

**Blockchains Architecture, Design and Use Cases**  
**Prof. Sandip Chakraborty**  
**Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture – 52**  
**Research Aspects - V (Algorand - I)**

Welcome back to the course on Blockchain architecture. So, we are discussing about the several research aspects in blockchain in terms of a protocol and consensus scalability. So, that today we will discuss a very interesting topic which is floating for the last 1 year in the community in the block chain community and it claims to have nice and scalable consensus architecture over block chain and accordingly proposes a new crypto currency.

(Refer Slide Time: 00:52)

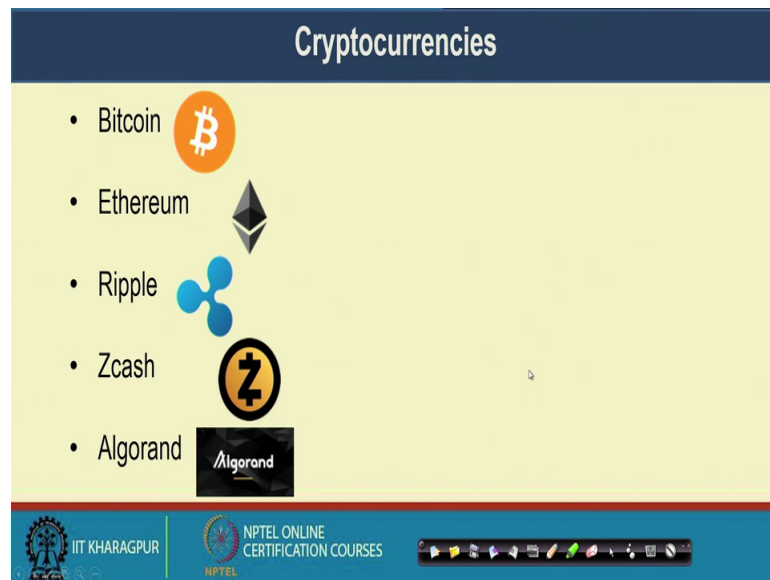


The slide features a blue cube icon with four lines extending from its corners to small circles, resembling a network or data structure. To the right of the icon, the following text is displayed: "Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017, October). *Algorand: Scaling byzantine agreements for cryptocurrencies*. In *Proceedings of the 26th Symposium on Operating Systems Principles 2017* ACM." Below this, the title "Algorand: Scaling Byzantine Agreements for Cryptocurrencies" is written in large, bold, orange letters. At the bottom of the slide, there are logos for IIT Kharagpur and NPTEL Online Certification Courses, along with a navigation bar.

So, the idea is to have a new crypto currency call the Algorand which was came from MIT artificial intelligence and computer science laboratory and the paper called the Algorand: Scaling byzantine agreement for cryptocurrencies.

The paper got published in last year SCM SOSB conference. So, we will discuss about this topic in details and there is Algorand website and the code for Algorand is also available publicly. So, you can explore it further by browsing to the Algorand website.

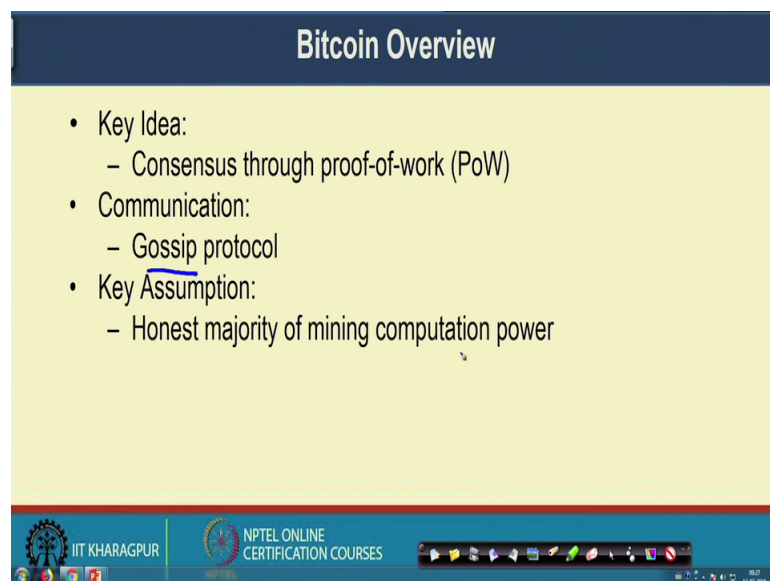
(Refer Slide Time: 01:35)



The slide is titled "Cryptocurrencies" in a dark blue header. Below the header, on a light yellow background, is a list of five cryptocurrencies with their respective logos: Bitcoin (orange circle with a white 'B'), Ethereum (black diamond with a white 'E'), Ripple (blue three-lobed shape), Zcash (black circle with a white 'Z'), and Algorand (black square with the word 'Algorand' in white). At the bottom of the slide, there is a blue footer containing the IIT KHARAGPUR logo, the text "NPTEL ONLINE CERTIFICATION COURSES", and a navigation bar with various icons.

So, let us go through the journey of Algorand. So, this concept of Crypto currencies, it is started from Bitcoin. an Then we have seen series of different Crypto currencies starting from Bitcoin to Ethereum, then Ripple, then Zcash and the finally, this Algorand and as this different kind of cryptocurrency progress they have kind of nice properties and people have thought of to improving different aspects like scalability, consensus finality and the security over the existing crypto currencies protocol and that way you have seen the evaluation of multiple crypto currencies which came into practice.

(Refer Slide Time: 02:23)



The slide is titled "Bitcoin Overview" in a dark blue header. Below the header, on a light yellow background, is a list of three key points: "Key Idea:" with a sub-point "Consensus through proof-of-work (PoW)", "Communication:" with a sub-point "Gossip protocol", and "Key Assumption:" with a sub-point "Honest majority of mining computation power". At the bottom of the slide, there is a blue footer containing the IIT KHARAGPUR logo, the text "NPTEL ONLINE CERTIFICATION COURSES", and a navigation bar with various icons.

So, let us look into the problems of Bitcoin which was there and the why or what is the basic motivation behind the design of a new crypto currency and then, we will go to the basic features of algorithm and discuss algorithm and architecture of Algorand in a higher level. So, I will not go to the very intrinsic details of for the cryptography protocols and mathematical proofs of Algorand. I will try to motivate you about the nice concepts and highlight the innovations which are there in the design of Algorand. So, let us look into a brief overview of Bitcoin.

So, the key idea Bitcoin is the consensus through proof of work. So, Bitcoin implies this proof of work mechanism to reach consensus, where the communication is through gossiping. So, every node in a Bitcoin network, they gossip it each other for the transactions as well as the blocks and that way they try to reach in a consensus cryptographic consensus and adding a block to the existing block chain and the key assumption in the Bitcoin proof of work design is that the honest majority of the mining computation power. So, it is like that the mining computation power is distributed among multiple miners and that way at different round of Bitcoin different miner will win and they will work as a leader to propose a new block.

So, that way the idea is that at based on the proof of work mechanism at a single round, a new miner will be get elected and that miner will propose the new block and that way we will have different miners at different round. So, no one will be able to control the entire work single headedly.

(Refer Slide Time: 04:22)

**Bitcoin: Technical Limitation**

- Resource wastage:
  - high computational, electricity cost
- Concentration of power
  - only ~5 mining pools control the entire system
- Vulnerable
  - easy to track miners
- Scalability
  - number of users not clear (1M, 10M, 100M??), high latency(~10minutes)
- Ambiguity
  - fork in blockchain

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, from there we move towards the technical limitations of Bitcoin. So, there are multiple limitations which have been there in the Bitcoin architecture. So, the first problem in Bitcoin which has been OLX plot in the existing literature is the resource wastage problem. So, whenever you are utilising these mining procedure to add a new block in the existing block chain, you require a high competition power to solve the hash problem which is associated with the Bitcoin proof of work mechanism.

So, as a result you require a huge amount of computation power to search for the nonce value which will produce the intended hash and as a result we have seen that many people or all the miners are actually trying to mine a new block, but only one or a very few of them will win to mine a new block and as a result the total wastage of power is significantly higher..

So, you have requirement of this high computational power and the electricity cost which is associated with the Bitcoin a protocol. Then, another limitation of Bitcoin is the concentration of the mining power. So, we have discussed about this concept of mining pools and with the concept of mining pools, you have seen that only around 5 mining pools they basically control the entire system and interestingly you know that who are those mining pools and where those mining pools are actually belongs.

So, as a result you can always track that where are those mining pools and it is easy to abide up or to corrupt those mining pool or to take a control of those mining pools. So,

vulnerability becomes have whenever know that who the miner is and who are actually controlling the entire mining procedure. So, if you just go to this block chain dot org website; and there you can see that this entire block generation mechanism of Bitcoin mining that is being controlled by around 5 or 6 different mining pools. So, that concept that concept of mining pool, although it help people to join in a mining pool and then participate in the mining procedure, but it is also one of the biggest threat in the sustainability of Bitcoin proof of work.

Then, as we have mentioned that the entire architecture is vulnerable because it easy is it is easy to track miners whenever you have some handful number of mining pools. The next problem that we discussed earlier that, the scalability is the major issue in block Bitcoin. So, there are 2 types of scalability. The scalability in terms of number of nodes and number of users that can be supported and a scalability in terms of transaction throughputs. So, we have already seen that scalability in transaction in terms of transaction throughput, it is very limited in case of Bitcoin. You can almost have around 7 to 9 transactions per second.

On the other hand, it is not very clear that how many number of simultaneous users can be supported in Bitcoins. So, whether it is 1 million or 10 million or 100 million. So, the limitation is not known and the because of this high latency which is approximately 10 minutes for committing a new block, your transaction throughput is very less. And then, another major problem in Bitcoin is this consensus finalities; so, absence of consensus finality which leads to ambiguity. So, there is always a possibility of fork and whenever there is a possibility of fork, you have multiple different paths in the block in the blockchain and as a result you do not know that which path is the current accepted path.

So, Bitcoin normally utilise this 50 percent rule where if you get a multiple copies of the current blockchain from your peers, they and all the or say you are disubing 2 different copies of the blockchain from your peers and both the copies have say equal chain length. In that case, you will accept the blockchain which is coming from majority of your peers. So, there is this ambiguity, but because of this ambiguity or because of this fork and there is no such consensus finality. There is always a possibility of wastage of resources. So, this orphan blocks are actually wasting a number of a handful of resources for the miners.

Because the miners are actually mining that block, but ultimately those particular block; although it is getting a valid block, but that block is not included in the longest chain of the blockchain which is there in the Bitcoin blockchain. So, these are the major technical limitations which are there in bitcoin.

(Refer Slide Time: 09:41)

**Algorand: Overview**

- Key Idea:
  - Consensus through Byzantine Agreement Protocol
- Communication:
  - Gossip protocol
- Key Assumption:
  - Honest majority of money

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And with this limitation, we give you a brief overview of Algorand. So, the key idea is Algorand, key idea of Algorand is consensus through byzantine agreement protocol based on the byzantine fault tolerance that we have discussed earlier. The communication is again gossiping and the key assumption is the honest majority of money. So, that you can find out that, who the peer is and you can find out the valid peers what they are in the system.

(Refer Slide Time: 10:14)

**Algorand: Technical Advancement**

- Trivial computation
  - simple operation like add, count
- True decentralization
  - no concentration of mining pool power, all equal miners and users
- Finality of payment
  - fork with very low probability, block appears and payment fixed forever
- Scalability
  - millions of users, only network latency (~1minute)
- Security
  - against bad adversary

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, these are the technical advancement of algorithm in comparison with the existing crypto currency like Bitcoin. So, first of all the operations which are introduced in Algorand, they are very trivial; simple operations like adds or counts. So, we do not have operations or costly operations like finding out the nonce to compute the target as well which actually results in a huge wastage of computational resource and mining power and it provides a true decentralization which is in contrast to the mining pool concept which is there in Bitcoin.

So, there is no concentration of mining pool power all are equal miners and there is equal users. So, anyone in Algorand, they can join in the mining procedure. So, as such Algorand does not call it as a mining because we are using the concept of this byzantine agreement here and as a result any user who is participating in the Algorand network, they can mine a new block and because the operations are very trivial and very simple, anyone can join and you do not require a complex mechanism or a complex system or a resourceful system to participate in the mining procedure then there is this finality of the payments.

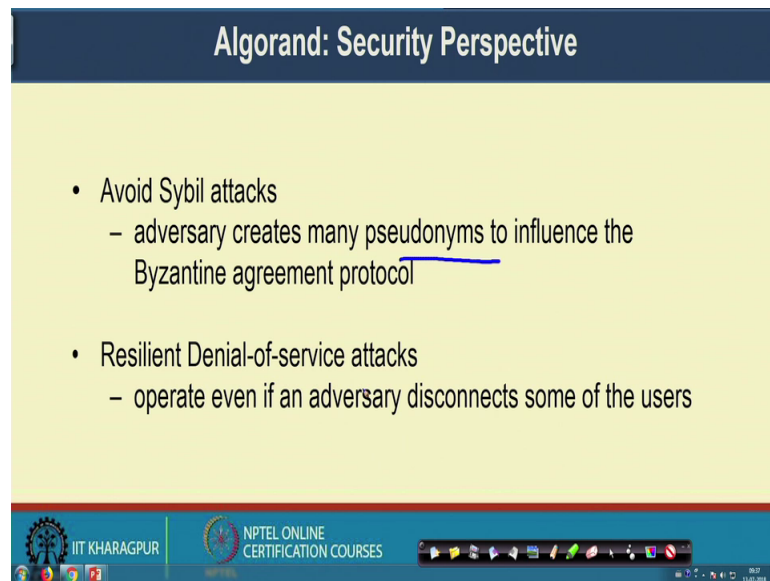
So, in Algorand, you can have a fork with very low probability; so, the probabilities around  $1$  by  $10$  to the power  $18$ ; so, which is a very very small number. Now, because of this finality of the payment, the block which are appears in a blockchain and the you

have certain transactions in that particular block, if the block is get added in the blockchain.

So, the transaction which is there inside the block they are final which is in contrast to Bitcoin. In case of Bitcoin even if a block is getting added to the blockchain, you do not know whether that block is going to be an orphan block in the future time instance or not which is not there in Algorand because of this consensus finality..

Then, the concept of scalability- So, Algorand can support millions of users and the delay of block commitment is only equal to the network latency. So, network latency is a kind of bottleneck which is there. So, if you have a good network you can have a quick consensus protocol. So, approximately in a general network it is close to around 1 minute and then, it provides a strong security; so, the security against bad adversary.

(Refer Slide Time: 13:05)



The slide is titled "Algorand: Security Perspective" and lists two main security features:

- Avoid Sybil attacks
  - adversary creates many pseudonyms to influence the Byzantine agreement protocol
- Resilient Denial-of-service attacks
  - operate even if an adversary disconnects some of the users

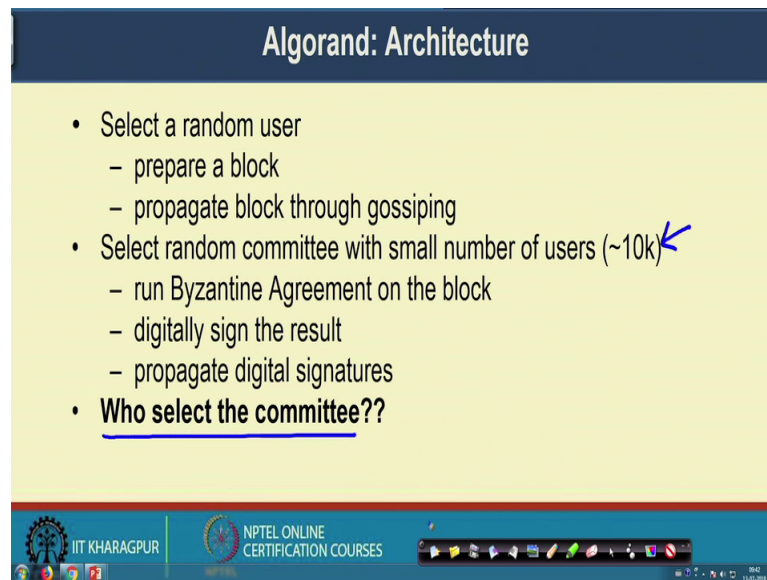
The slide footer includes the IIT KHARAGPUR logo, NPTEL ONLINE CERTIFICATION COURSES text, and a Windows taskbar with the time 11:03:28.

So, let us look into the security perspective of Algorand. So, it avoids Sybil's attack. So, as we have discussed earlier in Sybil attacks, the adversary creates many pseudonyms to influence the byzantine agreement protocol.

So, Algorand is free from this kind of Sybil attacks and it is also resilient from denial of service attack. So, it can operate even if an adversary disconnects some of the users from the Algorand network. Well. So, now, let us look that how the protocol actually works in practice.



(Refer Slide Time: 13:43)



The slide is titled "Algorand: Architecture" and contains the following content:

- Select a random user
  - prepare a block
  - propagate block through gossiping
- Select random committee with small number of users (~10k) ←
- run Byzantine Agreement on the block
- digitally sign the result
- propagate digital signatures
- Who select the committee??

At the bottom of the slide, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a system tray showing various application icons and a clock.

So, there are primarily 2 different steps in Algorand. So, assume that you have millions of users. So, what we do that in the first round, similar to Bitcoin this entire Algorand protocol works in rounds and every round at every round at the end of a round a block is getting added in the existing blockchain. So, at the beginning of a round the first task is to select a random user among the set of users who are participating in the Algorand procedure.

So, this random user prepares a block and propagates the block through gossiping. So, any of the users they can collect the transactions and after collecting the transaction, they can propose a new block. Now, the the node or the user who is going to propose a new block that is being selected by a nice algorithm called Cryptographic Sortition, we will come to that point. But the idea is that you are randomly selecting an user who will propose the block. Now that particular user which you are selecting for proposal of a new block that user may be a valid user or that user may be an adversary.

So, we require a kind of byzantine agreement protocol to find out that whether the block which is being proposed by the user by the random user, whether it is a valid block or not. So, for that we again select a random committee with a small number of users. Now here is the catch. So, if you remember the traditional byzantine agreement protocols, this byzantine agreement protocol like the byzantine fault tolerant even a practical byzantine

fault tolerant. They are they have over head. So, they are not scalable in terms of number of nodes.

So, what we are doing here in case of Algorand not all the users in the network, they are participating in the committee for participating in the byzantine agreement protocol rather we are selecting a small committee and that small committee will execute the byzantine agreement and that way it will make the inter system scalable. So, the idea here is that you select a random committee with a small number of users who will run the byzantine agreement on the block.

Now, if they find out that the block is some nice the block is correct, then they will digitally sign the result and propagate this digital signature in the network. Now, assume that well, you are in the network you are in the Algorand network and you have had a new block which is been proposed by a particular miner or a particular user Algorand user. Now, whenever after receiving that block if you are waiting say you are waiting for the digital signatures corresponds to block which are supposed to be coming from the committee members.

Now, if you are receiving some number of signatures; mood and certain threshold, then you are sure that well this particular block has been signed or this particular block has been agreed upon by though that many number of committee members which means that the block is actually a correct block. So, you can connect this correct block to the existing blockchain. So, that way to this entire architecture becomes very scalable. Now, there is an interesting question that who will select the committee.

Because if you have a node who is going to select the committee, the adversary can control that node and that way and that way we can control the internet work and remember that Algorand actually targets not a permission model rather it is considering a permission less model; that means, it is a open model where anyone can join in the network. Now, whenever you are in permission less model, you cannot have a single node or a single god who will select the committee.

(Refer Slide Time: 17:57)

**Cryptographic Sortition**

- Each committee member selects himself according to per-user weights
- Implemented using verifiable random functions (VRFs)

$\langle \text{hash}, \text{proof} \rangle \leftarrow \text{VRF}_{\text{sk}}(x)$

- $x$ : input string
- $(pk, sk)$ : public/private key pair
- hash: hashlenbit-long value that is uniquely determined by  $sk$  and  $x$
- proof: enables to check the hash indeed corresponds to  $x$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, the question comes who will select the committee and that is another interesting concept in Algorand that each committee member select himself according to per user weights.

So, the idea is that we have certain magic which will select every user or every user can elect himself as a part of the committee. So, it is it is just like a lottery kind of algorithm. You are running a lottery, if you are winning the lottery; then you have a proof that you have won the lottery and you can show that proof and participate in the committee. Now this particular concept is very interesting in the context of scalability. The idea here is that we do not have any distributed algorithm or decentralised algorithm that is actually electing a committee. So, the committee members are electing themselves.

So, as a result what is happening here that the individual committee members, they will just run certain local computations in their own machine and find out whether they are winning that lottery or not. If they are winning that lottery, they can participate in the committee and participate in validating that particular block which has been proposed in the current round. Now, at every different round we design this election procedure or we design this lottery mechanism in such a way so that at every individual round different members are elected as a part of the committee. Now, if different members are elected as a part of the committee at different round, then the interesting fact is that the adversary will not be able to take control of the committee members.

So, that concept is used in Algorand. So, we actually realise this particular concept with the help of something called Verifiable Random Functions or VRF's. So, this is a cryptographic sortition mechanism where every committee member they are individually selecting them self. So, they are running lottery protocol through which they will find out whether they are winning the lottery or not and if they are winning the lottery, they will show a proof. Now, what is a verifiable random function? So, this verifiable random function takes 2 parameter.

There is an input string and there is a public private key pair. So, you have a public key called  $p_k i$  and a private key pair at  $s_k i$  for round  $i$ . Now you are generating this verifiable random function with this private key pair. So, this private key is with every individual user. So, no one will be able to tamper the user. Now once this VRF function is executed, then it generates 2 parameters 1 hash and 1 proof. So, this has is a hashlenbit-long value that is uniquely determined by  $s_k$  and  $x$ . So, with the value of hash, you can uniquely identify a particular user.

And then, there is a proof this particular proof if someone has the public key  $p_k i$ , then they can check whether the hash indeed corresponds to  $x$ . So, if the hash is indeed corresponds to  $x$ ; that means, this particular proof will say that well, the user who is claiming to be the leader in the current round; then that particular member the um committee member has actually executed the protocol correctly, execute this VRF correctly and have has a proof that the he or she has won in the current round. So, that way this verifiable random function gives the nice architecture of this entire procedure.

So, we are not going to the formal proof of VRF. So, if you are interested you can dig into more detail inside VRF.

(Refer Slide Time: 22:00)

The slide features a dark blue header with the title "Cryptographic Sortition: Selection Procedure" in white. Below the header, on a light green background, is the function signature  $\langle \text{hash, proof, } j \rangle \leftarrow \text{Sortition}(\text{sk}, \text{seed}, \text{threshold}, \text{role}, w, W)$ . A curved arrow points from the function name to a small circular icon of a person's face. To the right of the icon is a bulleted list of parameter definitions. At the bottom of the slide, there is a blue footer containing the logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a standard Windows taskbar.

**Cryptographic Sortition: Selection Procedure**

$\langle \text{hash, proof, } j \rangle \leftarrow \text{Sortition}(\text{sk}, \text{seed}, \text{threshold}, \text{role}, w, W)$

- **seed**: publicly known random value
- **threshold**: determines the expected number of users selected for that role
- **role**: user for proposing a block/ committee member
- **w**: weight of a user
- **W**: weight of all users
- **j**: user gets to participate as  $j$  different "sub-users."

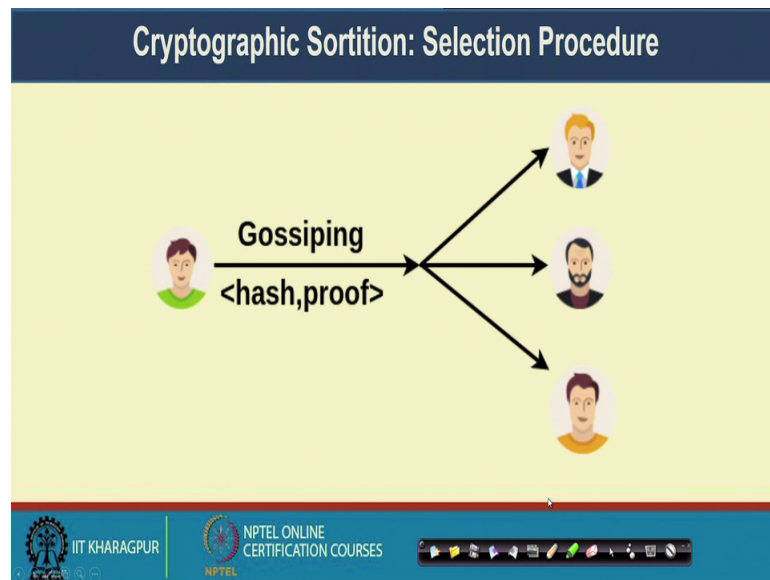
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this is the cryptographic sortition procedure which is used in member selection the committee member selection is Algorand. So, it takes some parameters the  $s$   $k$  value, a seed value that particular seed value is generated based on the current round number. So, this seed value actually ensures that at every different round, different shapes of members are being elected as a part of the committee. So, that way at every individual rounds the committee member gets changed.

And because the committee member gets changed at every individual round, the adversary has no way to control the committee members. You put a value threshold value, the role the role of the users in terms of proposing a new block; that means, the initial random users which we are going to select or whether the whether the user is going to be a committee member. Then,  $w$  is a weight of a user and capital  $W$  is the weight of all the users in the network.

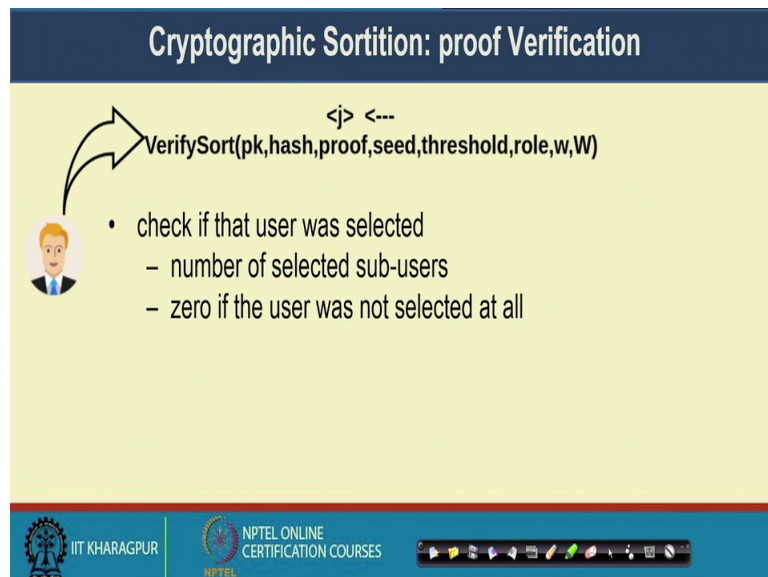
So, here this weight values comes from the factors like how much money the individual users have. That way the idea is there that this will give a way to individual users to be elected as a committee member or to be elected as a user to proposing a new block and  $j$  is the value like the user gets to participate as different sub user. So, this returns these 3 value hash proof along with this  $j$ .

(Refer Slide Time: 23:45)



So, this entire procedure works in this way. So, once you have this hash and a proof value, you make a gossiping among the pairs in the network. And you can show the proof that you have won the current round and that way you can be a member for the committee of the current round or you have the power to propose a new block well.

(Refer Slide Time: 24:06)

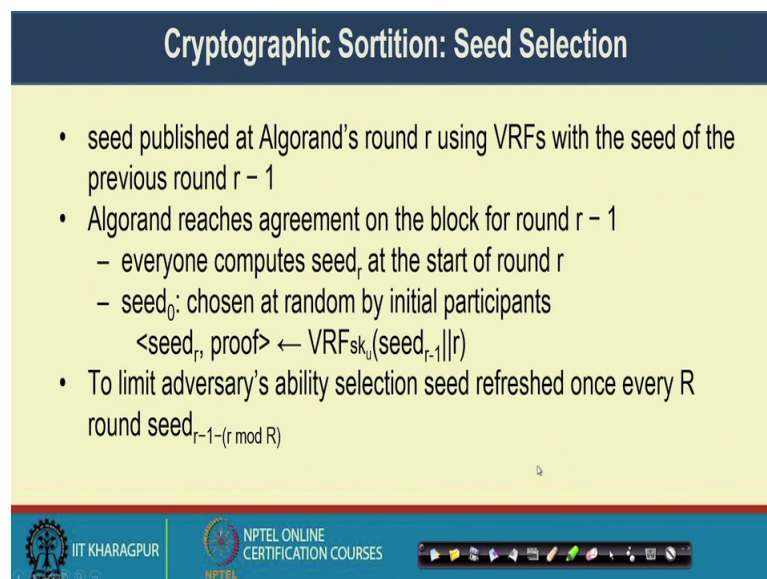


Now, every user every other user in the pair, they can verify the sortition procedure with the help of the public key  $p$   $k$ . So, you can find out that whether this particular user has actually won the lottery and got the power to be a member of the committee or got the

power to propose a new block. So, you can check if the user was selected. So, this particular function, it returns the number of selected sub users. This number actually determines that what would be the number of members in the committee and interestingly at different round; they are a different number of members who can be part of the committee.

And this number actually determines that well. These are the members of the committee and if these are the members of the committee, say if you are getting there are  $j$  number of members in the committee. Then, how many force you are expecting to prove that a particular proposed block is a valid block? So, if you are getting that many number of force based on the byzantine agreement protocol, then you are sure that well, this particular block has get is getting added to the existing blockchain and you can add that block with the blockchain and this function verify sort, it will return 0 if the user was not selected at all.

(Refer Slide Time: 25:33)



**Cryptographic Sortition: Seed Selection**

- seed published at Algorand's round  $r$  using VRFs with the seed of the previous round  $r - 1$
- Algorand reaches agreement on the block for round  $r - 1$ 
  - everyone computes  $seed_r$  at the start of round  $r$
  - $seed_0$ : chosen at random by initial participants
  - $\langle seed_r, proof \rangle \leftarrow VRF_{sk_i}(seed_{r-1} || r)$
- To limit adversary's ability selection seed refreshed once every  $R$  round  $seed_{r-1-(r \bmod R)}$

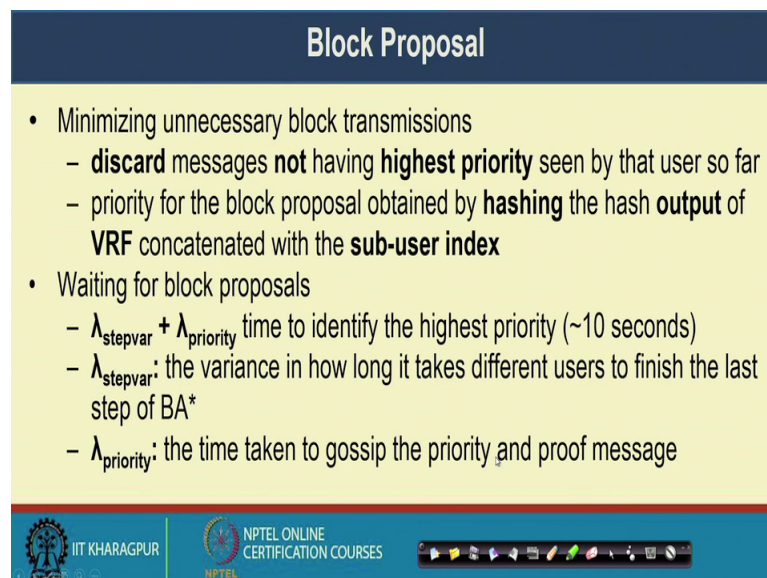
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NPTEL

Well. So, the seed selection procedure in cryptography sortition is something like this and this seed selection procedure ensures that well at different rounds different set of members will be there for validating the block in the committee and this particular round number determines who will be the members of that committee and remember that at different rounds different number of members can be there in the committee. This seed it

is published at Algorand's round  $r$  using this verifiable random function with the seed of the previous round  $r - 1$ .

So, Algorand reaches agreement on the block for round  $r - 1$ . Once everyone computes the seed at the start of round  $r$ ; so, once round  $r - 1$  is over, then everyone starts computing the seed value for round  $r$ . Now the seed  $0$ , it is chosen at random by initial participant with the help of a verifiable random function. So the, we are using this verifiable random function for the seed generation itself and to limit the adversary's ability selection of the seed value. So, the seed value is refreshed at every  $R$  round capital  $R$  round ok.

(Refer Slide Time: 27:02)



**Block Proposal**

- Minimizing unnecessary block transmissions
  - **discard** messages **not** having **highest priority** seen by that user so far
  - priority for the block proposal obtained by **hashing** the hash **output** of **VRF** concatenated with the **sub-user index**
- Waiting for block proposals
  - $\lambda_{\text{stepvar}} + \lambda_{\text{priority}}$  time to identify the highest priority (~10 seconds)
  - $\lambda_{\text{stepvar}}$ : the variance in how long it takes different users to finish the last step of BA\*
  - $\lambda_{\text{priority}}$ : the time taken to gossip the priority and proof message

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NPTEL

So, fine; next lambda block proposals. So, once you are elected as a node for proposing a new block, now there is the catcher that because every member is electing himself or herself for proposing a new block. At a instance of time or at a single round it may be possible that there are multiple users who are proposing a new block. So, if there are multiple users who are proposing a new block, then we require this agreement protocol to reach into a (Refer Time: 27:39) that which block to accept for. Now, to minimise the unnecessary block transmission Algorand apply a discard the messages not having the highest priority seen by that user so far.

So, we define a priority mechanism based on this hashing of the hash output of the VRF and it is concatenated with the sub user index. So, that hash value and the sub user index



together it provides a priority value and the user who has the highest priority that block is being propagated further in the network. So, what you are doing if you have elected yourself and if you have won the current round for the proposing of a proposal of a new block, then you wait for certain amount of time to see that whether any other block is coming or not.

So, you wait for this much number of time which is  $\lambda$  step variable plus  $\lambda$  priority; where,  $\lambda$  step variable is the variance in how long it takes different users to finish the last step of the byzantine agreement and  $\lambda$  priority the time taken to gossip the priority and the proof message and the total time is approximately ten seconds. So, you wait for approximately 10 seconds to see whether a new block is coming or not. If a new block is coming, then you see whether your block or your priority is more than the priority of the current block or not.

If your priorities more than the priority of the current block, then you propagate you a new block; otherwise you do not propagate the block any further. So, that way you can minimise the unnecessary block transmission in the network.

(Refer Slide Time: 29:31)

The slide is titled "BA\*" and contains the following text:

- **Two phase:**
  - reduces the problem of agreeing on a block to agreement on one of two options - *final consensus* or *tentative consensus*

The slide features two groups of stylized human icons. The left group consists of five icons in blue, orange, green, red, and purple. The right group consists of five icons in black, blue, orange, green, and red. The slide also includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES at the bottom, along with a navigation bar.

Then, once the block transmission is done then we run the byzantine agreement protocol. So, this byzantine agreement protocol it works in 2 phases; the first phase in the first phase, it reduces the problem of agreeing on a block to agreement on one of the two

options. So, in Algorand mechanism, we can have either a final consensus or we can have a tentative consensus.

Ah Today, will stop here up to this point that we have selected or we have proposed a new block. So, we have elected a member among the set of users for the proposal of a new block. So, say the block got propagated in the network. Now there are two conditions either the block is coming from a valid user; if the block is coming from a valid user, then you have to ensure that the block is coming from a valid user and you need to reach in a agreement that this particular block can be added to the blockchain. Otherwise if multiple users are proposing the block simultaneously, then you have to show that which particular block need to be added based on the priority to checking the priority value.

So, that way this Algorand consensus, it adds up a particular block in the existing blockchain. So, this works on this byzantine agreement protocol and byzantine agreement protocol in blockchain interestingly, it provides 2 type of consensus; one is called the final consensus and other is called a tentative consensus. So, today, we will stop here and in the next class we will discuss about this concept of final consensus and tentative and we will see that how byzantine agreement protocol actually helps in reach into the consensus even in the presence of adversary in the network.

So, thank you all for attending this class today.