distributed network of blockchain architecture, each participant within the network maintains, approves, and updates new entries. The system is controlled not only by separate individuals, but by everyone within the blockchain network. Each member ensures that all records and procedures are in order, which results in data validity and security. Thus, parties that do not necessarily trust each other are able to reach a common consensus.

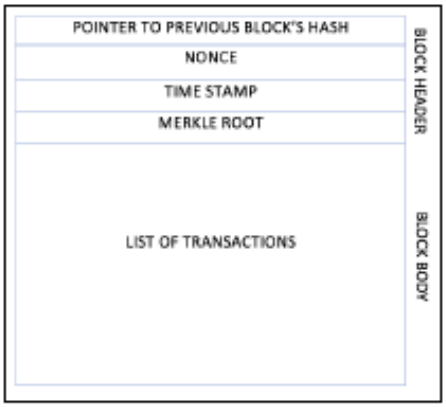
These are the core blockchain architecture components:

- Node - user or computer within the blockchain architecture (each has an independent copy of the whole blockchain ledger)
- Transaction - smallest building block of a blockchain system (records, information, etc.) that serves as the purpose of blockchain
- Block - a data structure used for keeping a set of transactions which is distributed to all nodes in the network
- Chain - a sequence of blocks in a specific order

- Miners - specific nodes which perform the block verification process before adding anything to the blockchain structure
- Consensus (consensus protocol) - a set of rules and arrangements to carry out blockchain operations

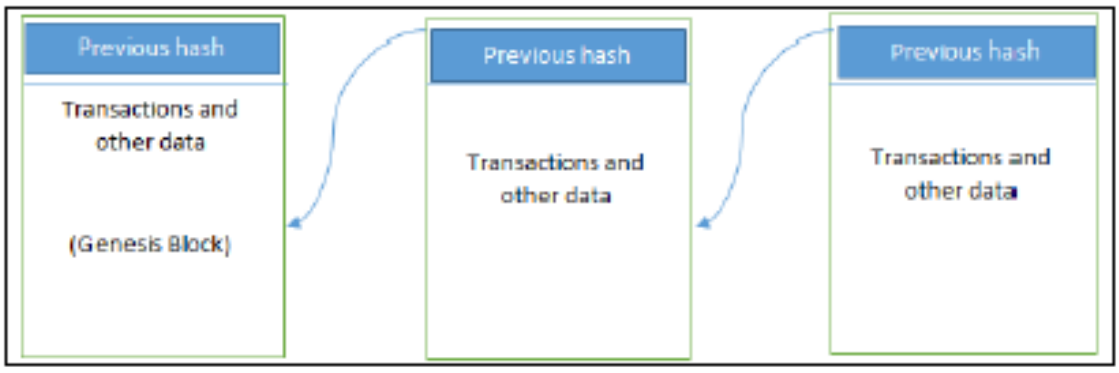
The structure of blockchain technology is represented by a list of blocks with transactions in a particular order. These lists can be stored as a flat file (txt. format) or in the form of a simple database. Two vital data structures used in blockchain include:

- Pointers - variables that keep information about the location of another variable. Specifically, this is pointing to the position of another variable.
- Linked lists - a sequence of blocks where each block has specific data and links to the following block with the help of a pointer.



Logically, the first block does not contain the pointer since this one is the first in a chain. At the same time, there is potentially going to be a final block within the blockchain database that has a pointer with no value.

Basically, the following blockchain sequence diagram is a connected list of records:



Generic structure of a blockchain

Blockchain architecture can serve the following purposes for organizations and enterprises:

- Cost reduction - lots of money is spent on sustaining centrally held databases (e.g. banks, governmental institutions) by keeping data current secure from cyber crimes and other corrupt intentions.

- History of data - within a blockchain structure, it is possible to check the history of any transaction at any moment in time. This is a ever-growing archive, while a centralized database is more of a snapshot of information at a specific point.
- Data validity & security - once entered, the data is hard to tamper with due to the blockchain's nature. It takes time to proceed with record validation, since the process occurs in each independent network rather than via compound processing power. This means that the system sacrifices performance speed, but instead guarantees high data security and validity.

Types of Blockchain Architecture

All blockchain structures fall into three categories:

- Public blockchain architecture: A public blockchain architecture means that the data and access to the system is available to anyone who is willing to participate (e.g. Bitcoin, Ethereum, and Litecoin blockchain systems are public).
- Private blockchain architecture: As opposed to public blockchain architecture, the private system is controlled only by users from a specific organization or authorized users who have an invitation for participation.
- Consortium blockchain architecture: This blockchain structure can consist of a few organizations. In a consortium, procedures are set up and controlled by the preliminary assigned users.

2.b) Define consensus algorithms in the context of block chain and their importance in reaching agreement among network participants. (7 M)

Sol: <u>(Description of any three algorithms mentioned below)</u>

The consensus algorithms available today, or that are being researched in the context of blockchain, are presented here. The following is not an exhaustive list, but it includes all notable algorithms:

- Proof of Work (PoW): This type of consensus mechanism relies on proof that adequate computational resources have been spent before proposing a value for acceptance by the network. This scheme is used in Bitcoin, Litecoin, and other cryptocurrency blockchains. Currently, it is the only algorithm that has proven to be astonishingly successful against any collusion attacks on a blockchain network, such as the Sybil attack.
- Proof of Stake (PoS): This algorithm works on the idea that a node or user has an adequate stake in the system; that is, the user has invested enough in the system so that any malicious attempt by that user would outweigh the benefits of performing such an attack on the network. This idea was first introduced by Peercoin, and it is going to be used in the Ethereum blockchain version called Serenity. Another important concept in PoS is coin age, which is a criterion derived from the amount of time and number of coins that have not been spent. In this model, the chances of proposing and signing the next block increase with the coin age.
- Delegated Proof of Stake (DPoS): This is an innovation over standard PoS, whereby each node that has a stake in the system can delegate the validation of a transaction to other nodes by voting. It is used in the BitShares blockchain.
- Proof of Elapsed Time (PoET): Introduced by Intel in 2016, PoET uses a Trusted Execution Environment (TEE) to provide randomness and safety in the leader election

process via a guaranteed wait time. It requires the Intel Software Guard Extensions (SGX) processor to provide the security guarantee for it to be secure.

- Proof of Deposit (PoD): In this case, nodes that wish to participate in the network have to make a security deposit before they can mine and propose blocks. This mechanism is used in the Tendermint blockchain.
- Proof of Importance (PoI): This idea is significant and different from PoS. PoI not only relies on how large a stake a user has in the system, but it also monitors the usage and movement of tokens by the user in order to establish a level of trust and importance. It is used in the NEM coin blockchain.
- Federated consensus or federated Byzantine consensus: This mechanism is used in the stellar consensus protocol. Nodes in this protocol retain a group of publicly-trusted peers and propagate only those transactions that have been validated by the majority of trusted nodes.
- Reputation-based mechanisms: As the name suggests, a leader is elected by the reputation it has built over time on the network. It is based on the votes of other members.
- PBFT: This mechanism achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols including PBFT, PAXOS, RAFT, and Federated Byzantine Agreement (FBA) are also being used or have been proposed for use in many different implementations of distributed systems and blockchains.
- Proof of Activity (PoA): This scheme is a combination of PoS and PoW, which ensures that a stakeholder is selected in a pseudorandom but uniform fashion. This is a comparatively more energy-efficient mechanism as compared to PoW. It utilizes a new concept called Follow the Satoshi. In this scheme, PoW and PoS are combined together to achieve consensus and good level of security. This scheme is more energy efficient as PoW is used only in the first stage of the mechanism, after the first stage it switches to PoS which consumes negligible energy.
- Proof of Capacity (PoC): This scheme uses hard disk space as a resource to mine the blocks. This is different from PoW, where CPU resources are used. In in PoC, hard disk space is utilized for mining and as such is also known as hard drive mining. This concept was first introduced in the Burstcoin cryptocurrency.
- Proof of Storage (PoS): This scheme allows for the outsourcing of storage capacity. This scheme is based on the concept that a particular piece of data is probably stored by a node which serves as a means to participate in the consensus mechanism. Several variations of this scheme have been proposed, such as Proof of Replication, Proof of Data Possession, Proof of Space, and Proof of Space-Time.

UNIT-II

3.a) How does block chain enable decentralization in practice? (7 M)

Sol:

Decentralization is a core benefit and service provided by blockchain technology. By design, blockchain is a perfect vehicle for providing a platform that does not need any intermediaries and that can function with many different leaders chosen via consensus mechanisms. This model allows anyone to compete to become the decision-making authority. This competition is governed by a consensus mechanism, and the most commonly used method is known as Proof of Work (PoW).

Decentralization is applied in varying degrees from a semi-decentralized model to a fully decentralized one depending on the requirements and circumstances. Decentralization can be

viewed from a blockchain perspective as a mechanism that provides a way to remodel existing applications and paradigms, or to build new applications, in order to give full control to users.

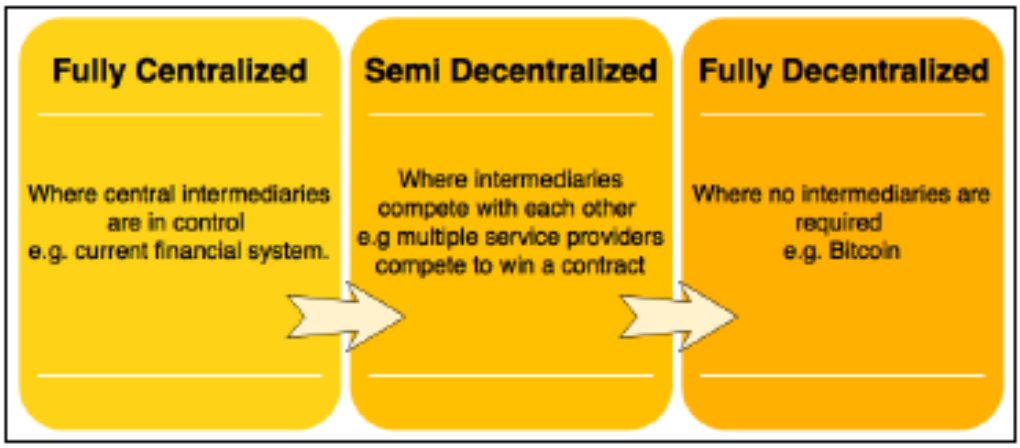
Information and Communication Technology (ICT) has conventionally been based on a centralized paradigm whereby database or application servers are under the control of a central authority, such as a system administrator. With Bitcoin and the advent of blockchain technology, this model has changed and now the technology exists, which allows anyone to start a decentralized system and operate it with no single point of failure or single trusted authority. It can either be run autonomously or by requiring some human intervention, depending on the type and model of governance used in the decentralized application running on blockchain.

3.b) Explain about contest-driven decentralization. (7 M)

Sol:

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-hand side, the conventional approach is shown where a central system is in control; on the righthand 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.



Scale of decentralization

4.a) Explain in detail about centralized, decentralized and distributed systems with a neat diagram? (7 M)

Sol:

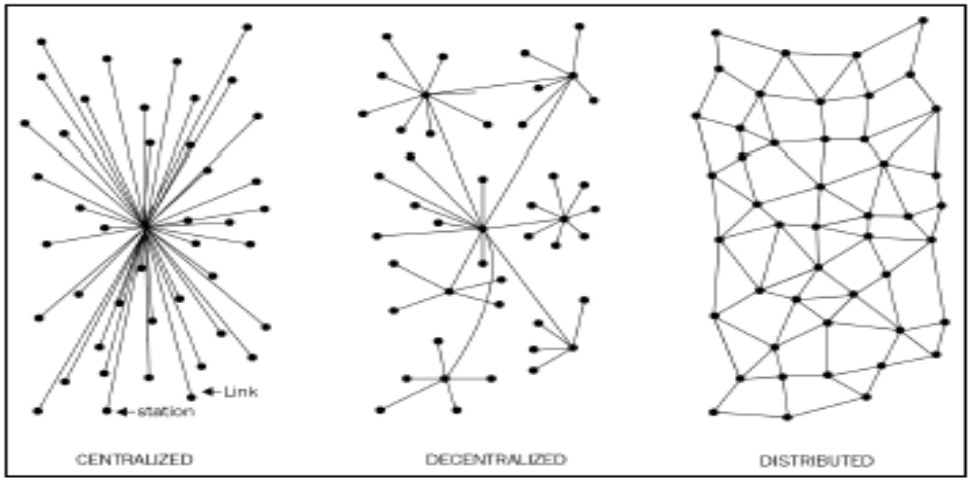
Centralized systems are conventional (client-server) IT systems in which there is a single authority that controls the system, and who is solely in charge of all operations on the system. All users of a centralized system are dependent on a single source of service. The majority of online service providers including Google, Amazon, eBay, Apple's App Store, and others use this conventional model for delivering services.

A distributed system, data and computation are spread across multiple nodes in the network. Sometimes, this term is confused with parallel computing. While there is some overlap in the definition, the main difference between these systems is that in a parallel computing system, computation is performed by all nodes simultaneously in order to achieve the result; for example, parallel computing platforms are used in weather research and forecasting, simulation and financial modeling. On the other hand, in a distributed system, computation may not happen in parallel and data is replicated across multiple nodes that users view as a single, coherent system. Variations of both of these models are used with to achieve fault tolerance and speed. In the parallel system model, there is still a central authority that has control over all nodes, which governs processing. This means that the system is still centralized in nature.

The critical difference between a decentralized system and distributed system is that in a distributed system, there still exists a central authority that governs the entire system; whereas, in a decentralized system, no such authority exists.

A decentralized system is a type of network where nodes are not dependent on a single master node; instead, control is distributed among many nodes. This is analogous to a model where each department in an organization is in charge of its own database server, thus taking away the power from the central server and distributing it to the subdepartments who manage their own databases.

A significant innovation in the decentralized paradigm that has given rise to this new era of decentralization of applications is decentralized consensus. This mechanism came into play with Bitcoin, and it enables a user to agree on something via a consensus algorithm without the need for a central, trusted third party, intermediary, or service provider.



Different types of networks/systems

4.b) Explain the major challenges in the decentralization of block chain technology.(7 M)

Sol:

In a decentralized system such as Bitcoin or Ethereum where security is normally provided by private keys, how can one ensure that a smart property associated with these private keys cannot be rendered useless if the private keys are lost or, due to a bug in the smart contract code or the decentralized application becomes vulnerable to attack? Before embarking on a journey to decentralize everything using blockchain and decentralized applications, it is essential that you understand that not everything can or needs to be decentralized.

Bitcoin and Cryptocurrency Technologies, Princeton University Press, that can be used to evaluate the decentralization requirements of a variety of issues in the context of blockchain technology. The framework raises four challenges/questions whose answers provide a clear understanding as to how a system can be decentralized:

1. What is being decentralized?
2. What level of decentralization is required?
3. What blockchain is used?
4. What security mechanism is used?

The first question simply asks you to identify what system is being decentralized. This can be any system, such as an identity system or a trading system.

The second question asks you to specify the level of decentralization required by examining the scale of decentralization as discussed earlier. It can be full disintermediation or partial disintermediation.

The third question asks developers to determine which blockchain is suitable for a particular application. It can be Bitcoin blockchain, Ethereum blockchain, or any other blockchain that is deemed fit for the specific application.

Finally, a fundamental question that needs to be addressed is how the security of a decentralized system will be guaranteed. For example, the security mechanism can be atomicity-based, where either the transaction executes in full or does not execute at all. This deterministic approach ensures the integrity of the system. Other mechanisms may include one based on reputation, which allows for varying degrees of trust in a system.

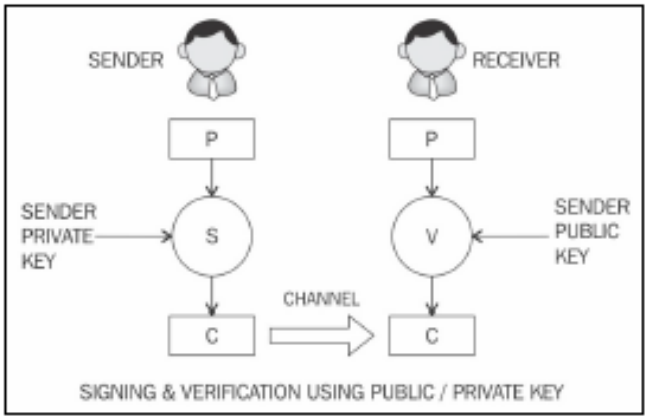
UNIT-III

5.a) Explain about Asymmetric Cryptography in Block chain? (7 M)

Sol:

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. This is also known as public key cryptography. It uses both public and private keys to encrypt and decrypt data, respectively. Various asymmetric cryptography schemes are in use, including RSA, DSA, and ElGammal.

An overview of public key cryptography is shown in the following diagram:



The preceding diagram shows that sender digitally signs the plaintext P with his private key using signing function S and produces data C which is sent to the receiver who verifies C using sender public key and function V to ensure the message has indeed come from the sender. Security mechanisms offered by public key cryptosystems include key establishment, digital signatures, identification, encryption, and decryption. Key establishment mechanisms are concerned with the design of protocols that allow the setting up of keys over an insecure channel. Non-repudiation services, a very desirable property in many scenarios, can be provided using digital signatures. Sometimes, it is important not only to authenticate a user but also to identify the entity involved in a transaction. This can also be achieved by a combination of digital signatures and challenge response protocols. Finally, the encryption mechanism to provide confidentiality can also be obtained using public key cryptosystems, such as RSA, ECC, and ElGammal.

Public key algorithms are slower in terms of computation than symmetric key algorithms. Therefore, they are not commonly used in the encryption of large files or the actual data that requires encryption. They are usually used to exchange keys for symmetric algorithm. Once the keys are established securely, symmetric key algorithms can be used to encrypt the data.

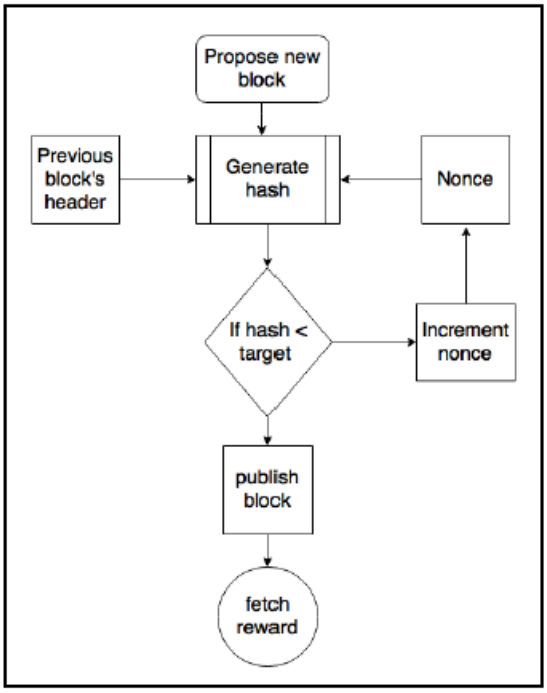
5.b) Explain the working functionality of mining algorithm in Bitcoin. (7 M)

Sol:

The mining algorithm consists of the following steps.

1. The previous block's header is retrieved from the bitcoin network.
2. Assemble a set of transactions broadcasted on the network into a block to be proposed.
3. Compute the double hash of the previous block's header combined with a nonce and the newly proposed block using the SHA-256 algorithm.
4. Check if the resultant hash is lower than the current difficulty level (target) then PoW is solved. As a result of successful PoW the discovered block is broadcasted to the network and miners fetch the reward.
5. If the resultant hash is not less than the current difficulty level (target), then repeat the process after incrementing the nonce.

As the hash rate of the bitcoin network increased, the total amount of 32-bit nonce was exhausted too quickly. In order to address this issue, the extra nonce solution was implemented, whereby the coinbase transaction is used as a source of extra nonce to provide a larger range of nonce to be searched by the miners. This process can be visualized by using the following flowchart:



6.a) Explain any two block ciphers with the example scenarios. (7 M)

Sol:

****Explanation of any two symmetric or public key cryptographic mechanisms such as below can be awarded marks.****

- RSA
- SHA
- Elliptic Curve Cryptography (ECC**)**

6.b) What are the hash rate mining systems? Explain any two mining systems.(7 M)

Sol:

The hash rate

The hashing rate basically represents the rate of calculating hashes per second. In other words, this is the speed at which miners in the Bitcoin network are calculating hashes to find a block. In early days of bitcoin, it used to be quite small as CPUs were used. However, with dedicated mining pools and ASICs now, this has gone up exponentially in the last few years. This has resulted in increased difficulty of the Bitcoin network. The following hash rate graph shows the hash rate increase over time and is currently measured in Exa hashes. This means that in 1 second, the Bitcoin network miners are computing more than 24,000,000,000,000,000,000 hashes per second.

Mining systems

Over time, bitcoin miners have used various methods to mine bitcoins. As the core principle behind mining is based on the double SHA-256 algorithm, overtime experts have developed

sophisticated systems to calculate the hash faster and faster. The following is a review of the different types of mining methods used in bitcoin and how they evolved with time.

CPU

CPU mining was the first type of mining available in the original bitcoin client. Users could even use laptop or desktop computers to mine bitcoins. CPU mining is no longer profitable and now more advanced mining methods such as ASIC-based mining is used. CPU mining only lasted for around just over a year since the introduction of Bitcoin and soon other methods were explored and tried by the miners.

GPU

Due to the increased difficulty of the bitcoin network and the general tendency of finding faster methods to mine, miners started to use GPUs or graphics cards available in PCs to perform mining. GPUs support faster and parallelized calculations that are usually programmed using the OpenCL language. This turned out to be a faster option as compared to CPUs. Users also used techniques such as overclocking to gain maximum benefit of the GPU power. Also, the possibility of using multiple graphics cards increased the popularity of graphics cards' usage for bitcoin mining. GPU mining, however, has some limitations, such as overheating and the requirement for specialized motherboards and extra hardware to house multiple graphics cards. From another angle, graphics cards have become quite expensive due to increased demand and this has impacted gamers and graphic software users.

FPGA

Even GPU mining did not last long, and soon miners found another way to perform mining using FPGAs. Field Programmable Gate Array (FPGA) is basically an integrated circuit that can be programmed to perform specific operations. FPGAs are usually programmed in Hardware Description Languages (HDLs), such as Verilog and VHDL. Double SHA-256 quickly became an attractive programming task for FPGA programmers and several open source projects started too. FPGA offered much better performance as compared to GPUs; however, issues such as accessibility, programming difficulty, and the requirement for specialized knowledge to program and configure FPGAs resulted in a short life of the FPGA era for bitcoin mining.

ASICs

Application Specific Integrated Circuit (ASIC) was designed to perform the SHA-256 operation. These special chips were sold by various manufacturers and offered a very high hashing rate. This worked for some time, but due to the quickly increasing mining difficulty level, single-unit ASICs are no longer profitable.

UNIT-IV

7.a) Explain the life cycle of a smart contract. (7 M)

Sol:

A smart contract is a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable.

Dissecting this definition further reveals that a smart contract is, in fact, a computer program that is written in a language that a computer or target machine can understand. Also, it encompasses agreements between parties in the form of business logic. Another fundamental idea is that smart contracts are automatically executed when certain conditions are met. They are enforceable, which means that all contractual terms are executed as defined and expected, even in the presence of adversaries. Enforcement is a broader term that encompasses traditional enforcement in the form of law, along with the implementation of specific measures and controls that make it possible to execute contract terms without requiring any mediation. It should be noted that true smart contracts should not rely on traditional methods of enforcement. Instead, they should work on the principle that code is law, meaning that there is no need for an arbitrator or a third party to control or influence the execution of the smart contract. Smart contracts are self enforcing as opposed to legally enforceable. This idea might be regarded as a libertarian's dream, but it is entirely possible and is in line with the true spirit of smart contracts. Moreover, they are secure and unstoppable, which means that these computer programs are required to be designed in such a fashion that they are fault-tolerant and executable in a reasonable amount of time. These programs should be able to execute and maintain a healthy internal state, even if external factors are unfavorable. For example, imagine a typical computer program that is encoded with some logic and executes according to the instruction coded within it. However, if the environment it is running in or external factors it relies on deviate from the normal or expected state, the program may react arbitrarily or simply abort. It is essential that smart contracts be immune to this type of issue.

Secure and unstoppable may well be considered requirements or desirable features but it will provide more significant benefits in the long run if security and unstoppable properties are included in the smart contract definition from the beginning. This will allow researchers to focus on these aspects from the start and will help to build strong foundations on which further research can then be based. There is also a suggestion by some researchers that smart contracts need not be automatically executable; instead, they can be what's called automatable, due to manual human input required in some scenarios. For example, a manual verification of a medical record might be required by a qualified medical professional. In such cases fully automated approaches may not work best. While it is true that in some cases human input and control is desirable, it is not necessary; and, for a contract to be truly smart, in the author's opinion, it has to be fully automated. Some inputs that need to be provided by people can and should also be automated via the use of Oracles. Smart contracts usually operate by managing their internal state using a state machine
model. This allows development of an effective framework for programming smart contracts, where the state of a contract is advanced further based on some predefined criteria and conditions.
There is also on-going debate on the question of whether the code is acceptable as a contract in a court of law. A smart contract is different in presentation from traditional legal prose, albeit they do represent and enforce all contractual clauses but a court of law does not understand the code. This dilemma raises several questions about how a smart contract can be legally binding: can it be developed in such a way that it is readily acceptable and understandable in a court of law? How can dispute resolution be implemented within the code, and is it possible? Moreover,

regulatory and compliance requirements is another topic that needs to be addressed before smart contracts can be used as efficiently as traditional legal documents. Even though smart contracts are named smart, they in fact only do what they have been programmed to do, and that is fine because this very property of smart contracts ensures that smart contracts produce same output every time they are executed. This deterministic nature of smart contracts is highly desirable in blockchain platforms due to consistent consensus requirements. This means that smart contracts are not really smart, they are simply doing what they are programmed to do.

Smart contracts are inherently required to be deterministic. This property will allow a smart contract to be run by any node on a network and achieve the same result. If the result differs even slightly between nodes, then consensus cannot be achieved, and a whole paradigm of distributed consensus on blockchain can fail. Moreover, it is also desirable that the contract language itself is deterministic, thus ensuring integrity and stability of the smart contracts. Deterministic in the sense that there are no non-deterministic functions used in the language, which can produce different results on various nodes.

In summary, a smart contract has the following four properties:

- Automatically executable
- Enforceable
- Semantically sound
- Secure and unstoppable

The first two properties are required as a minimum, whereas the latter two may not be required or implementable in some scenarios and can be relaxed. For example, a financial derivatives contract does not perhaps need to be semantically sound and unstoppable but should at least be automatically executable and enforceable at a fundamental level. On the other hand, a title deed needs to be semantically sound and complete, therefore, for it to be implemented as a smart contract, the language must be understood by both computers and people.

7.b) What is Ethereum network? Explain the components of the Ethereum ecosystem.

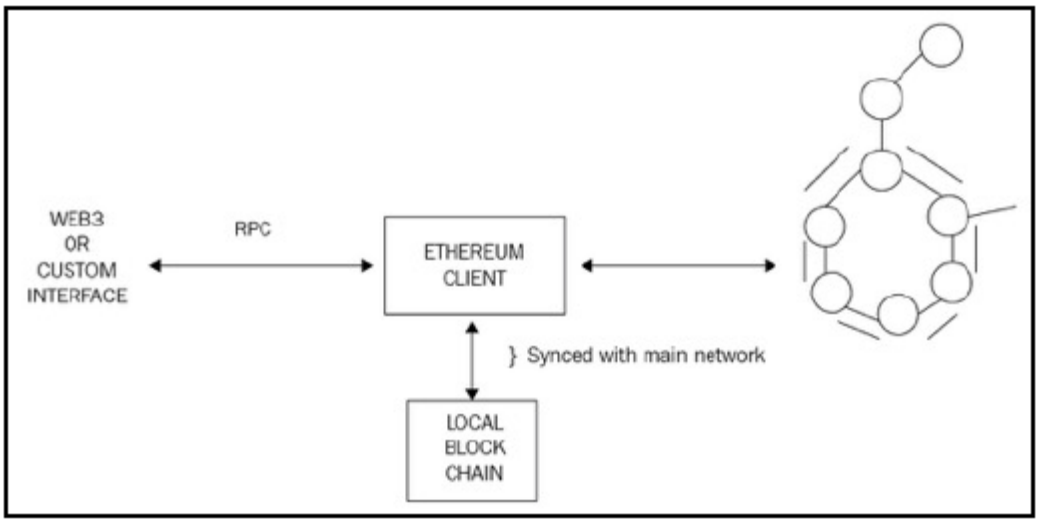
(7 M)

Sol:

The Ethereum network is a peer-to-peer network where nodes participate in order to maintain the blockchain and contribute to the consensus mechanism. Networks can be divided into three types, based on requirements and usage.

The Ethereum blockchain stack consists of various components. At the core, there is the Ethereum blockchain running on the peer-to-peer Ethereum network. Secondly, there's an Ethereum client (usually Geth) that runs on the nodes and connects to the peer-to-peer Ethereum network from where blockchain is downloaded and stored locally. It provides various functions, such as mining and account management. The local copy of the blockchain is synchronized regularly with the network.

This architecture can be visualized in the following diagram:



The Ethereum stack showing various components

A formal list of all high-level elements present in the Ethereum blockchain is presented here:

Keys and addresses

Keys and addresses are used in Ethereum blockchain mainly to represent ownership and transfer of Ether. Keys are used in pairs of private and public type. The private key is generated randomly and is kept secret whereas a public key is derived from the private key. Addresses are derived from the public keys which are a 20-bytes code used to identify accounts.

Accounts

Accounts are one of the main building blocks of the Ethereum blockchain. Ethereum, being a transaction driven state machine, the state is created or updated as a result of the interaction between accounts and transaction execution. Operations performed between and on the accounts, represent state transitions.

Transactions and messages

A transaction in Ethereum is a digitally signed data packet using a private key that contains the instructions that, when completed, either result in a message call or contract creation.

Messages

Messages, as defined in the yellow paper, are the data and value that are passed between two accounts. A message is a data packet passed between two accounts. This data packet contains data and value (amount of ether). It can either be sent via a smart contract (autonomous object) or from an external actor (externally owned account) in the form of a transaction that has been digitally signed by the sender.

Ether cryptocurrency / tokens (ETC and ETH)

As an incentive to the miners, Ethereum also rewards its own native currency called Ether, abbreviated as ETH. After the DAO hack (described later in this chapter), a hard fork was

proposed in order to mitigate the issue; therefore, there are now two Ethereum blockchains: one is called Ethereum Classic, and its currency is represented by ETC, whereas the hardforked version is ETH, which continues to grow and on which active development is being carried out. ETC, however, has its following with a dedicated community that is further developing ETC, which is the unforked original version of Ethereum.

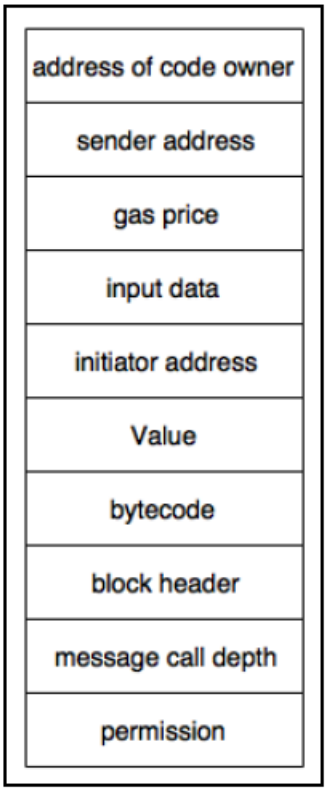
The Ethereum Virtual Machine (EVM)
EVM is a simple stack-based execution machine that runs bytecode instructions to transform the system state from one state to another.

Smart Contract
A smart contract is a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable.

8.a) Explain about execution environment in Ethereum virtual machine. (7 M)
Sol:
There are some key elements that are required by the execution environment to execute the code. The key parameters are provided by the execution agent, for example, a transaction.
These are listed as follows:

- System state.
- Remaining gas for execution.
- The address of the account that owns the executing code.
- The address of the sender of the transaction.
- The originating address of this execution (it can be different from the sender).
- The gas price of the transaction that initiated the execution.
- Input data or transaction data depending on the type of executing agent. This is a byte array; in the case of a message call, if the execution agent is a transaction, then the transaction data is included as input data.
- The address of the account that initiated the code execution or transaction sender.
- This is the address of the sender in case the code execution is initiated by a transaction; otherwise, it is the address of the account.
- The value or transaction value. This is the amount in Wei. If the execution agent is a transaction, then it is the transaction value.
- The code to be executed presented as a byte array that the iterator function picks up in each execution cycle.
- The block header of the current block.
- The number of message calls or contract creation transactions currently in execution. In other words, this is the number of CALLs or CREATEs currently in execution.
- Permission to make modifications to the state.

The execution environment can be visualized as a tuple of ten elements, as follows:



Execution environment tuple

The execution results in producing the resulting state, the gas remaining after the execution, self-destruct or suicide set (described later), log series (described later), and any gas refunds.

8.b) Explain the operations of a DApp in Ethereum. (7 M)

Sol:

There are various implementations of DAOs and smart contracts in Ethereum, most notably, the DAO, which was recently misused due to a weakness in the code and required a hard fork for funds to be recovered that have been syphoned out by the attackers. The DAO was created to serve as a decentralized platform to collect and distribute investments. Augur is another DApp that has been implemented on Ethereum, which is a decentralized prediction market.

Development tools, IDEs, and clients

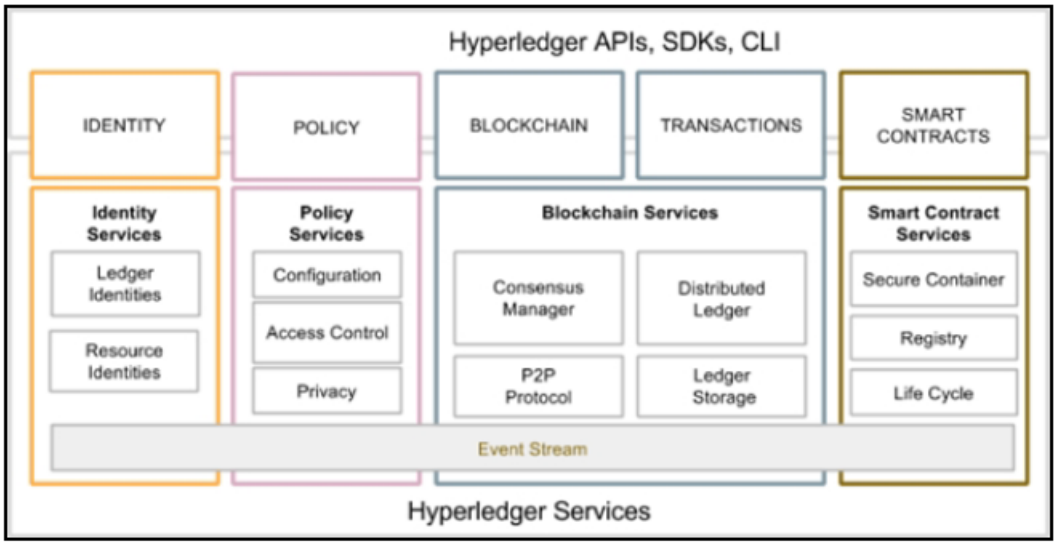
- Remix
- Ganache
- EthereumJS
- TestRPC
- MetaMask
- Truffle

UNIT-V

9.a) Explain the fundamental components of the Hyperledger reference architecture. **(7 M)**

Sol:

The reference architecture consists of various components that form a business blockchain. These high-level components are shown in the reference architecture diagram shown here:



Starting from the left we see that we have five top-level components which provide various services.

First is identity, that provides authorization, identification, and authentication services under membership services. Then is the policy component, which provides policy services. After this, ledger and transactions come, which consists of the distributed ledger, ordering service, network protocols, and endorsement and validation services. This ledger is updateable only via consensus among the participants of the blockchain network. Finally, we have the smart contracts layer, which provides chaincode services in Hyperledger and makes use of secure container technology to host smart contracts. We will see all these in more detail in the Hyperledger Fabric section shortly. Generally, from a components point of view Hyperledger contains various elements described here:

- Consensus layer: These services are responsible for facilitating the agreement process between the participants on the blockchain network. The consensus is required to make sure that the order and state of transactions is validated and agreed upon in the blockchain network.
- Smart contract layer: These services are responsible for implementing business logic as per the requirements of the users. Transaction are processed based on the logic defined in the smart contracts that reside on the blockchain.
- Communication layer: This layer is responsible for message transmission and exchange between the nodes on the blockchain network.
- Security and crypto layer: These services are responsible for providing a capability to allow various cryptographic algorithms or modules to provide privacy, confidentiality and non-repudiations services.
- Data stores: This layer provides an ability to use different data stores for storing state of the ledger. This means that data stores are also pluggable and allows usage of any database backend.

- Policy services: This set of services provide the ability to manage different policies required for the blockchain network. This includes endorsement policy and consensus policy.
- APIs and SDKs: This layer allows clients and applications to interact with the blockchain. An SDK is used to provide mechanisms to deploy and execute chaincode, query blocks and monitor events on the blockchain.

There are certain requirements of a blockchain service. In the next section, we are going to discuss the design goals of Hyperledger Fabric.

9.b) Describe about Hyperledger Frabric in Detail. (7 M)

Sol:

The fabric is the contribution made initially by IBM and Digital Assets to the Hyperledger project. This contribution aims to enable a modular, open, and flexible approach towards building blockchain networks.

Various functions in the fabric are pluggable, and it also allows the use of any language to develop smart contracts. This functionality is possible because it is based on container technology (Docker), which can host any language.

Fabric can be defined as a collection of components providing a foundation layer that can be used to deliver a blockchain network. There are various types and capabilities of a fabric network, but all fabrics share common attributes such as immutability and are consensus-driven. Some fabrics can provide a modular approach towards building blockchain networks. In this case, the blockchain network can have multiple pluggable modules to perform a various function on the network.

For example, consensus algorithms can be a pluggable module in a blockchain network where, depending on the requirements of the network, an appropriate consensus algorithm can be chosen and plugged into the network. The modules can be based on some particular specification of the fabric and can include APIs, access control, and various other components.

Fabrics can also be designed either to be private or public and can allow the creation of multiple business networks. As an example, Bitcoin is an application that runs on top of its fabric (blockchain network).

10.a) What is Hyperledger, and what distinguishes it from other block chain platforms? Explain its requirements & Design goals. (7 M)

Sol:

Hyperledger is not a blockchain, but it is a project that was initiated by the Linux Foundation in December 2015 to advance blockchain technology. This project is a collaborative effort by its members to build an open source distributed ledger framework that can be used to develop and implement cross-industry blockchain applications and systems. The principal focus is to develop and run platforms that support global business transactions. The project also focuses on improving the reliability and performance of blockchain systems.

Requirements and design goals of Hyperledger

Fabric

There are certain requirements of a blockchain service. The reference architecture is driven by the needs and requirements raised by the participants of the Hyperledger project and after studying the industry use cases.

The modular approach

The main requirement of Hyperledger is a modular structure. It is expected that as a cross industry fabric (blockchain), it will be used in many business scenarios. As such, functions related to storage, policy, chaincode, access control, consensus, and many other blockchain services should be modular and pluggable. The specification suggests that the modules should be plug and play and users should be able to easily remove and add a different module that meets the requirements of the business.

Privacy and confidentiality

This requirement is one of the most critical factors. As traditional blockchains are permissionless, in the permissioned model like Hyperledger Fabric, it is of utmost importance that transactions on the network are visible to only those who are allowed to view it.

Scalability

This is another major requirement which once met will allow reasonable transaction throughput, which will be sufficient for all business requirements and also a large number of users.

Deterministic transactions

This is a core requirement in any blockchain because if transactions do not produce the same result every time they are executed regardless of who and where the transaction is executed, then achieving consensus is impossible. Therefore, deterministic transactions become a key requirement in any blockchain network.

Identity

In order to provide privacy and confidentiality services, a flexible PKI model that can be used to handle the access control functionality is also required. The strength and type of cryptographic mechanisms is also expected to vary according to the needs and requirements of the users. In certain scenarios, it might be required for a user to hide their identity, and as such, the Hyperledger is expected to provide this functionality.

Auditability

Auditability is another requirement of Hyperledger Fabric. It is expected that an immutable audit trail of all identities, related operations, and any changes is kept.

Interoperability

Currently, there are many blockchain platforms available, but they cannot communicate with each other and this can be a limiting factor in the growth of a blockchain-based global business

ecosystem. It is envisaged that many blockchain networks will operate in the business world for specific needs, but it is important that they are able to communicate with each other. There should be a common set of standards that all blockchains can follow in order to allow communication between different ledgers. It is expected that a protocol will be developed that will allow the exchange of information between many fabrics.

Portability

The portability requirement is concerned with the ability to run across multiple platforms and environments without the need to change anything at code level. Hyperledger Fabric is envisaged to be portable, not only at infrastructure level but also at code, libraries, and API levels, so that it can support uniform development across various implementations of Hyperledger.

Rich data queries

The blockchain network should allow rich queries to be run on the network. This can be used to query the current state of the ledger using traditional query languages, which will allow for wider adoption and ease of use.

10.b) How does block chain Quorum address the specific needs of industries beyond crypto- currency, such as finance or supply chain management? (7 M)

Sol:

Finance

Blockchain has many applications in the finance industry. Blockchain in finance is the hottest topic in the industry currently, and major banks and financial organizations are researching to find ways to adapt blockchain technology primarily due to its highly-desired potential to cost-save.

Supply Chain Management

There are many potential use cases for blockchain technology in supply chain management. Here are some of the most promising applications. Traceability sits at the root of most of the use cases and benefits that derive from that ability.

Traceability and Transparency

One of the biggest challenges for supply chain management executives is maintaining visibility across the network. Blockchain technology can help address this challenge by providing a secure and transparent way to track goods as they move through the supply chain. This can help reduce the problems already mentioned while improving efficiency, security and the shopping experience for businesses, wholesalers, retailers and ultimately consumers.

Environmental and Ethical Sustainability

As sustainability and ESG aspects become more important in our world, blockchain technology will be used to promote environmental sustainability by tracking carbon emissions and other environmental impacts throughout the supply chain. This information can then be used to

identify areas for improvement and reduce overall environmental impact. In addition, blockchain technology can help companies ensure that their products are ethically sourced. By tracking products from the point of origin, businesses can identify and flag any potential ethical issues such as child or slave labour, fair wages or safe working environments in the supply chain.

Quality Assurance

Blockchain technology can be used to ensure that products meet certain quality standards throughout the supply chain. By recording data on the blockchain at each stage of production, companies can track and verify compliance with specific requirements.

Counterfeit Prevention

Brand and product piracy through counterfeit products is a significant problem in many industries, especially luxury goods and pharmaceuticals. Blockchain technology can help prevent counterfeiting by creating a tamper-proof record of product ownership and authenticity.

Streamlining Payment Processing

Blockchain technology can also be used to streamline payment processing in supply chains. By using smart contracts, payments can be automated based on predefined conditions such as delivery confirmation or quality inspection.

These are just a few examples of how blockchain technology can be used in supply chain management.