Advanced Financial Instruments for Sustainable Business and Decentralized Markets

Prof. Abhinava Tripathi

Department of Management Sciences

Indian Institute of Technology, Kanpur

Week 10

Lesson 29

In this lesson, we start the discussion with the functions of blockchains. We introduce the concept of cryptographic public and private keys, peer-to-peer network and digital ledger. We also introduce the proof of work and proof of stake mechanisms. We introduce three major concepts including hash chain storage, digital signature and consensus mechanism. We also discuss hash pointers and Merkle tree. We also discuss the public infrastructure concept.

Next we introduce the concept of blockchain transactions. We also discuss some fundamental aspects of blockchain. These include hash encryptions, authentication and authorization, mining, proof of work and proof of stake concepts. Next we discuss cryptographic primitives.

These include digital signature comprising public and private keys, role of consensus mechanism and hash functions. Subsequently we discuss hash functions and their key properties including hiding or pre-image resistance, collision resistance and puzzle friendliness. Lastly we also discuss some applications of hash functions including address derivation, block identification and securing the block data through chaining the blocks. In this video we will briefly introduce the functioning of blockchain.

How Does the Blockchain Work

- Blockchain is a combination of three important technologies - cryptographic keys (public key and private key), a peer-to-peer network, and a digital ledger.
- A deal or transaction is authorized by a mathematical verification (Consensus mechanism) in a peer-to-peer network.
- Proof of stake achieves consensus by requiring participants to stake crypto behind the new block they want added to a cryptocurrency's blockchain. Meanwhile, proof of work achieves consensus by requiring participants (miners) to spend computational power — and electricity — in order to generate a new valid block. Proof of work has the advantage of making it very expensive to attack a cryptocurrency's network, yet it comes at a growing environmental cost
- All of these transactions are stored in a structure known as the digital ledger. The information contained in the digital ledger is highly secure, and the digital signature safeguards it from being tampered with.

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

To begin with blockchain is a combination of three important technologies namely cryptographic keys which includes a public and private key, a peer-to-peer network and a digital ledger.

Now these cryptographic keys are of two types as I said private and public. Each individual or node on the blockchain has both of these keys and they are used to create digital signature. This digital signature is a unique and secure digital identity reference and the most important aspect of blockchain technology which makes up for your smart valet also. So you have smart valet based on this public and private key technology and every transaction is authorized by the digital signature of the owner as in who is initiating the transaction. Now here a deal or transaction is authorized by a mathematical function often called a consensus mechanism.

This mathematical calculation is taking place in a peer-to-peer network. This peer-to-peer network is a large group of individuals who act as authorities to reach a consensus on transactions among other things. Some of these consensus mechanisms are like proof of work, other is proof of stake and these different mechanisms are used by cryptocurrencies for example for achieving consensus on which new blocks to add to their blockchains. Now these nodes they solve the basic problem of verifying the transactions without using a central authority and that is why it is decentralized. Let's take example of proof of stake mechanism.

So proof of stake achieves consensus by requiring participants to stake or put at stake their cryptocurrencies behind the new block they want to get added to a cryptocurrency

blockchain. That is about proof of stake. Conversely there is proof of work. So proof of work achieves consensus by requiring participants or miners to spend computational power, electricity in order to generate a new valid block and proof of work has the advantage the way we discussed of making it very expensive to attach a cryptocurrency network yet it comes at a growing environmental cost. While proof of stake avoids the massive energy consumption that is associated with proof of work.

However proof of stake has not been proven to be as secure and as stable as proof of work at large scale. Moreover a final feature of blockchain technology is the structure of the database itself. So all of these transactions are stored in a structure known as digital ledger. The information contained in the digital ledger is highly secure and the digital signature safeguards is from being tampered with. This is where blockchain gets its name block and chain.

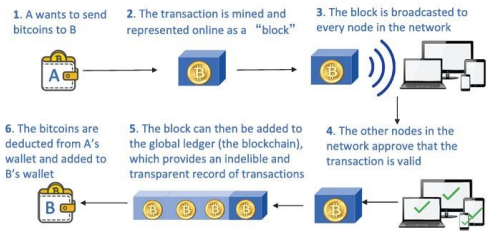
Here data is added to the database in a series of blocks, series of blocks get added as transactions happen a series of blocks get added to this chain and therefore the name blockchain. These blocks are linked together and form an order chain. This structure allows for each entry or transaction in the database to be ordered. Let's take an example. Let's say you are a node on the network and you have a file of transactions on your computer which we are calling as ledger.

Now there are other nodes who have the same file on their network that is why it is distributed. So they have the same file on the same information on their system. Now let's say you initiate a transaction maybe giving a block cryptocurrency or a bitcoin to somebody else. As you make the transaction your system sends a signal to the all the nodes your system sells signal to all the nodes node 1, node 2, node 3 and so on. Now these nodes rush to verify this transaction that you have created through some proof of work or proof of stake mechanism so that they can get paid their salary or bitcoins.

Now the first one to check and validate communicates their logic for verifying this transaction maybe proof of work this is called proof of work. If other nodes agree with them then everybody objects their files. So they may use a consensus mechanism like proof of work or proof of stake to verify the transaction. Once the transaction is verified another block is added to the chain and this information new information is appended.

How Does the Blockchain Work

- In a Bitcoin network, if client A wants to send some bitcoins to another client B, it will create a bitcoin transaction by client A
- Three basic and important capabilities that are supported by the blockchain implementation in Bitcoin are: (1) the hash chained storage, (2) digital signature, and (3) the commitment consensus for adding a new block to the globally chained storage



Source: Security and Privacy on Blockchain
https://dl.acm.org/doi/10.1145/3316481

So to summarize this transaction a blockchain functionally serves as a distributed and secure database or transaction box and in a bitcoin network let's say client A wants to send some money or bitcoin to B.

Now the blockchain will initiate this transaction on behalf of client A. So client A will initiate the transaction. The transaction has to be approved by the miners so it is so this is this transaction is mined by the miners before it gets added in the form of a new block of information about the transaction on the network. So it needs to be added before and verified by the miners. Once it is verified it is added and once it is added to the blockchain this block is broadcasted communicated to every node in the network.

Those nodes that are miners will collect the transaction in the form of a block verify the transaction and broadcast this block. This verification process may require some consensus protocol like proof of stake or proof of work. Once they broadcast other miners will verify their proof of work. They will verify this transaction and when others node verify the transaction and prove that or sort of agree that the block is valid the block can be added to the blockchain. Only when the block containing the transaction approved by other nodes and added to the blockchain this bitcoin transfers from A to B.

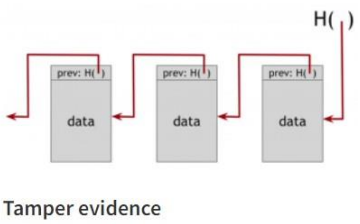
So the transfer takes place and finally the bitcoin transfer from A to B becomes finalized and becomes or considered as legitimate. Now three and very important capabilities that are supported by such blockchain implementation for example on bitcoin are the hash chain storage, digital signature and the commitment consensus for adding a new block to the global chain storage. This is an elegant combination of a suite of popularly secure technique

such as hash chain muckle t and digital signature with consensus mechanisms. The bitcoin blockchain can prevent both the double spending problem of bitcoins as we discussed earlier and stop the retrospective modification of three transaction data in a block after the block has been successfully added or committed to the blockchain. So once the block has been added to the blockchain on the global ledger these properties or sort of components of blockchain that is hash chain storage digital signature and commitment consensus ensures that it is immutable a malicious party cannot modify any block or the information contained therein.

So to summarize in this video we introduced the blockchain function we showed how a transaction takes place and why it is immutable and secure and cannot be modified by an attacker or malicious party.

Hash Chained Storage

- Hash pointer and Merkle tree are the two fundamental building blocks for implementing the blockchain in Bitcoin using the hash chained storage
- Hash pointer is a hash of the data by cryptography, pointing to the location in which the data is stored.



Tamper evidence

- If an adversary attempts to change data in any block in the entire chain, in order to disguise the tampering, the adversary has to change the hash pointers of all previous blocks

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

In this video we will discuss hash pointer, muckle tree and digital signature which are the key components of a blockchain. First let us start with hash chain storage. Hash pointer and muckle tree are two fundamental building blocks for implementing the blockchain on a blockchain network such as bitcoin using hash chain storage. Hash pointer is a hash of the data created by cryptography pointing to the location in which data is stored.

So on a blockchain a hash pointer would point towards the previous subsequently previous block it is created from the hash of the information available in the previous block so you apply hash function to create hash from a given block and this is stored as a hash pointer on the subsequent blocks. So if the flow is in this direction you have the header of a block containing information in the form of hash of the previous block it also contains a pointer which points to the block in which from which the information is taken. Thus a hash pointer

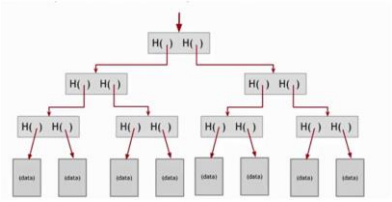
can be used to check whether or not the data has been tampered with. In a blockchain which is organized using hash pointers to link data blocks together like this with the hash function or hash pointer pointing to the predecessor block so each block is pointing to the predecessor block and indicates the address where the data of the predecessor block is stored. Moreover the hash of the stored data can be publicly verified by users to prove that the stored data has not been tampered with.

Let's take an example. If an adversary attempts to change the data in any block in the entire chain in order to disguise the tampering the adversary has to change the hash pointers of all the following blocks. So for example if I am standing here with my information and this block contains at its header all the information about the previous block which includes the hash of the block itself and anybody tampers let's say this block then this the hash of this and information and generated resulting generated hash pointer will not match with this because hash function has a collision resistance property as we will discuss shortly but the point being here is that hash generated from this block will not match with its hash pointer here. Similarly if somebody tries to tamper with this block then this hash would not match with this and subsequently it will be tampered with. Now suppose they not only tamper with this but tamper with this also then again the information and the subsequent hash generated from this will not match with this hash pointer. In this fashion they have to change all the way all the subsequent for any block we have to change all the subsequent blocks which is next to impossible and that's why it's sort of tamper evident this blockchain is tamper evident.

Ultimately that adversary or malicious party has to stop tampering because they will not be able to falsify data on the entire chain and the chain starts from the genesis block or the head of the chain which was initially generated once the system was built we can call this initial opening block in the chain as genesis block. Ultimately the adversaries tampering with the block will be uncovered because by recording this single root hash pointer of the genesis block one can effectively make the entire chain have the property of tamper resilience as well. Users are allowed to go back to some special block and verify it from the beginning of the chain. So you can go at any block and starting from there up till beginning of the chain you can verify the chain of blocks. Again here another important component is Merkle tree.

Merkle Tree

- A Merkle tree is defined as a binary search tree with its tree nodes linked to one another using hash pointers
- A Merkle tree has the ability of preventing data from tampering by traversing down through the hash pointers to any node in the tree



- An advantage of Merkle tree is that it can prove effectively and concisely the membership of a data node by showing this data node and all of its ancestor nodes on its upward pathway to the root node

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Merkle tree appears like this and it is defined as a binary search tree with its tree nodes linked to one another using hash pointers. It is another useful data structure used for building a blockchain. Here we group these nodes into disjointed groups such that each time two nodes at the lower level are grouped into one at the parent level and for each pair of lower nodes the Merkle tree construction algorithm is creating a new data node. So if the data was stored here two data blocks are connected with this node and then another two nodes with this and two nodes are connected to create another parent node and so on. Ultimately there will be some root node or the starting node.

So in this fashion grouping these nodes into disjointed groups such that each time two nodes at the lower level are grouped into one at the parent level and for each pair of lower level nodes the Merkle tree construction algorithm is creating a new data node which contains the hash value of each. So it contains the hash value of this node also and this node also kind of hash pointer. So this process is repeated until we reach the root and a Merkle tree has the ability of preventing data from tampering by traversing down through the hash pointers in any node in the tree. So you have each node pointing to the information of the previous subsequent nodes here pointing to these nodes through hash pointers. Specifically when an adversary tries to tamper with the data maybe here, here, here or here at any of these leaf nodes or here it will cause a change in the hash value of its parent node.

So if you change data here the parent its hash value will not meet with the hash value at the parent or if you change this node's information then hash pointer here which is pointing to this node should be changed. It will cause a change in the hash value of the parent node

even if it continues to tamper with the upper node. So even if you keep on tampering the upper and upper nodes above and above the malicious party needs to change all the nodes on the path so they need to change all the nodes on the path up to the top and therefore one can easily detect that data has been tampered with since the hash pointer of this root node, the final root node does not match with the hash pointer that has been stored. So whatever that the party here has I have the hash which was created at the beginning it will not match if any of these blocks are tampered the hash pointer of this root will change and it will not match with the hash with me. So the advantage of Merkle Tree is that it can prove effectively and concisely the membership of data node at any point here.

By showing that this data node and all of its ancestor nodes on its upward pathway to the root node the membership of Merkle Tree can be easily verified in this kind of fashion by examining the hashes on the path and checking whether the hash value is matching with the root. So if it does match the root based on this tampering ultimately the hash pointer at this root node will change and if it does not match with the hash with the originator you know that there is some tampering.

Digital Signature

- A digital signature establishes the validity of a piece of data by using a cryptographic algorithm
- There are three core components that formulate a digital signature scheme
  - Key generation algorithm: Public Key and Private Key
  - Signing algorithm
  - Verification algorithm

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

We have digital signature which is also a very important component. A digital signature establishes the validity of a piece of data by using a cryptographic algorithm. It is also a scheme for verifying that a piece of data has not been tampered with.

There are three core components that formulate a digital signature scheme. First component is the Key Generation algorithm which creates two keys. One is used to sign the messages and be kept privately and called the private key. The other is made available to the public as the name suggests and thus called the public key. It is used to validate whether the

message has the signature signed with the corresponding private key.

The second core component is the signing algorithm. It produces a signature on the input message endorsed by using the given private key. The third very important component is the verification algorithm. It takes a signature, a message and a public key as inputs and validates the message signature using the public key and returns a Boolean kind of value. A well defined and secure signature algorithm should have two properties.

The first property is that valid signatures must be verifiable. The second property is that signatures are existentially unforgeable. They can't be changed. It means that an adversary who has your public key cannot forge signatures on some messages with an overwhelming probability. So, to summarize this video, we discussed Hash chain storage which included hash pointer and mercury.

We also discussed digital signature. These three components form a very important and intriguing component of blockchain. Its security property, they contribute to its security property, its security so that data cannot be altered on a blockchain. They are a very integral part of the cryptography associated with the blockchain.

Public Key Infrastructure (PKI)

- The advantage of using a digital signature is to effectively validate the authenticity of a message by utilizing public key infrastructure (PKI)
- One can obtain the key pair from a trusted party
- The process of signature verification is automatically translated into identity verification of the signer based on the assurance level of the binding

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

In this video, we will discuss two very important components of blockchain that is public infrastructure and consensus mechanism. Let us start the discussion with public infrastructure or public keys that act as pseudonyms.

The advantage of using additional signature is to effectively validate the authenticity of a message by utilizing public infrastructure, such that the writer of the message signs it with

her private key before sending it out. The recipient of this signed message can use the sender's public key to prove the validity of the message. Moreover, here one can obtain the key pair from a trusted third party in most application scenarios. Public infrastructure is used to manage the public keys by establishing a binding agreement between respective identities of entities, for example, name, email and ID, etc. and their public keys. Such binding is done by registering and issuing certificates with the certificate authority. The process of signature verification is automatically translated into identity verification of the signer based on the assurance level of the binding. The public key can be seen as an identity in these scenarios. While blockchains, Bitcoin adopts decentralizing identity management without having a central authority to register a user in a system, key pairs are generated by users themselves. So the members or nodes, they generate the key pair of public private key themselves in a sort of decentralized permissionless blockchain like Bitcoin.

And users can generate these key pairs as many as they want these identities or what we call as hashes. Hashes of public keys are called addresses in Bitcoin, because there is no central management of public keys. These identities are actually pseudonyms made by the users themselves. Next we come to the consensus mechanism. So in the context of decentralized blockchains, when a new block is sent by broadcasting to the network, each node has the option to add that block to their copy of the global ledger or ignore it.

Consensus

- When a new block is sent by broadcasting to the network, each node has the option to add that block to their copy of the global ledger or to ignore it
- The consensus is employed to seek for the majority of the network to agree upon a single state update in order to secure the expansion of the global ledger
- Adversarial offense could happen when a node decides to tamper with the state of his copy of the global ledger
- In order to enable the blockchain to function on a global scale with security and correctness guarantee, the shared public ledger needs an efficient and secure consensus algorithm

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Now here the consensus mechanism is employed to seek from the majority of the network to agree upon a single state update in order to secure the expansion of the global ledger or the blockchain. So the block will be added when majority that means 51% of them will

agree about verifying the transaction and prevent dishonest attempts or malicious attacks. Concretely, given that the blockchain is huge, shared by global ledger, anyone may update it. Now adversarial offenses could happen when a node decides to tamper with the state of his copy of the global ledger or when several bodies collusively attempt on such tampering. For example, if Alice was sending 10 bitcoins to Bob from her wallet, she would like to be sure that no one in the network can tamper with the transaction content and change 100 or 10 bitcoins to 100 bitcoins.

Now in order to enable the blockchain to function on a global scale with security and correctness guarantee, the shared public ledger needs an efficient and secure consensus algorithm, which must be built fault tolerant and ensure that first all nodes simultaneously maintain and identically chain no blocks and also that it does not rely on central authority to keep malicious adversaries from disrupting the coordination process of reaching consensus.

Consensus

- Every message transmitted between the nodes has to be approved by a majority of participants of the network
- A good consensus mechanism used in the blockchain implementation also ensures a robust transaction ledger with two important properties: persistence and liveness

So basically we are saying that in short every message transmitted between the nodes has to be approved by a majority of participants of the network through a consensus based agreement and also the network as a whole should be resilient to the partial failures and attacks such as when a group of nodes are malicious, they become malicious or messages in transit is corrupted. So, thus a good consensus mechanism used in the blockchain implementation also he also he shows a robust transaction ledger with two important properties persistence and liveliness liveness. Persistence guarantees the consistent response from the system regarding the state of transaction. For example, if one node on the network states that a transaction is in the state will stay then the other nodes on the network should also report it as stable if queried and responded to honestly and liveness

states that all nodes or processes eventually agree on a decision or a value by eventually it indicates that it may take a sufficient amount of time for reaching the agreement by combining persistence and liveness it showed that a transaction ledger is robust such that only authentic transactions are approved and become permanent.

To summarize this video, the role of blockchain in the Bitcoin system is to replace the central ized database with the authoritarian or central access control. Once some data has been recorded in the global ledger blockchain, it should be impossible to change the blockchain and by enforcing the majority agreement or what we call is consensus of update validity through consensus. It ensures the blockchain ensures the consistency state and prevents the double spending problem. So, the same Bitcoin cannot be spent two times because a majority or consensus verifies the transaction information. In a series of next two videos, we will discuss the dynamics of blockchain transactions.

What is a transaction in Blockchain?

- A transaction refers to a contract, agreement, transfer, or exchange of assets between two or more parties.
- A blockchain transaction would typically involve (a) data about the transaction (b) data about the participants, and (c) block specific data
- The three key elements to a blockchain transaction are (a) cryptographic keys, (b) a P2P network, and (c)a computer network to store and record transactions

A transaction refers to a contract, agreement, transfer or exchange of assets between two or more parties. Thus, it is typically like cash or property in general. In case of blockchain, it may be a Bitcoin. So like a normal transaction, a blockchain transaction is nothing but data transmission across the network of computers in a blockchain system. The network of computers in a blockchain store the transaction data as replicas with the storage typically referred to as a digital ledger.

Digital ledger as we discussed at the beginning of the lesson we discussed these concepts. A blockchain transaction could typically involve data about the transaction such as the date, time, amount of money paid, place etc. Data about the participants of the blockchain

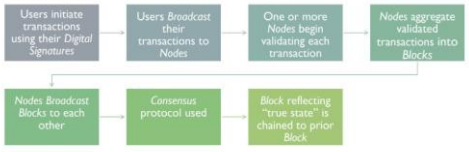
transaction or the username, identity maybe. Blockchain specific data sorry block specific data or hash a unique code that distinguishes one block from other. Now, for this transaction blockchain involves three key elements cryptographic keys, a peer to peer network, a computer network to store and record transactions.

A cryptographic key is a unique and digital secure identity reference used for managing and authorizing transactions. Upon merging with the peer to peer network, the digital signature is used by individuals on the network to reach a consensus on transactions. Once a deal is authorized, a mathematical verification certifies it resulting in a successful transaction between the two connected parties in the network. Besides financial transactions, blockchain are also capable to hold legal contracts, product inventories, transaction details of other assets like vehicle property among others. Now, a blockchain transaction has to undergo several steps before it becomes part of the blockchain.

A critical aspect of the blockchain technology being the way it authorizes and confirms transactions.

Steps of the Blockchain Transaction Process

1) Entry of a new transaction
2) Transmission of the transaction to a global network of peer-to-peer computers
3) Peer network of computers confirms the validity of the transaction
4) Confirmed legitimate transactions are clustered into blocks
5) The blocks are chained together to create a long history of all transactions
6) Completion of the transaction



Source: Blockchain Technology-Based Holiday Exchange Network
https://link.springer.com/chapter/10.1007/978981-99-3608-3_10

Let's see the steps below to highlight how a transaction takes place. So first entry of a transaction, the user initiates the transaction using digital signatures, then the transmission of the transaction to the global network of peer to peer computers. This is done through by users by broadcasting their transaction to nodes. Then you have a peer to peer network of computers confirm the validity of transaction.

First one or more nodes begin validating these transactions, then nodes aggregate the validated transactions into blocks. So once a particular transaction is verified, it is added

to blocks. Basically here the confirmed legitimate transactions are clustered into blocks. So with all that proof of stake or proof of work mechanism, a transaction is verified and then added to the block. These blocks, the new blocks are then chained together to create a long history of transactions.

So there is a series of blocks, new block is added and chained. Once it is chained to the previous chain, the chain is elongated further lengthens and as you keep on increasing the chain, as the name suggests, the block and the chain it forms. So block referring true state is chained to the prior block. So once it is verified, it is added to the existing chain of blocks and for that consensus protocol is used. To summarize, in this video, we introduced some of the steps in a blockchain transaction. In this video, we'll conclude our discussion on blockchain transactions.

Fundamentals of a Blockchain Transaction

- Hash encryptions: Blockchain employs hashing and encryption technology, mainly the Secure Hash Algorithms
- Authentication and authorization: Blockchain transactions are authenticated using cryptographic keys that are essentially strings of data identifying a blockchain user and giving access to their account on the system
- Mining: Means adding transactions to the distributed digital public ledger of existing transactions (or the blockchain)
- Proof of work: The decision to add a transaction to a public blockchain is made by consensus whereby a majority of the computers (nodes) in the network must agree to the validity of a transaction
- Proof of stake: Proof of stake is a validation consensus protocol in a blockchain for processing transactions and creating new blocks

One of the most significant attributes of blockchain transactions is its security. Let us look at the key aspects of blockchain technology that contribute to the safety of every blockchain transaction. First, you have hash encryptions. So blockchain employs hashing and encryption technology, mainly the secure hash algorithms, for example, SHA-256 algorithm and some other similar algorithms to ensure data security. Therefore, these secure hash algorithms transmit the transaction details as encrypted information like hash encryption, which gets added to the blockchain post verification.

Thanks to this secure hash algorithm, hash encryption becomes practically impossible to have. Next, you have authentication and authorization. So blockchain transactions are authenticated using cryptographic keys that are essentially strings of data, identifying a

blockchain user and giving access to their account on the system. The two cryptographic keys that ensure successful and secure transactions between two properties, are private and public keys. Using these keys, a blockchain user creates a secure digital identity for controlling and authorizing transactions.

Next, very important, you have mining process. In the blockchain technology, mining means adding transactions to the distributed digital public ledger of existing transactions or blockchain. Although primarily associated with Bitcoin, mining also applies to other blockchain using scenarios. The mining process involves generating a hash of a block of transactions. So hash is like some kind of cryptographic code, which is created by some hash function using the information contained by the block. Mining also applies to other blockchain usage scenarios and the mining process involves generating a hash of a block transaction.

Since the hash is unforgeable, it's difficult to forge hash, it protects the integrity of the entire blockchain without requiring a central system or central authority like central bank or a regulatory body. Next you have proof of work. So this is the decision to add a transaction to public blockchain. It is made by consensus whereby a majority of the computers or what you call as nodes or members in the network must agree to the validity of transaction. Thus people who own the nodes in the network need to solve a complex mathematical puzzle known as the proof of work.

Problem to add, proof of work problem to add a block to the chain, solving the proof of work problem is also called mining, which requires some energy consumption by people who are mining it, the miners. The people doing it are called miners and they are rewarded for verifying the transactions in the form of probably native currency like on Bitcoin blockchain, it will be Bitcoin. Then you have proof of stake mechanism. Now in the proof of stake mechanism is a validation consensus protocol in a blockchain for processing transactions and creating new blocks. It entails that blockchain participants must have a stake in the blockchain, typically by owning cryptocurrency, native, some native cryptocurrency or native currency to that blockchain.

Hence that native cryptocurrency owner gets a chance to validate transactions by offering their stakes as a collateral to their some kind of currency stake is on offer and alternative to proof of work. And this proof of stake mechanism saves significant computing power and resources, but it may not be as secure on large scale as proof of work, but saves on energy.

Fundamentals of a Blockchain Transaction

- A transaction represents an interaction between parties, for example, a transaction represents a transfer of the cryptocurrency between blockchain network users
- Inputs – The inputs are usually a list of the digital assets to be transferred
- Outputs – The outputs are usually the accounts that will be the recipients of the digital assets along with how much digital asset they will receive
- While primarily used to transfer digital assets, transactions can be more generally used to transfer data
- Regardless of how the data is formed and transacted, determining the validity and authenticity of a transaction is important

So all in all, a transaction here represents an interaction between parties. For example, with cryptocurrency, a transaction represents a transfer of cryptocurrency between blockchain network users. For business to business scenarios, a transaction could be a way of recording activities occurring on digital or physical assets also.

Each block in a blockchain can contain zero or more transactions for some blockchain implementations, a constant supply of new blocks, even with zero transactions is very important to maintain the security of the blockchain network. This is by having a constant supply of new blocks being published. It prevents malicious users from ever catching up and manufacturing a longer altered blockchain. The data which comprises a transaction can be different for every blockchain implementation. However, the mechanism for transaction is largely the same a blockchain network user sends information to the network.

So they send information to the network, the members of the network. The information sent may include the sender's address or any other relevant identifier send this publicly a digital signature transaction inputs and transaction outputs. Now a single cryptocurrency transaction typically requires at least one input and one output. Let's start with the inputs. The inputs are usually a list of digital assets to be transferred. A transaction will reference to the source of digital assets providing provenance either the previous transaction where it was given to the sender or for the case of a new digital assets, the original event, the event where it originated.

Since the input to the transaction is a reference to the past events, the digital assets do not change. In the case of cryptocurrencies, this means that the value cannot be added or

removed from existing digital assets. Instead of single digital assets can be split into multiple new digital assets, each with lesser value or multiple digital assets can be combined to form a fewer new digital assets with a corresponding greater value. This splitting or joining of assets will be specified within the transaction output.

The sender must also provide proof that they have access to the referred inputs. Generally by digitally signing the transaction and providing access to the private key which is in the form of public. So output is the next point. So the outputs are usually the accounts that will be recipients of the digital assets along with how much digital assets they will receive. So how much output specifies the number of digital assets to be transferred to the new owner, the identifier of the new owner and the set of conditions, the new owners must need to spend that value. If the digital assets provided are more than required, the extra funds must be explicitly sent back to the sender.

This mechanism to make change, this sort of make change mechanism. So more formally you may think of a transaction like there was 20 bitcoins which were owned by A. Now out of these 20, if 17 bitcoins are transferred, the 3 will be remaining, 17s are transferred to B. So now after this transaction, we will own the 17 and 3 will remain with A. While primarily used to transfer the digital assets, the transaction can be more generally used to transfer data. In a simple case, someone may simply want to permanently and publicly post data on blockchain.

In the case of smart contract systems, transactions can be used to send data, process the data and store some results on the blockchain. For example, a transaction can be used to change an attribute of a digitized asset such as the location of a shipment within blockchain technology based supply chain system. Regardless of how the data is formed and transacted, determining the validity and authenticity of the transaction is important. The validity of transaction ensures that the transaction meets the protocol requirements and any form live data formats or smart contract requirements specific to the blockchain implementation. The authenticity of a transaction is also important as it determines that the sender of digital asset had access to those digital assets. transactions are typically digitally signed by the sender's associated private key. There's asymmetric key cryptography, which we will discuss shortly, but that would include the public, that will include a public and private key, private key with which the initiator will sign the transaction, but it will be private to the owner. In the other member nodes, they will have this public key through which they will verify the transaction done by the private key. To summarize this video, blockchain technology is making headlines with its wide-ranging practical utilities in various sectors and industries. However, the most widespread and well-known blockchain usage is in cryptocurrency. Apart from crypto, the applications of blockchain technology extend to traditional investments using blockchain for financial

transactions that is fast growing and cost effective and allows investors greater control over their assets with no involvement of any third party.

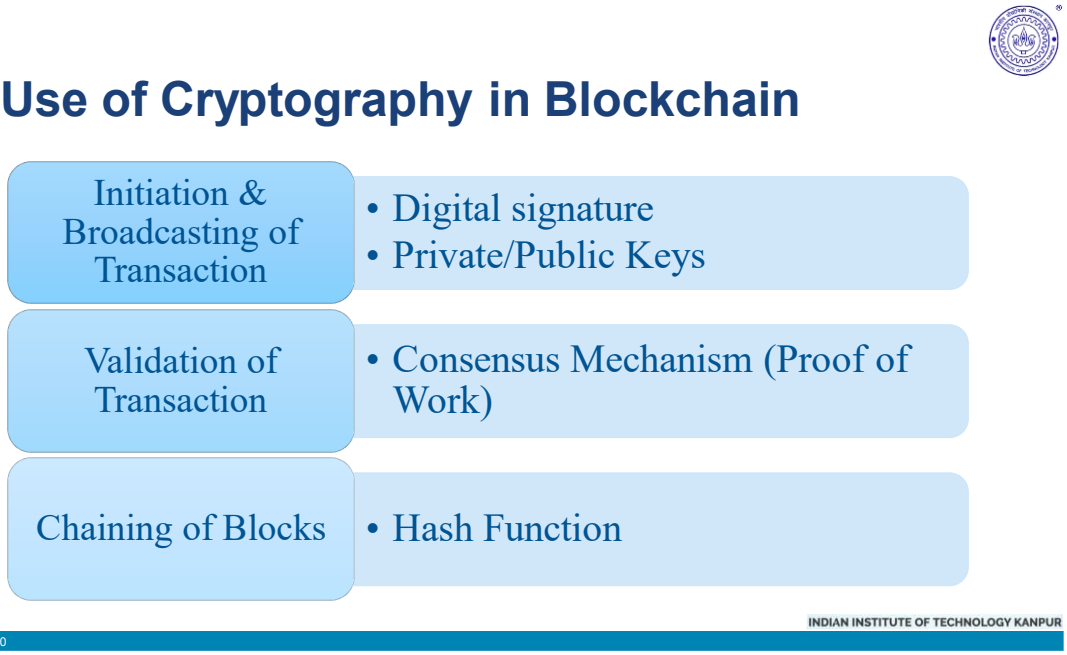
Fundamentals of a Blockchain Transaction

- All currencies need some way to control supply and enforce various security properties to prevent cheating
- Cryptography provides a mechanism for securely encoding the rules of a cryptocurrency system in the system itself
- Cryptography is a deep academic research field utilizing many advanced mathematical techniques that are complicated to understand.
- Three basic and important capabilities that are supported by the blockchain implementation in Bitcoin are: (1) the hash chained storage, (2) digital signature, and (3) the commitment consensus for adding a new block to the globally chained storage

Besides, blockchain operations are efficient, accurate and secure, making them ideal for sensitive operations in lending, insurance, real estate, voting, personal identity information and so on. Starting with this video, we will introduce cryptographic primitives in a series of videos from here onwards. Please note that all currencies, fiat currencies, central bank currencies, they need some way to control supply and enforce various security properties to prevent cheating. In fiat currencies or conventional regular currencies, organizations like central bank they control money supply and add entire counter-heating features to fiscal currency. Security features raise the bar for an attacker, but they don't make money impossible to counterfeit.

Unfortunately, law enforcement is necessary for stopping people from breaking the rules of the system and cryptocurrencies too must have security measures that prevent people from tampering with the state of the system and from equivocating that is making mutually inconsistent statements of different people. For example, if Alice convinces Bob that she paid him in a digital coin, for example, she should not be able to convince Carol that she paid her the same coin. But unlike fiat currencies, the security rules of cryptocurrencies need to be enforced purely technologically and without relying on a central authority. So as the name suggests, cryptocurrencies make heavy use of cryptography. Cryptography provides a mechanism for securely encoding the rules of cryptographic system in the system itself.

We can use it to prevent tampering and equivocation as well as to encode the rules for creation of new units of the currency into a mathematical protocol before we can properly understand cryptocurrencies. Then we will need to delve into cryptographic foundations that they rely upon. So cryptography is a deep academic research field utilizing many advanced mathematical techniques that are notoriously subtle and complicated to understand. Fortunately, Bitcoin only relies on a handful of relatively simple and well-known cryptographic constructions. For our discussions, we will specifically study cryptographic hashes, digital signatures and consensus mechanisms that provide to be very useful for building cryptographies.



Now, three basic and important capabilities that are supported by blockchain implementation are first the hash chain storage, digital signature and commitment consensus as we discussed for adding a new block to the globally chained storage. By an elegant combination of a suite of popular security techniques such as hash chain, digital signature and consensus mechanisms, this Bitcoin blockchain can prevent both the double spending problem of bitcoins that Alice can send the same coin to B and C. So, thus the Bitcoin blockchain can prevent both the double spending problem of bitcoins and also stop the retrospective modification of any transaction data that is immutability in a block after the block has been successfully committed into the blockchain. So what we are saying, let us just summarize our discussion here. So first, let us say Alice wants to perform a transaction, maybe give her 100 dollars of worth Bitcoin, give her 100 dollar worth of Bitcoin to B.

So, she will initiate and broadcast the transaction. So Alice will broadcast the transaction.

The transaction will have her digital signature which she did using private key and in her broadcast the public keys to that private transaction and some other transaction details will be there. So there will be member nodes. Now these member nodes will rush to validate the transaction. One of the nodes with some speed and capacity may earlier in time provide a proof of work mechanism to verify the transaction and his proof of work will be circulated across all the nodes who will confirm his or verify his proof of work and verify the transaction which is called consensus mechanism. So 51% of such nodes, they will verify this proof of work by this minor node and validate the transaction.

Once the transaction is validated, it will be added to the existing blocks, added to the existing blocks, the new block will be added. So it is sort of this new block will also include the hash, information hash, sort of hash of the previous block on its header. It will add that, it will sort of provide a linkage. So this hash will provide kind of you can think of linkage of the previous block to this block. So this hash will add as a linkage so previous blocks information will also be contained in the header of the new block.

So this is sort of summary of this entire cryptographic primitive of the transaction.

Hash function

- Hash functions are commonly used data structures in computing systems for tasks such as checking the integrity of messages and authenticating information.
- A hash function is a mathematical function with the following three properties-

1. Its input can be any string of any size.
2. It produces a fixed-size output (it's deterministic- so if you take a certain set of data, it will always give you the same hash)
3. It is efficiently computable.

In this video, we will discuss a very important component of cryptographic primitive that is hash function. Please note that blockchain technology can be seen to be very complex. However it can be simplified by examining each component individually. At a high level blockchain technology utilizes well-known computer science mechanisms and cryptographic primitives like cryptographic hash functions, digital signatures, asymmetric key cryptography, mixed with record keeping concepts such as append-only ledgers.

Here we must make a distinction between hashing and encryption. The main difference between encryption and hashing is that encryption is used to secure data whereas hashing is used to check the integrity of the data. Now this hashing is a very important component of blockchain technology and it is the use of cryptographic hash functions for many such operations. Hash functions are commonly used data structures in computing systems for tasks such as checking the integrity of messages and authenticating information. Hashing is a method of applying a cryptographic hash function to data which calculates a relatively unique output or provides a relatively unique output called a message digest or just digest. For an input of nearly any size, file, text or image, it allows the individuals to independently take the input data, hash the data and derive the same result proving that there was no change in the data.

Then the smallest change to the input, changing a single bit result in completely different output or digest. So you have a, let us say some data, maybe numeric, maybe image, in that data you apply a hash function that will transform into a digest or what you can say a hash for that information, probably a part of transaction information or something. And this digest is unique, even changing a small, small part of this data will result in a very unique and different hash. So thus a hash function is a mathematical function with three key properties.

First and foremost, its input can be any string of any size. It produces a fixed size output. It is efficiently computable. Intuitively, this means that for a given input string, you can figure out what the output of hash function is in a reasonable amount of time. Those properties define a general hash function, one that could be used to build a data structure such as a hash table.

Now we are going to focus exclusively on the cryptographic hash functions. Let us discuss the hash function. So for a hash function to be cryptographically secure, we are discussing the cryptographic properties of the hash function. For it to be cryptographically secure, we are going to require that it has the following additional security properties. First and foremost is called pre-image resistance or hiding. So hiding is the very first property that we want from our hash function. The hiding property assets that if we are given the output of the hash function, let us call it the output as y equal to hx, there is no feasible way to figure out the input.

So from knowing why we should not be able to figure out x. And this has the following implications.

Hash Function Cryptographic Properties

- For a hash function to be cryptographically secure, we're going to require that it has the following additional (security) properties (1) Preimage Resistance (Hiding), (2) Collision Resistance, and (3) Puzzle-friendliness.
- Preimage Resistant meaning it's one way. You can only go one way, meaning it's infeasible to determine the input from the output. It's infeasible to determine the x from the hash of x.
- Hiding. A hash function H is hiding if: when a secret value r is chosen from a probability distribution that has high min-entropy , then given H(r ‖ x) it is infeasible to find x .

First, it makes the hash function pre-image resistant. So it makes the hash function pre-image resistant, meaning that they are only one way, they are one way. It is computationally infeasible to compute the correct input x given the output. So for example, given the digest y or output, which was generated from a hash function, I should not be able to figure out x. Second, they are also pre-image, second pre-image resistance, that means that one cannot find an input that hashes to a specific output.

So there are second pre-image resistance also, which means one cannot find an input that hashes to a specific output, more specifically cryptographic hash functions are designed So that given a specific input x, it is computationally infeasible to find a second input which produces the same output. So if x produces the output y, and if there is another input x2 that produces the same output y, we should not be able to find that input where hash of x1 equal to hash of x2. So we should not be able to find any such x2 also which is able to produce the same output. The only approach available is to exhaustively which we call as brute force method.

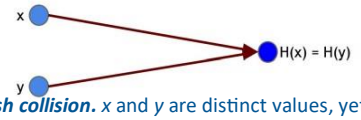
So the only approach which is to exhaustively search the entire input space. But this is computationally infeasible to do with any chance of success. So the only possibility is to do with the brute force approach and that should be very computationally intensive in a normal course of business. So to summarize hiding, a hash function h is hiding when a secret value r is chosen from a probability distribution that has a high min-trophy, then given h r, given x, it is infeasible to find that input x. Now, just to quickly summarize the min-entropy, in information theory, min-entropy is a measure of how predictable an outcome is and high min-entropy captures the intuitive idea that the distribution of that random variable is very spread out, very spread. What this means specifically is that when

we sample from the distribution, there is no particular value that is likely to occur.

So because the distribution is so far that none of the values are highly likely. So for a complete example, if you choose r uniformly among all of the strings that are bits long, then any particular string has a probability of getting chosen is so small like 2 to the power 256, 1 by 2 to the power 256 or put it more clearly 1 by 2 to the power 256 almost close to zero. So small probability of any string being chosen which means randomly you provide a random answer probability of you being correct about input x is almost close to zero. To summarize in this video, we discussed the very important property of the hash function which is hiding or pre-image resistance. In the next set of videos, we will discuss about collision resistance and puzzle friendliness. In this video, we will discuss two important properties of hash function that is collision resistance and puzzle friendliness.

Hash function cryptographic properties

- **Collision Resistant** A collision occurs when two distinct inputs produce the same output. A hash function H(.) is collision-resistant if nobody can find a collision [e.g., find an x and y which hash(x) = hash(y)].
  - A hash function H is said to be collision resistant if it is infeasible to find two values, x and y , such that $x \neq y$, yet $H(x) = H(y)$.



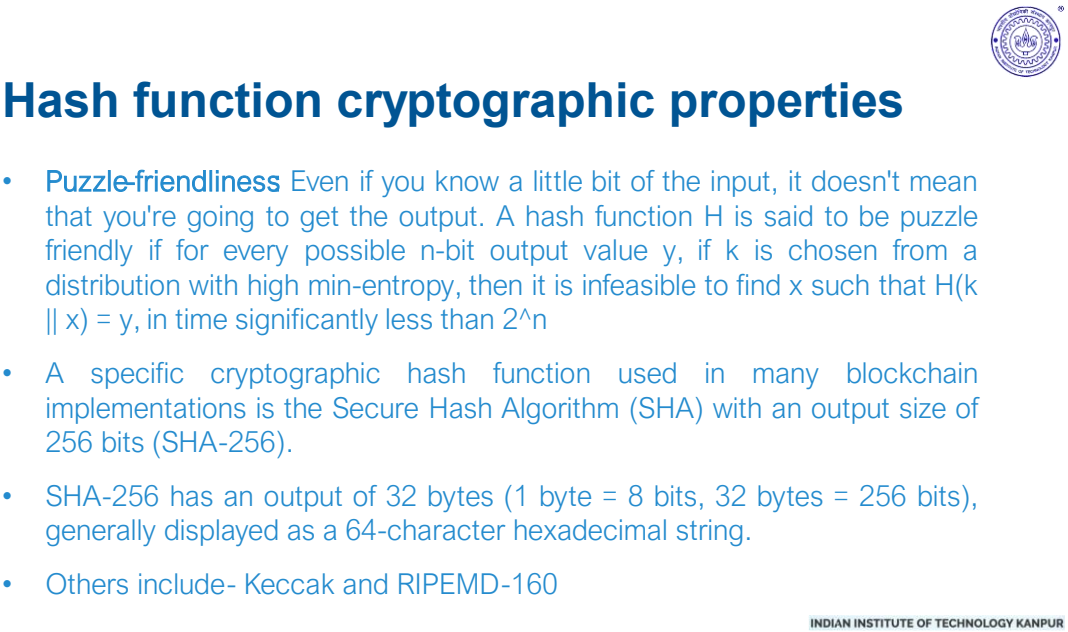
A hash collision. *x* and *y* are distinct values, yet when input into hash function *H* , they produce the same output.

Source: Bitcoin and Cryptocurrency Technologies(Page 24)
https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies

First, it is often said that hash function should be collision resistant. This means that one cannot or one should not be able to find two inputs that hash or result in a hash with same output. More specifically, it is computationally it should be computationally infeasible to find any such two inputs, maybe x and y that produce the same digest or same output from a hash function. That means the hash of x is equal to hash of y, this should not be there, then they are called collision resistant. A collision between x and y here it occurs where both of these distinct inputs produce the same output from the hash function like we said.

A hash function is collision resistant if nobody can find a collision like this. So here a hash collision where x and y are distinct values yet when they are input into a hash function, they produce the same output. So it should not happen idly which is called the property of collision resistance. A hash function is said to be collision resistant if it is infeasible to find two values x and y such that they are not equal to each other and yet hx equal to hy. Then

why should not be able to get this kind of relationship and if it is infeasible then they are said to be collision resistant.



Let us discuss the property of puzzle friendliness which means even if you know a little bit of input it doesn't mean that you are going to get the output by just randomly putting up values.

So let us discuss it in more detail. If a hash function h is said to be puzzle friendly, if for every possible n-bit output of value y, so y is your hash function output and if you choose k from a distribution with Heimann entropy which we have discussed, Heimann entropy we have discussed. So I have chosen k and the remaining part to the input which is some part, so the one component of x input is known as k, the remaining part it is infeasible to find the remaining part of x such that hk given x equal to y where x is the input and hash function applied to this hk equal to y, you can randomly you can find the remaining part of that x in a time significantly less than 2 to the power n. Now the non-mathematical interpretation here is more intuitive which says that if I take this idea as a puzzle, this hash function as a puzzle, then solving that puzzle is more convenient than randomly guessing the values and randomly guessing the values for the remaining. So out of the entire x input, I know k part of it.

So the remaining part I am randomly guessing and putting into it to find that x which produces this y through the hashing. It's not very feasible. It's almost infeasible and extremely time consuming like 2 to the power n which is extremely time consuming and computationally and it may take if I use some kind of brute force method by trying each possible value, it's very very time consuming. So instead of that it is more convenient to

form that hash function as a puzzle and solve that puzzle. So this is called puzzle friendliness. Now one specific cryptographic function, hash function that is used in many blockchain implementations is called secure hash algorithm, SHA with an output size of 256 bits often called SHA-256.

Many computers support this kind of algorithm in hardware making it fast to compute and efficient. SHA-256 has an output of 32 bytes where 1 byte equal to 8 bits and 32 bytes equal to 256 bits. Generally displayed as a 64-character hexadecimal string. This means that there are 2 to the power 256 or 10 to the power 77 possible digest values that can be produced by changing the combinations of these bits. Since there are an infinite number of possible input values and a finite number of, there is a large number of input, almost infinite values and a possible output values of y, it is possible but highly unlikely to have a collision where hash of x equal to hash of y that is hash of two different input produces the same digest. SHA-256 is said to be collision resistant since to find a collision in SHA-256 one would have to exclude the algorithm on average 2 to the power 128 times which is nearly infeasible.

There are other families of cryptographic hash functions utilized in the blockchain in addition to SHA-256 such as KICAC which was selected by NST as the winner of in competition in three hashing standard as well as RIPE EMD 160. So there are others also. To summarize, in this video we discussed the property of collision resistant of hash functions which means two inputs should not, two inputs x and y should not be producing the same output when passed through hash function. Next time we discuss puzzle friendliness that means a hash function should be more amenable to solving as a puzzle rather than randomly guessing all the possible values through some kind of brute force attack method.

It should be more easier and convenient to solve it as a puzzle.

Address derivation

- Address derivation: It is a short, alpha-numeric string of characters derived from the blockchain network user's public key using a cryptographic hash function. Most blockchain implementations make use of addresses as the "to" and "from" endpoints in a transaction.
- Public key ⟹ hash function ⟹ address
- Blockchain network users may not be the only source of addresses within blockchain networks

In the next two videos, we will discuss the application of hash functions. One very important application is transaction that is blockchain transactions. This we have discussed in the previous video. Another very important application is address verification. Some blockchain networks make use of an address which is a short alphanumeric kind of string of characters derived from the blockchain networks uses public key.

Using a cryptographic hash function along with some additional data may be version number, checksums and some identifiers. Most blockchain implementations make use of addresses as the to, to and from kind of endpoints in a transaction and these addresses are usually shorter than the public keys and are not secret. So one method to generate an address is to create a public key. So this public key applying a cryptographic hash function, we will apply a cryptographic hash function on the public key and convert that hash to text which will result in the address. Now each blockchain implementation may implement a different method to arrive at an address for permissionless blockchain networks for example, which allow anonymous account creation.

A blockchain network user can generate as many asymmetric KPS. What is asymmetric KPS? Asymmetric KPS are private public key pairs. So these are called asymmetric key pairs. And therefore addresses as many addresses as possible as many as asymmetric key pairs as many addresses allowing for a varying degree of pseudo anonymity and addresses may act as the public facing identifier. So this private key is for the user itself. This public key is like public facing identifier in the blockchain network for the user.

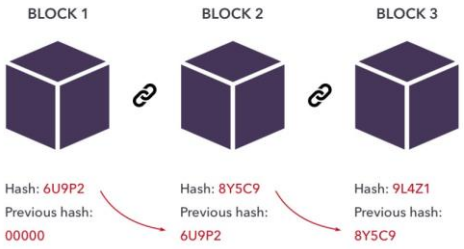
And oftentimes an address will be converted into a quick response code or QR code which

is a two dimensional barcode. So two dimensional kind of QR code which can contain some arbitrary data for easier use with mobile devices you can sort of scan the QR code with your mobile and verify the address. So the blockchain network users may not be the only source of addresses here within the blockchain networks. For example, they may not be the only users. It is also necessary to provide a method of accessing a smart contract once it has been deployed within a blockchain network.

For example, Ethereum smart contracts are accessible via a special address called a contract amount account. Now this contract account address is created when a smart contract is deployed. The address for a contract account is deterministically computed from the smart contract account address and this contract account allows for the contract to be executed whenever it receives a transaction as well as create additional smart contracts in turn.

Block Identification

- Creating unique identifiers.
- Block is a combination of the two words Block meaning block or piece and Chain meaning chain
- A blockchain's data structure is expressed as a linked list of blocks in which transactions are ordered



Source: Page 6: Application of blockchain in BIM: a systematic review
(PDF) Application of blockchain in BIM: a systematic review (researchgate.net)

Another very important application of hash function in identifying blocks creating a sort of unique identifiers because these blocks are linked through these hash chains. The subsequent blocks are linked through hash chains. So hash, these hash functions can also be used to identify as unique identifier they can be used because these blocks are combination of two words block meaning block or a piece and chain meaning chain.

So literally blockchain means it is a chain of blocks and a block is a space that can store this block, it can store certain amount of information by joining these blocks together a chain of blocks is obtained that contains interconnected data. So you have a chain of blocks that contains interconnected data. Now here this connection is made with hashes.

These hashes are used to make these connections. Hash is the same cryptographic code that

is based on the information in this block. So this the information this block is used to create hash which will connect these two blocks. Hashes are set of numbers and letters that contain the entire information of a block. When a block is written and its information is completed hash is assigned to it and its hash will be placed in the next block header to determine if the new block is a continuation of the previous block. So this hash will link the information of previous block and it will be put in the header of this next block so that they are linked. Also the production of these hashes continues and the previous block hash is placed on the next block and by following the hashes the first generated block can be used.

So you have each block linked so you can look at this block from here and then next block and go up till the first block or genesis block. So for example this hash is the first bitcoin block known as the Genesis block. So the first block in a blockchain would be called a Genesis block and anything can be turned into a hash and encrypted. So with the hashing algorithm large data can be reduced to a very large volume of data can be reduced to a very few set of letters, numbers and in this way the information of the block can be converted into a sort of multi bit expression. Here you can see the components of the blockchain so there is a block and this is chained together with the hash so the previous block is chained to the next block through this hash.

So as we saw here a blocks data structure is expressed as the linked list of blocks in which transactions are ordered. This data structure of the blockchain consists of two fundamental elements one is a pointers hash pointers and a linked list. A linked list is like a list of chained blocks with data and pointers to the previous block so the pointer to the previous block through this hash function or hash pointer. Pointers are like variables that refer to the position of another variable and a linked list a list of chained blocks with the data and pointers to the previous block and here we also use the Merkle tree structure which is like a binary tree of hashes. So it's like a binary tree of hashes this Merkle tree structure. This block contains the root of the hash of the Merkle tree and information like the preceding blocks hash, timestamp, non block version number and current difficulty goal.

So for blockchain system a Merkle tree provides security, integrity and irrefutability we discussed the Merkle tree in the previous videos. The blockchain system is built on Merkle trees cryptography and consensus algorithms because it is the first in the chain the genesis block which is the first in the chain the first block does not contain the pointer because there is nothing preceding it. To protect the security and integrity of the data contained in blockchain transactions are digitally signed and then a private key is used to sign the transaction so the private key is used to sign the transactions and anyone with the public key so this anyone with the matching public key may verify the signer. So this digital signature this combination results in a digital signature that detects the information manipulation because the data is encrypted and also sign digital signatures ensure unity.

As a result any manipulation will render the signature invalid so if there is any manipulation here it will render the signature invalid the signature will not be able to reveal or verify the data if there is some contamination with the data and the data cannot be discovered because it is encrypted it cannot be tampered with again even if it is caught. So the sender's or owner's identity is also protected by this digital signature as a result a signature is sort of legally linked to its owner and cannot be disregarded.

To summarize we discussed two applications of hash function which is address derivation transaction we already discussed and block identification. In the next video we will discuss consensus algorithms and securing the blockchain for chaining of blocks.

Consensus Algorithms

- In PoW, miners compete to find a hash value that meets certain criteria
- Solving a hash involves computing a proof-of-work, called a NONCE, or "number used once"
- The only way to find a valid proof-of-work is to run guesses through the algorithm
- Brute Force Search: Computing from right to left is called a "brute force" search

Source: Bitcoin and Cryptocurrency Technologies (page 33 & 34)
https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies

In this video we will discuss two key applications of hash functions that is consensus algorithm and securing and chaining the blocks. To begin with let us start with the example of cryptocurrency or bitcoin applications or blockchain. In the most famous cryptocurrency bitcoin it uses hash functions in the blockchain there are powerful computers they are often called miners and as a part of their proof of work they race with each other in brute force searches to try and solve such hashes in order to earn the mining rewards of native currency that is new bitcoins as well as some processing fee that users pay to record their transactions on the blockchain. Solving a hash function involves computing a proof of work called nonce or number used once that when added to the block causes the blocks hash to begin with a certain number of zeros once a valid proof of work is discovered the block is considered valid and can be added to the blockchain.

Now since each blocks hash is created by a cryptographic algorithm for example bitcoin is called as sha256 secured hash algorithm the only way to find the valid proof of work is to

run guesses through the algorithm until the right number is found that creates a hash that starts with the right number of zeros this is what bitcoin miners are doing running numbers through a cryptographic algorithm until they get the valid nonce number used once. Now there is something called brute force search method that we discussed so this brute force search basically it is possible to compute the input input to a hash input x to a hash this x input given the output hash value let us say output hash value is y and that involves lot of computing possible compute x using y which was computed with the hash function computing from this kind of computing from this side y to x is called brute force search using trial and error to find the message that fits the hash value and see produces the match so you try to find the message x which produces this hash value y using trial and error.

Securing the block data/ Chaining blocks

- Securing the block data/ Chaining blocks: Each block contains a hash of the previous block, forming a chain.
- If the data in any block is tampered with, it will change the hash value, breaking the chain and indicating an alteration
- To tamper the blocks, adversary has to tamper all the way up to the "Genesis Block"

Source: Bitcoin and Cryptocurrency Technologies (page 33 & 34)
https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies

Now next application here is securing the block data or chaining the blocks which we introduced earlier also here each block contains a hash of the previous block forming a chain like this a hash pointer is simply a pointer which points to the information like it points to the where information is stored together with the cryptographic hash of that information so the hash of this information is also there with this whereas a regular pointer would just give you the direction to retrieve the information a hash pointer also gives you a way to verify the information if it is an it has not changed because of this hash inside it hash of that information so if the data in any block is tampered so data is any let us say in any of this block is tampered let us say here I tamper with this block it will change the hash value so the hash value of this block will be changed and breaking the chain and indicating the alteration so because this will change the hash will change and therefore it will not match with this header where the hash and the pointer is stored hash pointer is stored so a hash pointer is basically a pointer where the data which points to where the data is stored along with the cryptographic hash of the value of that data so it also has that cryptographic

hash so you alter with this block it will not match with the hash here and this kind of hash pointer is used to build a linked list of blocks using hash pointer so this linked list is created with produce hash pointer so they are chained together with these hash pointers now whereas in a regular linked list where you have series of blocks each block has data as well as pointer to the previous block in the list in a blockchain the previous block pointer will be replaced with a hash pointer so each block not only tells us where the value of the previous block was but it also contains a digest of that value that allows us to verify that this is the output of the hash function so it allows us to verify that the value hasn't changed we store the head of the list which is just a regular hash pointer that points to the most recent data block so head here the head pointer here header of each block will store this hash pointer which will be linked to the which will be the linking mechanism through previous block in the head rate that hash pointer data will be stored a use case for blockchain is tamper evident block that is we want to build a long data structure like this that stores a bunch of data and allows us to open the data open the new data on to the end of the log but if somebody alters that data that is earlier in the log if this some of the earlier previous blocks are tampered with we are going to detect it let us understand how so if the data in any block is tampered with it will change the hash value in the header of the next block breaking the chain and the casing alteration so if this block is changed the header it will not meet with the header match with the header here in the next block to understand why a blockchain achieve this tamper evident property let us ask what happens if an adversary wants to tamper with the data that is in the middle of the chain specifically the adversary's goal is to do in such a way that someone who remembers only the hash pointer at the head of the block chain won't be able to detect the tampering so I should be having this side of the block he should be able to take the tampering not to achieve this goal adversary changes the data of some block key so if he changes this key block data since the data has been changed the hash in the block k plus 1 so the hash here this a plus 1 pointer which is a hash of the entire block K the information here in this K is not going to match so this header which contains the hash of this block K will not match now remember that we are statistically almost guaranteed that the new hash will not match the altered content since the hash functions are called in the middle you can refer to the previous video they are called collision resistance so the two information pieces cannot result in the same hash and so we'll detect the inconsistency inconsistency between the altered block here K and the hash pointer in K plus 1 of course the adversary can continue to try and cover up the chain by changing the next block also so you can use you can try and change this block also and they can continue doing this but this strategy will fail when they reach the head of the list specifically as long as we store the hash pointer at the head the first genesis block in Genesis block in some store place where the adversary cannot change it they will be unable to change any block without being detected so unless until they also are able to change all the blocks this thing because this essentially because of that avalanche effect if you recall the hash pointer here is a combination of all the previous block information so

unless until they are able to change up till here till up till the end entire thing this will not they will not be able to achieve this hash and the upshot of this is that if the adversary wants to tamper with the data anywhere in the chain in order to keep the story consistent they have to tamper with the hash pointers all the way back up to the beginning and they are ultimately going to run into a roadblock because they won't be able to tamper with the head of the list the final head for which we have secured and we probably we have kept the somewhere kept it somewhere the information of this header and thus it emerges that by just remembering the single hash pointer at the first block we have essentially remembered a tamper evident hash of the entire list so the entire list so unless until they are able to change all the information of this block including the header block so in this fashion we can build a blockchain like this containing as many black blocks as we want we can build as many blocks as we want going back to some special box that may be the genesis block at the beginning of the list so we call it genesis block for which information or the entire information is kept in a secure manner which can't be accessed at any cost and therefore even if we the adversary modifies all the blocks free earlier in time they won't be able to if as long as they won't be able to touch the final block they it becomes tamper evident so i would know that the blocks have been tampered because my header hash here is not matching with the previous blocks to summarize in this video we discussed two key important applications of hash functions that is consensus and securing and changing the problem to summarize this lesson blockchain is a combination of three important technologies namely cryptographic public and private keys a peer-to-peer network and digital ledger on blockchain a transaction is verified and authorized by nodes on a peer-to-peer network through some consensus mechanism such as proof of work or proof of stake which is a kind of mathematical puzzle solving these technologies are employed to secure the digital ledger using blockchain this requires application of hash functions these hash functions generate hash pointers which along with the merkle tree structure anchor each block to its previous block and make the blockchain tamper evident and tamper resistant for any malicious attack to modify the information it also has to modify the information in a large number of such blocks to successfully change the transaction the transaction originator verifies the transaction with her private key and announces by sending the public key and other important information on the same network to the nodes and these participating nodes or miners verify this transaction through some kind of consensus mechanism which requires majority of the member nodes to verify the transaction information hash functions such as secure hash algorithm such as sha 256 converts the transaction information contained in each block into a cryptic hash code these hash functions have some key properties for example they are collision resistant that is two different inputs cannot result in the same output code they are pre-image resistant or what we call as hiding property that is using the output one should not be able to generate the input message but for each input there is a unique output lastly they are puzzle friendly that is a random brute force attack which randomly tries all the combinations of the code should

not be able to or should be too time consuming as compared to solving the cryptographic code as a mathematical puzzle lastly we noted that hash functions can be used in creating blockchain addresses act as unique block identifiers facilitate consensus mechanisms and lastly in securing the block through chaining process.