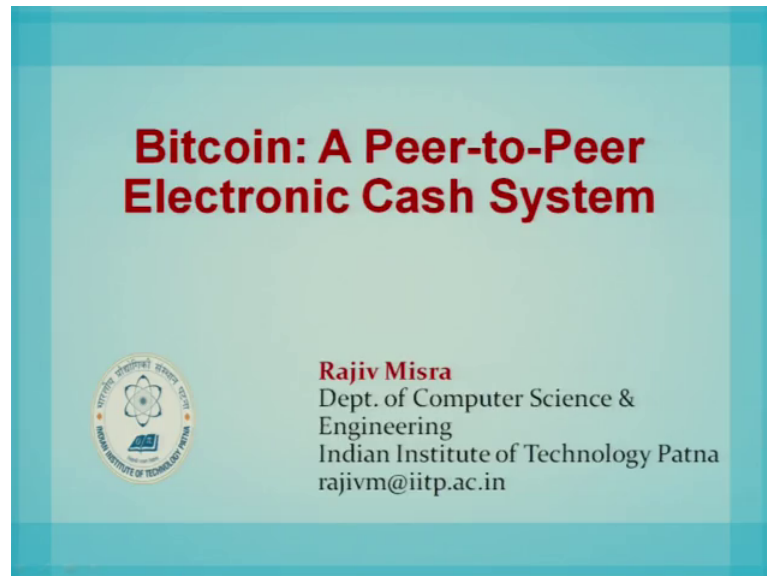


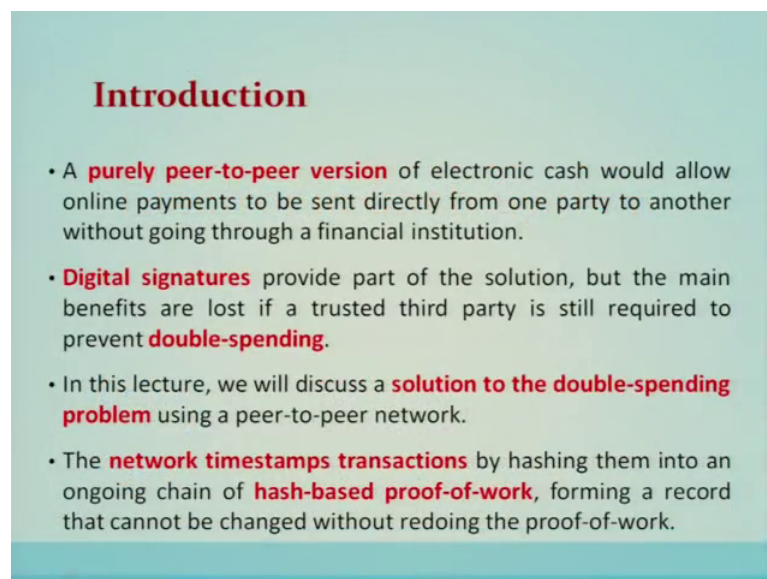
**Distributed Systems**  
**Dr. Rajiv Misra**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Patna**

**Lecture – 26**  
**Bitcoin: A Peer-to-Peer Electronic Cash System**

(Refer Slide Time: 00:16)



(Refer Slide Time: 00:19)

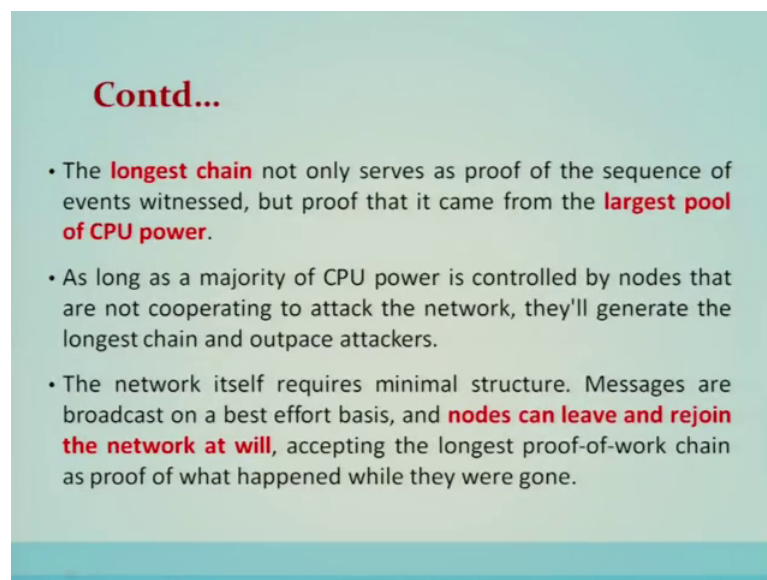


Bitcoin a peer to peer electronic cash system, introduction a purely peer to peer version of electronic cash would allow online payment to be sent directly from one party to

another without going through the financial institution. Digital signatures provide part of the solution, but the main benefit are lost if the trusted third party is still required to prevent double spending. This lecture we will discuss the solution to the double spending problem using a peer to peer network.

A network timestamp transactions by hashing them into an ongoing chain of hash based proof of work forming a record that cannot be changed without redoing proof of work. The longest chain not only serves as a proof of sequence of the events witnessed, but proof that it came from the largest pool of CPU power. As long as the majority of CPU power is controlled by the nodes that are not cooperating to attack the network we will generate the longest chain and outpace the attackers.

(Refer Slide Time: 1:18)



**Contd...**

- The **longest chain** not only serves as proof of the sequence of events witnessed, but proof that it came from the **largest pool of CPU power**.
- As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers.
- The network itself requires minimal structure. Messages are broadcast on a best effort basis, and **nodes can leave and rejoin the network at will**, accepting the longest proof-of-work chain as proof of what happened while they were gone.

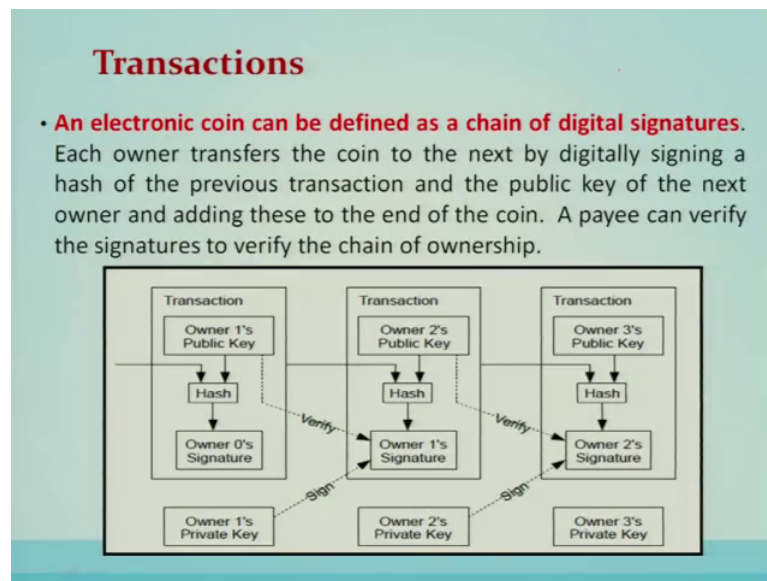
The network itself is minimal structure messages are broadcast on the best effort basis and the nodes can leave and rejoin the network at will accepting the longest proof of work chain as a proof of what has happened while they were gone. So, in a nutshell before we go ahead into more discussion into more detail, let us see that this part of the discussion is basically a case study of a applying peer to peer to peer system concept of decentralization.

So, let me give you an example that between the transaction if let us say A one to give some money to B, one to transfer cash it requires to involve the bank then only the fund can be transferred from A to B with the help of a bank. Or if it is online transaction then

basically it is called a payment gateway like these are and other things. So, this particular way of the transfer of money is basically in the form of a client and server manner. So, this is these are all call server. So, they are the clients, which they want to transfer the money with the help of a server. Now there are different issues how this particular centralized system can be made the distributed form of or the how it can be decentralized. So, that between two parties A and B can directly transfer their money without involving the bank or any payment gateway and this is done with the help of using the decentralization technology which is called a peer to peer systems.

So, we will see in this particular discussion how Bitcoin using peer to peer system has able to achieve this kind of decentralization without involving centralized system like bank or a payment gateway. Yet it ensures all kind of safety and security and avoid the double spending issues in this part of the work. So, what is the transaction? So, electronic coin can be defined as a chain of digital signatures.

(Refer Slide Time: 04:15)

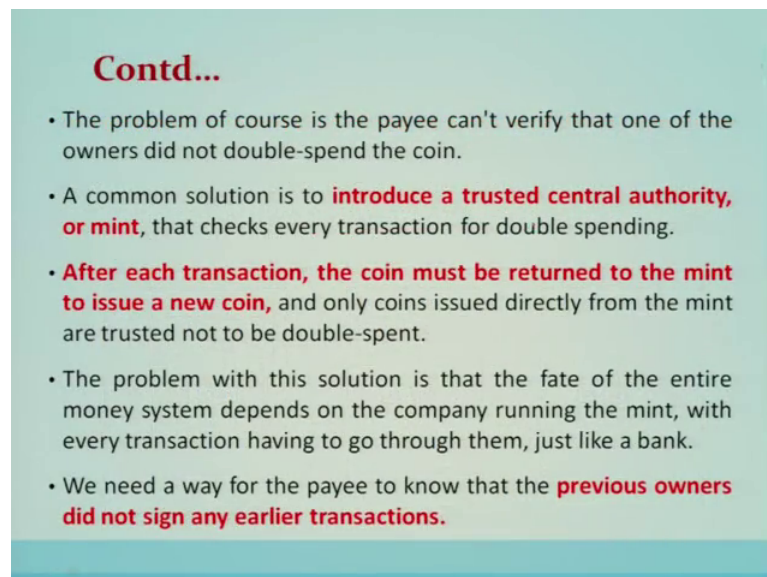


So, the each owner transfers a coin to the next by distantly signing a hash of previous transactions and the public key of the next owner and adding these to the end of the chain a payee can verify the signature to verify the chain of the ownership. Meaning to say that Bitcoin owner has a private key, the other part which is called a public key is basically stored in a thing which is called the block chain.

So, if a request for a money is basically sent by a user. So, it has to sign digitally using the private key and the private key will basically indicate he is the owner of that Bitcoin or a digital signa or digital currency by signing the digital signature which is nothing, but a private key. So, an electronic coin can be defined as a chain of digital signatures. So, each owner transfers the coin to the next by digitally signing the hash of a previous transactions and the public key of the next owner and adding these to the end of the coin. So, basically here these particular transactions are chained together and this is shown here in the picture. So, every transaction is chained with the previous transactions and so on. So, just see that this particular transaction is chained with the previous transaction and this in turn chain with the previous transactions and so on.

So, whenever a new transaction comes it will be joining at the tail of the transactions and this particular process at the end of the coin and the payee can verify the signature to verify the claim of the owner. So, we are going to see all these aspects how the how the transactions are basically implemented so that it can avoid the double spending of a coin. So, the problem of course, is that paying payee cannot verify that one of the owners did not double spend the coin.

(Refer Slide Time: 06:40)



**Contd...**

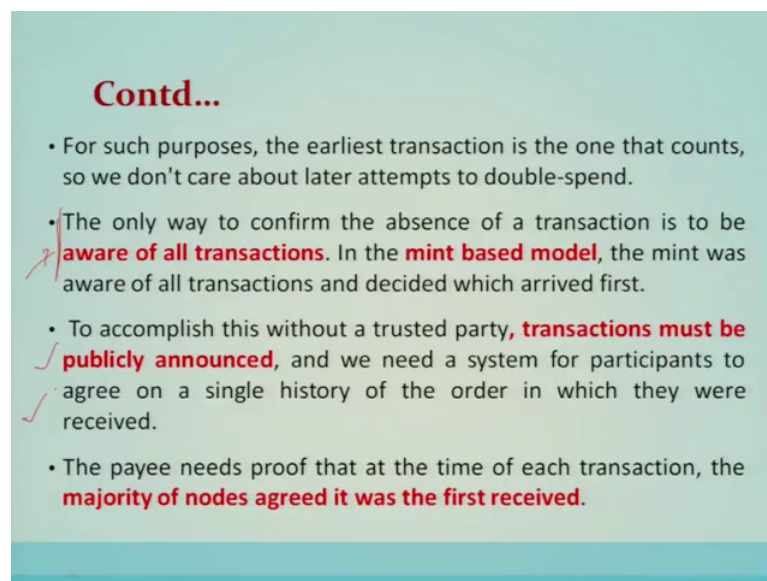
- The problem of course is the payee can't verify that one of the owners did not double-spend the coin.
- A common solution is to **introduce a trusted central authority, or mint**, that checks every transaction for double spending.
- **After each transaction, the coin must be returned to the mint to issue a new coin**, and only coins issued directly from the mint are trusted not to be double-spent.
- The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.
- We need a way for the payee to know that the **previous owners did not sign any earlier transactions**.

So, a common solution is to introduce a trusted central authority or a mint, but for that purpose we are going to go back again in a scenario which is basically a client over and

involving a payment gateway; how you without involving the payment gateway and a bank how this all how this double spending can be avoided.

So, after each transaction the coin must be returned to the mint to issue a new coin and only coin issue directly from the mint are trusted not to be double spent. The problem of this solution is that the fate of the entire money system depends on the company running the mint that is the bank. We need a way for the pay to know the previous owners did not signed any earlier transactions. So, the only way to confirm the absence of a transition is to be aware of all the transactions. So, this is basically the way in the mint based model the mint was aware of all the transactions, but if we are going for a decentralization who will be how this particular awareness of all the transactions is being incorporated implemented. So, to accomplish this without a trusted third party a transaction must be publicly announced and we need a system participants to agree on a single history of the order in which they are received.

(Refer Slide Time: 08:08)



**Contd...**

- For such purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend.
- The only way to confirm the absence of a transaction is to be **aware of all transactions**. In the **mint based model**, the mint was aware of all transactions and decided which arrived first.
- To accomplish this without a trusted party, **transactions must be publicly announced**, and we need a system for participants to agree on a single history of the order in which they were received.
- The payee needs proof that at the time of each transaction, the **majority of nodes agreed it was the first received**.

So, the payee needs a proof that at the time of each transactions the majority of the node agreed it was first received.

Now, timestamp servers; the solution begins with the timestamp server timestamp server works by taking the hash of the block of item to be time stamped and widely publishing the hash such that such as in the newspaper or in a use net post. The timestamp proves that the data must have existed at the time; obviously, in order to get into the hash.

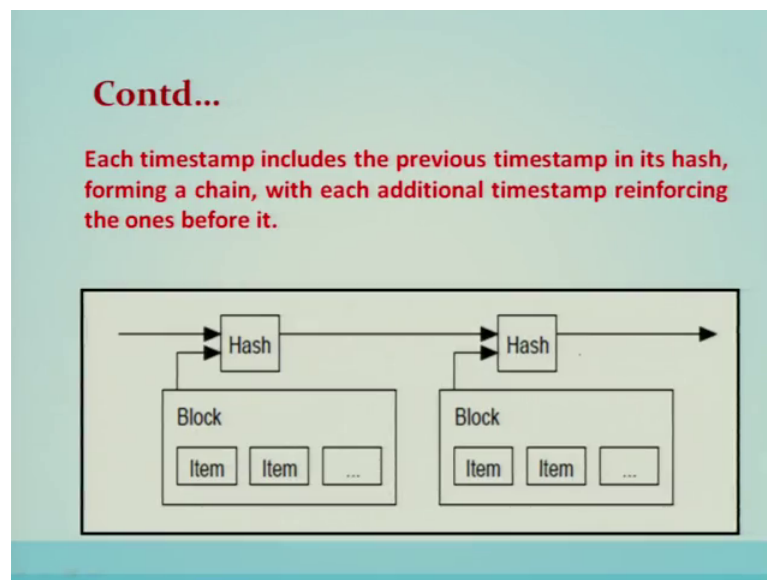
(Refer Slide Time: 08:45)

### Timestamp Server

- The solution begins with a **timestamp server**.
- A timestamp server works by taking a **hash of a block of items** to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post.
- The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash.

So, each timestamp includes the previous timestamp in its hash forming a chain with each additional timestamp reinforcing the one before it.

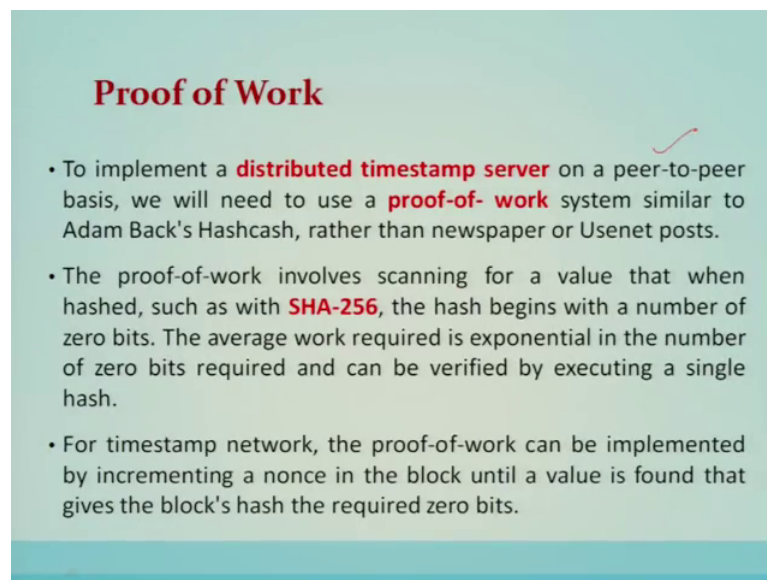
(Refer Slide Time: 09:01)



So, here we can see that these timestamps are nothing, but they are the hash and they are being chained with each other, and this particular chaining will indicate that a particular transaction at that instant it has happened and if it is in the chain; and this will also give a proof of work.

To implement the distributed timestamp server on a peer to peer basis, we need to use a proof of work system instead of publishing it on a newspaper. So, proof of work involves scanning of the values that one hashed such as SHA-256, and the hash begins with the number of 0 bits.

(Refer Slide Time: 09:30)



**Proof of Work**

- To implement a **distributed timestamp server** on a peer-to-peer basis, we will need to use a **proof-of-work** system similar to Adam Back's Hashcash, rather than newspaper or Usenet posts.
- The proof-of-work involves scanning for a value that when hashed, such as with **SHA-256**, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.
- For timestamp network, the proof-of-work can be implemented by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits.

The average work required is exponential in the number of 0 bits required and can be verified by executing a single hash. For a timestamp network the proof of work can be implemented by incrementing a nonce in the block until the value is found that gives the blocks hash the required zero bits, now who will be doing this? The peers who will be doing this kind of work for the proof of work they are called miners. So, all the miners will try to find out the solution of this particular puzzle and the ones and the minor who basically gives the solution for the first then he will be awarded with the bitcoin.

So, once the CPU, it requires to compute this particular proof of work it requires the CPU effort.

(Refer Slide Time: 10:54)

**Contd...**

- Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

```
graph LR; B1[Block] --> B2[Block]; subgraph B1; direction TB; PH1[Prev Hash]; N1[Nonce]; T1[Tx]; T2[Tx]; T3[...]; end; subgraph B2; direction TB; PH2[Prev Hash]; N2[Nonce]; T4[Tx]; T5[Tx]; T6[...]; end;
```

So, the block cannot be changed without redoing the work in this particular manner later blocks are changed after that the work to change the block would include redoing all the blocks after that. So, see all the blocks are being changed in this particular manner. So, if somebody want to change a particular block, it has to be changed at all the places then only the change is possible and computing it and doing changes is not possible why because its computationally expensive and many peers together they are solving this particular they are maintaining it and they are solving it this particular puzzle.

So, proof of work also solves the problem of determining the representation in a majority decision making. So, if the majority were based on 1 IP address, one vote it could be supported by anyone able to allocate many IP's.



(Refer Slide Time: 11:52)

### Contd...

- To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes.
- It can also be shown that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.
- To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Now to compensate for increasing the hardware speed and varying interest in running nodes over the time of proof of work difficulty is determined by moving an average targeting and average number of blocks per hour. If they are generated too fast the difficulty increases, hence it is basically very very difficult for the attacker to basically do the double spending and the timestamps or work is basically its nothing, but they are using the peer to peer concept of this particular hash chains and this has chain computations by the attacker is basically very very difficult computationally hence it is store and proof of work basically is established.

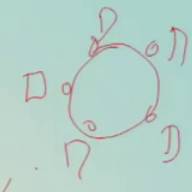
Now, the network the steps to run the networks are as follows new transactions are broadcast to all the nodes. So, all the peers involved. So, whenever a new transaction comes it has to be broadcasted to all of them all the nodes. So, each node collects new transition in a; into a block, each node works on finding a difficult proof of work.

(Refer Slide Time: 13:04)

**Network** ✓

The **steps to run the network** are as follows:

- 1) New transactions are broadcast to all nodes. ✓
- 2) Each node collects new transactions into a block. ✓
- 3) Each node works on finding a difficult proof-of-work for its block.
- 4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
- 5) Nodes accept the block only if all transactions in it are valid and not already spent.
- 6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.



For its block when a node finds a proof for work it broadcast, the block to all the nodes nodes accepts the block only if all the transactions in it are valid and not already spent node express their acceptance of the block by working on creating the next block in the chain using the hash of the accepted block of the previous hash.

So, nodes always consider the longest chain to be the correct one and we will keep working on extending it. So, if 2 nodes broadcast a different versions of the next block simultaneously, some nodes may receive one or the other first. In that case they work on the first one they receive and that will be the winner and that will be awarded by the bitcoin.

(Refer Slide Time: 13:25)

### Contd...

- **Nodes always consider the longest chain to be the correct one** and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first.
- In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.
- New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

Hence these selfish miners or a peers they basically are responsible to implement this particular proof of work and they are awarded also in this particular process.

So, the new transition broadcasts do not necessarily need to reach all the nodes and basically the majority will be useful. So, incentive by convention the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block this adds an incentive for the node to support the network and provides a way to initially distribute the coins into the circulation. Since there is no central authority to issue them as I told you already. So, the steady addition of a constant amount of new coins is analogous to gold miners, expanding the resources to add gold to circulations.

(Refer Slide Time: 14:57)

## Incentive

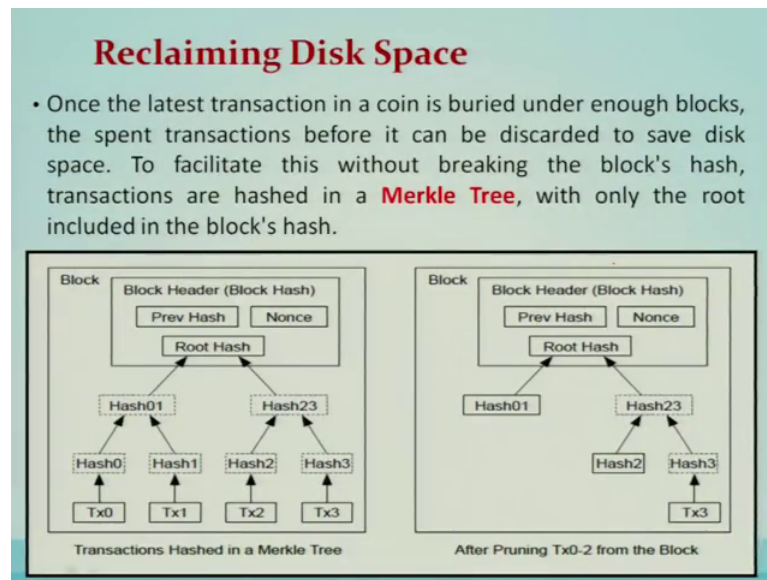
- By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block.
- This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. ✓
- The steady addition of a constant amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.
- The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction.

So, here in our case, this mining is nothing, but involving the CPU time and the electricity and that is why the miner who basically establishes or starts a new coin will get the incentive for that. So, incentive can also be funded with the transaction fees, if the output of the output value of a transition less than the input value the difference is the transaction fees that is added to the incentive value of the block containing the transaction.

So, the incentive may help encourage the nodes to stay honest. So, if a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to different people by stealing back his payments or using it to generate the new coins. He ought to find it more profitable to play by the rules such rules that favor him will be more new coins to everyone else combined.

Reclaiming the disk space;: So, once the latest transaction in the coin is buried under the enough blocks by spent transaction before it can be discarded and to facilitate it basically a Merkle Tree is being used.

(Refer Slide Time: 16:01)



So, Merkle Tree is nothing, but all these transactions if you see they are they basically pass through one time hash function and they will generate a hash and they will combine these hash and finally, root hash will come and this particular root will have the knowledge if there is any change at any point, then basically it cannot be recomputed and that is why that is how it will be verified that some changes has happened nobody no transaction can now change afterwards. So, hence the Merkle Tree is basically a solarization. So, to facilitate this without breaking the blocks hash transactions are hashed in a Merkle Tree like this with only the root include in the blocks hash. So, this will get the complete information with the root hash itself.

A simplified payment verification; it is possible to verify the payment without running the full network node, the user only need to keep a copy of the block headers of the longest proof of work chain which he can get by querying the network nodes.

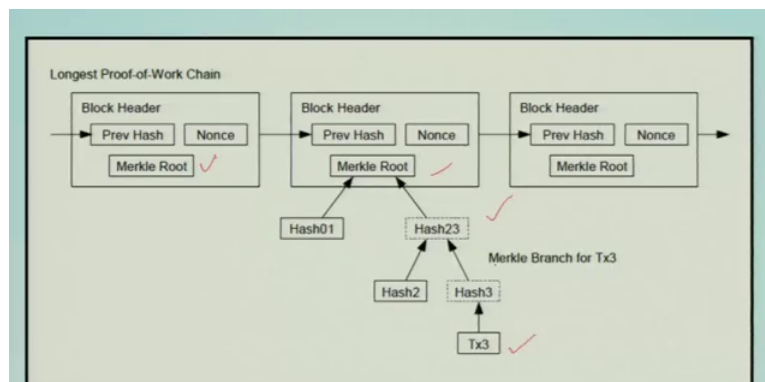
(Refer Slide Time: 17:25)

## Simplified Payment Verification

- It is possible to verify payments without running a full network node.
- A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in.
- He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.

Until he is convinced that he has the longest chain and obtain the Merkle branch linking the transaction to the block its timestamp in. He cannot check the transaction for himself, but by linking it to a place in the chain he can see that a network node has accepted it and the block added after it further confirms the network has accepted it. So, this is the example of a longest proof of chain, here you see that each block will contain only the Merkle root.

(Refer Slide Time: 18:03)



**Figure:** As shown in the figure, a user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in.

And here the Martin root will basically keep the transaction that particular link which basically is linked to a transaction which is linked to a Merkle node and this will be the indication of proof of work that is the longest chain.

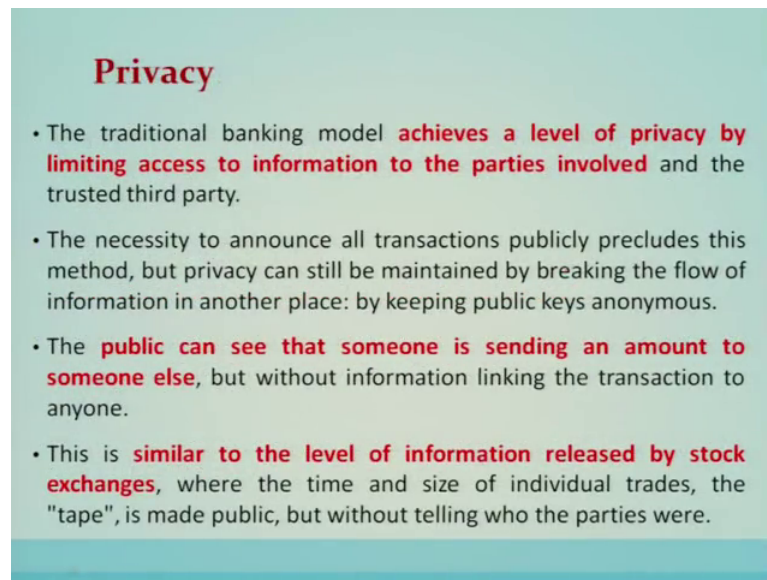
So, as shown in the figure the user only need to keep the copy of the block headers of the longest proof of work chain, which he can get by querying the network nodes until he is convinced that he has the longest chain and obtain the Merkle branch linking the transaction to the block a timestamp in. So, combining any splitting value, although it would be possible to handle the coins individually it would be unwieldy to make a separate transaction for every cent in a transfer to allow the value to be split and combined transaction contains multiple input and output that is shown over here.

(Refer Slide Time: 18:57)

### Contd...

- Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs:
  - one for the payment, and
  - one returning the change, if any, back to the sender.
- It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here.
- There is never the need to extract a complete standalone copy of a transaction's history.

(Refer Slide Time: 19:02)



**Privacy**

- The traditional banking model **achieves a level of privacy by limiting access to information to the parties involved** and the trusted third party.
- The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous.
- The **public can see that someone is sending an amount to someone else**, but without information linking the transaction to anyone.
- This is **similar to the level of information released by stock exchanges**, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

So, the privacy; the traditional banking model achieves the level of privacy by limiting access to the information to the parties involved and trusted third party, the necessity to announce all the transaction publicly precludes this method of privacy, but the privacy can be maintained by breaking the flow of information in another place by keeping public keys anonymous a public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone.

This is similar to the level of information released by the stock exchanges, where the time and size of the individual trades the tape is made public, but without telling who the parties were. So, you see the in a traditional privacy model there will be a trusted third party and custard third party the transaction having an identity linked, when they approaches with trusted third party then basically it can be verified. In a new privacy model all the there is no need of the identity that transactions are made public and they are basically implementing the privacy in this particular system.

So, the calculations in the form of for example, in honest chain and an attacker chain can be characterized as a binomial random walk, the success event is the honest chain being extended by one block increasing its lead by plus 1 and the failure event is the attacker chain being extended by one block reducing the gap.



(Refer Slide Time: 20:40)

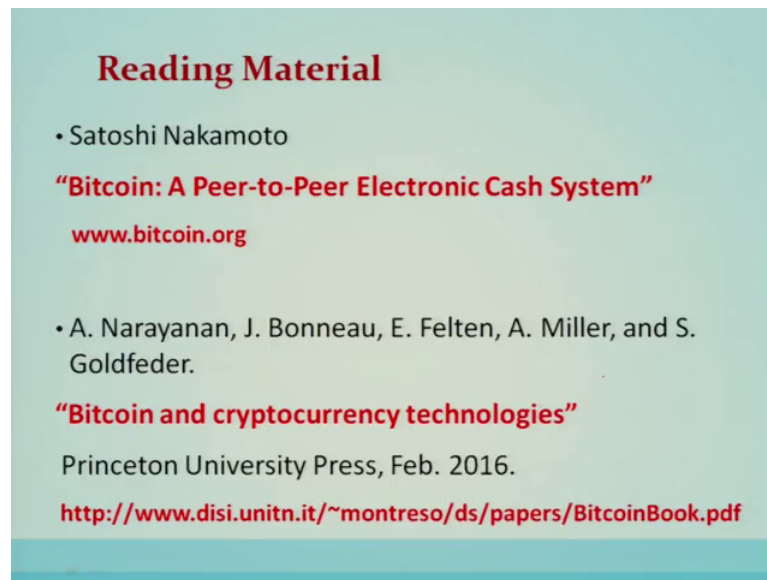
### Calculations

- Consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker.
- Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.
- The race between the honest chain and an attacker chain can be characterized as a **Binomial Random Walk**. The success event is the honest chain being extended by one block, increasing its **lead by +1**, and the failure event is the attacker's chain being extended by one block, reducing the **gap by -1**.

So obviously, the honest if the if the honest nodes are more compared to the dishonest one, then basically this particular gap will keep on increasing and will never be able to overpower and basically this particular chain keeps on growing in this manner.

So, this can be modeled using poisson distribution and the probability of attacker could still catch up now multiply the poisson density and summing up will rearranging to avoid the summing the infinite tail of the distribution and so, it is basically computationally becoming very very impossible to find out the length of the longest chain or to construct a length of the longest chain. Now reading materials are available for a Bitcoin a peer to peer electronic systems and Bitcoin and cryptocurrency technologies.

(Refer Slide Time: 21:31)

A slide with a light teal background and a dark teal border. The title "Reading Material" is in bold red text. Below it, the first entry is by Satoshi Nakamoto, with the title "Bitcoin: A Peer-to-Peer Electronic Cash System" in red and the URL "www.bitcoin.org" in red. The second entry is by A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, with the title "Bitcoin and cryptocurrency technologies" in red and the publisher "Princeton University Press, Feb. 2016." in black. The URL "http://www.disi.unitn.it/~montreso/ds/papers/BitcoinBook.pdf" is in red at the bottom.

**Reading Material**

- Satoshi Nakamoto  
**"Bitcoin: A Peer-to-Peer Electronic Cash System"**  
[www.bitcoin.org](http://www.bitcoin.org)
- A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder.  
**"Bitcoin and cryptocurrency technologies"**  
Princeton University Press, Feb. 2016.  
<http://www.disi.unitn.it/~montreso/ds/papers/BitcoinBook.pdf>

Conclusion in this lecture we have discussed a system for electronic transaction without relying on the trust, we have started with a usual framework of coin made from the digital signatures, which provides strong control of ownership, but is incomplete without the way to prevent the double is spending. To solve this we have discussed the peer to peer network using the proof of work to record the public history of the transaction that quickly becomes computationally impractical for an attacker to change, if the honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity nodes can leave and join the network at will and work their CPU power expressing the acceptance of the valid blocks.

(Refer Slide Time: 21:50)

## Conclusion

- In this lecture, we have discussed a **system for electronic transactions without relying on trust**.
- We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to **prevent double-spending**.
- To solve this, we discussed a **peer-to-peer network using proof-of-work** to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power.

So, they needed the rules and incentives they can be enforced with this consensus mechanism.

Thank you.