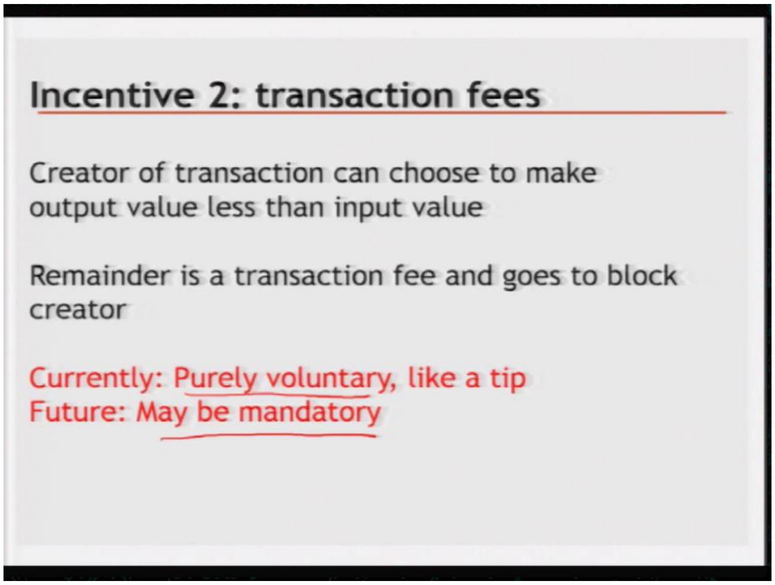
**Introduction to Blockchain Technology and Applications**
**Prof. Sandeep Shukla**
**Department of Computer Science and Engineering**
**Indian Institute of Technology-Kanpur**

**Lecture No. 07**
**Blockchain Technology and Applications**

Welcome back to this session where we would be continuing with the incentive mechanism that is used in Bitcoin blockchain. So last time, we spoke about using incentives for making the participants or the miners behave according to rules by allowing them to earn some incentive.

**(Refer Slide Time: 00:34)**



Incentive 2: transaction fees

Creator of transaction can choose to make output value less than input value

Remainder is a transaction fee and goes to block creator

Currently: Purely voluntary, like a tip
Future: May be mandatory

And we also talked about first type of incentive which is the block reward. So if your block gets selected, then you can pay yourself some very fresh Bitcoin. And now at this moment, this number of coins that you can win is 12.5 bitcoins, but it started with 50 bitcoins and then it started it became 25 bitcoins, and now it is 12.5 bitcoins so that is and bitcoin price as of now, as of today is about close to $8,000. So, therefore 12.5 bitcoins is quite a bit of money.

The second type of incentive is transaction fees. So, if you want your transaction to be put in a blockchain then you want to have you want to pay some transaction fee and more transaction fee you pay it better in sensitivities for a miner to include your transaction in the block. So, what you have to do to pay transaction is that you have to remember in bitcoin you basically say what are your inputs and what which input which transactions you are inputting from.

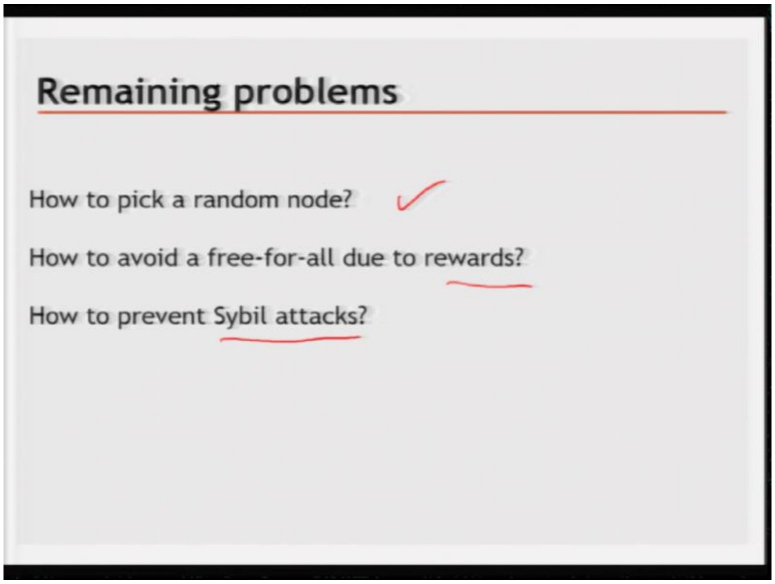
And those inputs from the, which should be from the output of that previous transaction and then you have you specify output where the money is going. So and the rule is that the total input, a total number of coins in the input should be equal to total number of coins in the output. That is how you create a change address and you if there is any left, you put it to that you add directed to that address which is under your control.

Now, if you want to pay transaction fee, what you have to do is that you make the output value that is all the different addresses where you are paying from this through this transaction should be less than the input that you are putting in. And therefore, the rest of the amount that you are not saying anything about can be collected by the miner. So every miner who creates a block will calculate from each transaction how much transaction fee it is getting.

And then it will put another transaction that gives that money to an address that is controlled by the miner. Now, not all miners will be winning. So, whoever wins that round whose block is made into the next block will actually be able to redeem that amount? So, now, it is purely voluntary like a tip, but in the future it may become mandatory because as we said before, there will be only 21 million bitcoins and by 20, 40 the rewards will stop you know will become zero.

Basically, because we are reducing the reward number of coins that we reward, you know, by half every 210,000 blocks and therefore, at that point the transaction fee should be the only incentive for somebody to include a particular transaction into the block that they mined.

**(Refer Slide Time: 03:56)**



Remaining problems

How to pick a random node? ✓

How to avoid a free-for-all due to rewards?

How to prevent Sybil attacks?

So, now we have seen the problem of malicious or rogue users or rogue miners and we have seen that incentives is one way of engineering their behavior. Therefore, they might actually be have correctly because if they did not behave correctly, they have no chance of winning the block and therefore, that they will not get any of this new words. Now, there are still more problems to be solved in the consensus.

First of all, we said that every round, every time a new block has to be added, everybody is creating their own block and each of these blocks could be different from each other. So we said that, we will actually use a raffle kind of mechanism everybody will be given some kind of a random ID and one of the ID will be chosen as a winner and that winner's block will be propagated and added to every copy of the chain.

Now, it is not as easy in a decentralized system as you might think. Because there is no central authority to assign random numbers, random IDs, and then there is no central authority to actually carry out the selection. So it has to be self-selection kind of process. B other issue is that when a block is when you choose somebody, block then that even if there was a central authority to choose that, that minor.

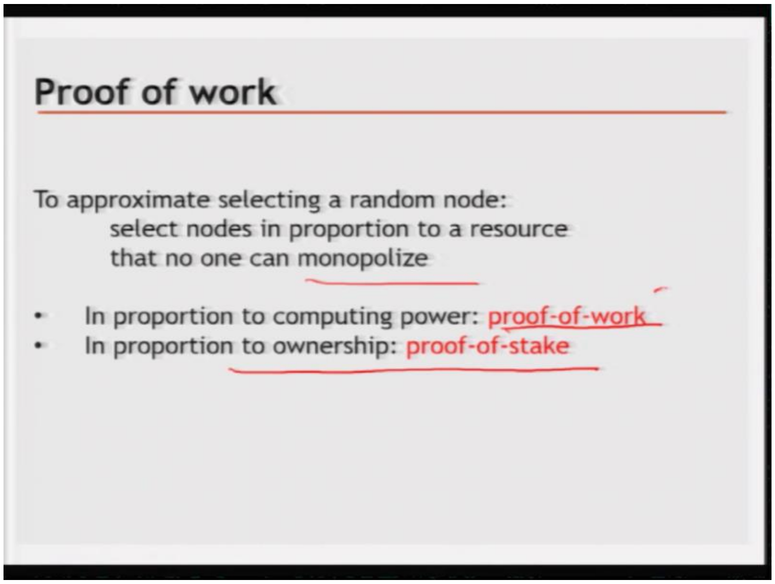
Even then the propagating that that information throughout the entire network, which is vast, would actually take time. In the meantime, some other person might actually consider himself or herself as a winner, and then start adding the block in its vicinity, and therefore, there may be a race condition. That is so, we will see how that problem is solved. Second is that you know, if everybody tries to mind, then there will be a huge.

You know, number of races and how to resolve those races in order to avoid such a situation. And then there is this issue of the Sybil attack that if somebody creates many, IDs, then he might feel that his probability of being selected this way would be more. So, therefore, we have to do some mechanism by which having more ideas does not help you to have a higher probability of winning the race.

Let us see how the random were node is selected. So, and this mechanism in bitcoin particularly or in also in aetherium is proof of work. So what you do is that all these different nodes have different resources at their disposal. So computational resources, memory, the GPUs and spatial specialized E6 hardware, specialized FPGA and all that stuff, and then they accordingly they have invested and also as I said before that to be keeping that interest.

You know large scale computing infrastructure running, they have to pay for electricity they have to pay for cooling. So, you have to kind of decide who wins based on the random node will be selected based on the proportion of the computing power that a node has so, therefore, if a node has 30% of the computing power, then we should have a 30% chance of being selected as the winner.

**(Refer Slide Time: 07:55)**



## Proof of work

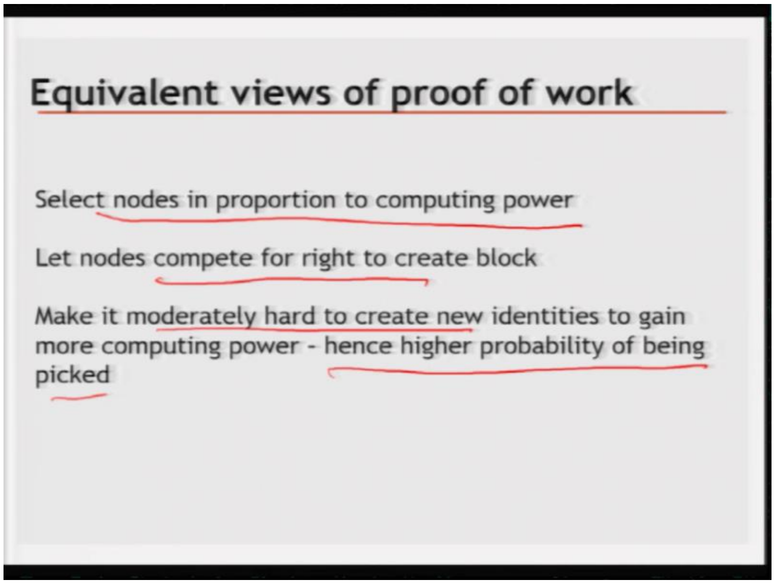
To approximate selecting a random node:
select nodes in proportion to a resource
that no one can monopolize

- In proportion to computing power: proof-of-work
- In proportion to ownership: proof-of-stake

So, let us see, and this is called how do you know the how much resources they have so, you have to ask them to solve some puzzle. And that is where the hash puzzle comes in. And then whoever does it, whoever has more computing power will have will do it faster. And therefore time taken to solve the puzzle kind of indicates how much computational resources they are putting in and therefore, it is called a proof of work.

So proof of work is proving how much computational resources they have there is also other possibilities, for example, aetherium now also has proof of stake, proof of stake is in proportion to ownership. So how much you already have acquired that according to that your proportion of being selected is decided. But in Bitcoin, it is all only proof of work. So we will know now focus on the proof of work.

**(Refer Slide Time: 08:45)**



## Equivalent views of proof of work

Select nodes in proportion to computing power

Let nodes compete for right to create block

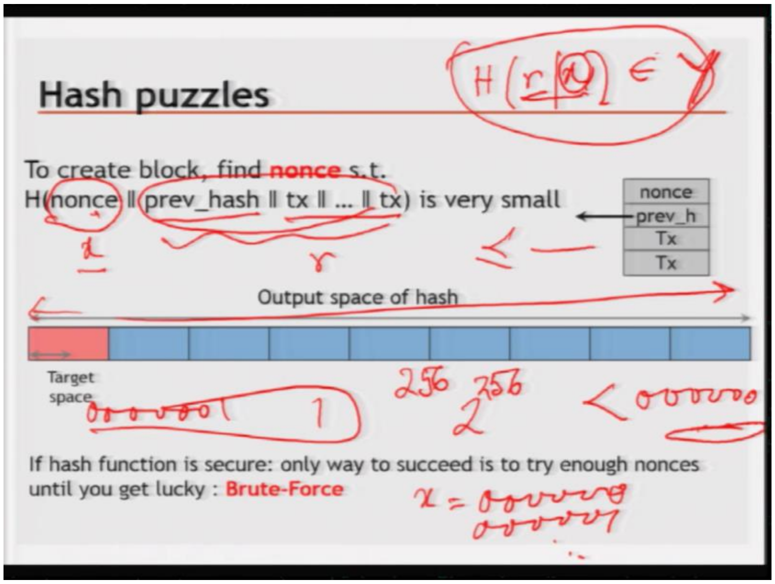
Make it moderately hard to create new identities to gain more computing power - hence higher probability of being picked

So what is happening with proof of work? So you can, as I said that proof of work basically allows you to select nodes according to the proportion of computing power they have for the inside the entire infrastructure. So to prove that they have that computing power they have to compete for to create the block and it should be moderately hard to create new identities to gain more computing power. So, hence higher probability of being picked.

So, by creating more identities will not increase your computing power because you have a fixed amount of computational resources. So, you have you can create 1000s of ID’s but then that computational resources will be used by all of them and therefore, total computational resources you have will be the same so, that basically, focuses us on this on avoiding the Sybil attack. So, so, if you have invested so much that you own the 30% of the total computational resources in the entire network.

Then dividing yourself into 300 different identities will not change that from the 30% to you know, 300% or something. So therefore, you have no way to increase your computational power. And in that sense your probability of being picked as the next block. So that is, how it is done. So let us see how it is really done.

**(Refer Slide Time: 10:19)**



Hash puzzles

To create block, find nonce s.t.
H(nonce ‖ prev_hash ‖ tx ‖ … ‖ tx) is very small

If hash function is secure: only way to succeed is to try enough nonces until you get lucky : **Brute-Force**

So, we talked about hash puzzles, so remember hash puzzle was, we said that if you have a random number, and you challenge them to give me give you an x, so that that random number concatenated with x will have the number will belong to a set y. So y is the target set. And then you have to find an x given R, what is the, hash such that it belongs to y. So that is what it is what is done here. So, you take the block, so you take the previous hash and all the transactions in that block, and then you add x.

So this is your x. And this is your r basically in the hash puzzle set, formal setup. So you have to find an x such that with this r, you will be able to do h on x concatenate with r less than or equal to some number. And that means that is a set of possible hash values. So if this is the output space of the hash values, let us say you are using 256 bit hash values, then your output spaces 2 to the 256. So that is very large this from here to here.

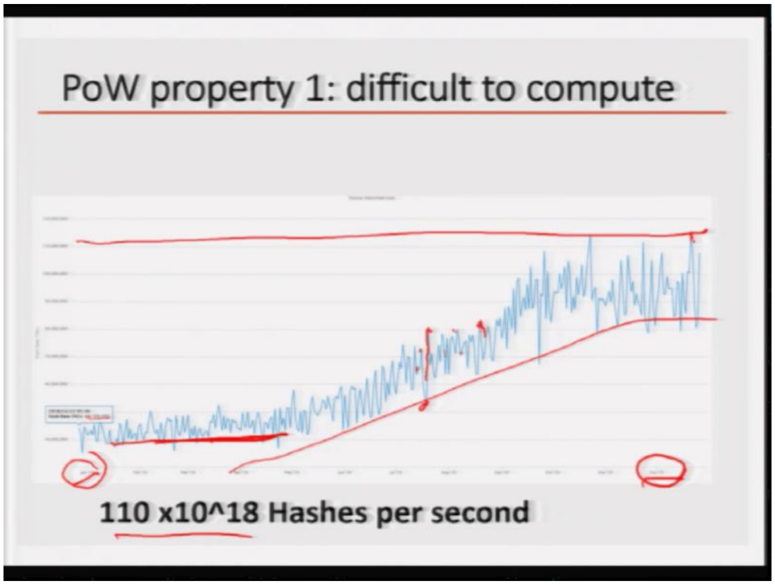
But you say that, if you will be winning, if you are the when you come when you choose an X such that, that your hash value is basically less than, let us say, so many leading zeros followed by any numbers. So which means; that you are saying that not any number from 2 to the 256 possible numbers can be selected. Only the ones with so many leading zeros, which mean relatively smaller numbers, any x that will satisfy this will be waiting for the hash puzzle.

Now, many miners let us say there are 3 miners each having 30% of the resources. So there is a probability that all three will solve them almost at the same time. And then when they solve it, they think that they have won. Because, there is no central authority to say that you are the winner. So, all three will have their blocks as competing blocks for being added to the blockchain. So therefore, there will be a race condition.

But before we go into the race condition, you see that the hash puzzle has this property that we discussed earlier, is that the only way to solve this is through brute force. You start by saying that, 0000 x will be this. And then when you try to hash if the hash does not come out, to be in that range, then you try 000001 and so on. And then you have to, you may have to try all of them. And depending on what you have chosen here, as your list of transactions that you are putting in the block.

You might actually have an issue with respect to there; you may not even get there even if you exhaust all the possible things. So, but by the time you exhaust all possible things; somebody will hit on the right x for his choice of the block, and then he will be one of the potential winners.

**(Refer Slide Time: 13:31)**



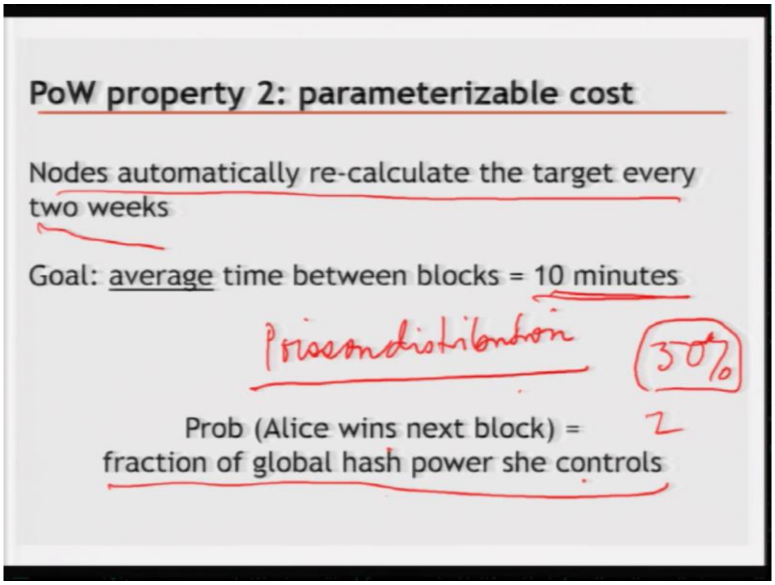
PoW property 1: difficult to compute

110 x10^18 Hashes per second

So, now everybody's competing hash so, the every second now, I looked this up and in December 2019, as you can see, that over almost like 110 million times 10 to the 12 hashes per second. So, this is Tera hashes. So, that many hashes are computed per second by the blockchain ecosystem. Now, all these hashes are not computed by single node all nodes are trying. So, this number of hashes per second is actually the total cumulative number of hashes computed.

And some of them will be doing much less number of fascists and win. So, but the point is point here is that the proof of work is a very difficult computation brute force computation and there is a lot of computation that goes on at every mining node that is trying to solve this hash puzzle.

**(Refer Slide Time: 14:30)**



PoW property 2: parameterizable cost

Nodes automatically re-calculate the target every two weeks

Goal: average time between blocks = 10 minutes

Poissondistribution

30%

2

Prob (Alice wins next block) =
fraction of global hash power she controls

Now, the problem is that in the beginning, everybody was using desktop computers and then the hash rate was much less as you can see even within like this is from January 19 to December 19. In January 19, the hash rate was 40 million terahertz, tera hashes and so if you look at this, is 44 million. Here we are looking at here we are looking at almost 110 million. So, within a span of a year the hash rate has increased.

Which means that people are throwing in a lot more resources, a lot more parallel computation, a lot more GPUs and so on. To do this hash computations and therefore, we are seeing a surge in the hash rate of the entire network. So, therefore, what will certainly happen is that in the beginning, the let us say I keep I give you a hash puzzle, and you can solve on an average within 10 minutes, then after you throw in more computational power.

You can do try parallely more many more hash combinations, many more nonce combinations, and therefore, you will be solving it faster. And then more computation you give more computational resources you give you can compute even more, you know, efficiently. So therefore, what the, what it happens is that every 2 weeks the nodes automatically recalculate the targets set. So how many leading zeros you are going to require to have in your hash to when the hash puzzle keeps increasing.

So if you see here, let us say this is the target space today, they after 2 weeks, we find that the number of hashes per second has increased. So it will be solved faster. So we will decrease the space a little bit and this will be the new space and then it will become new space. So, you make it harder and harder to reach this part of the hash output space through your computation. Now, the way it is done is that we want the average time to actually mined a block should be roughly 10 minutes.

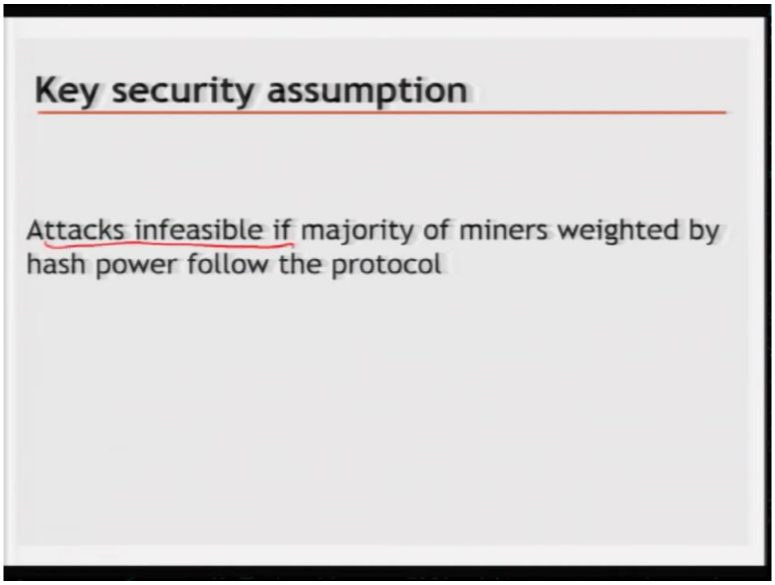
So usually this follows a positive distribution. So, arrival of new blocks follows upon the distribution. So therefore the time interval between blocks follows an exponential distribution. And this exponential distribution, we want it is mean to be 10 minutes. So that this adjustment that is done automatically. Now this adjustment is roughly in two weeks. So if you if for multiple weeks, you see that the hash rate is not going up, then you did not need to adjust it.

So you actually can see that here there was a month from March, April, the hash rate was pretty steady. Actually, it was pretty steady here. So you did not necessarily have to adjust it. However, as soon as you adjusted the time, the hash rate goes down, so this is the hash rate. So these are the points when you did adjustments. So you see this is a repeated pattern. So these are the points when hash rate went down because you made the problem harder.

But then they learn, you know how to get it faster. And then over time, the hash rate increases, then you again adjust and your hash rate goes down. So this is how the thing is dynamically adjusted. So probability that somebody wins, the next block is the fraction of the global hash power tree controls. So that is how it is designed. So if you throw in more resources, your probability is going to be higher.

So if you have 50% of the resources, then you have a chance that every 2 blocks you will be mining one block, so that is actually quite on the edge. Because at that point, you know, if you have more than 50%, then you might actually be doing more blocks than anybody else. And therefore, you can actually bypass the original chain and create more blocks and build another chain which is longer and that is the 51% issue that we talked about.

**(Refer Slide Time: 18:49)**



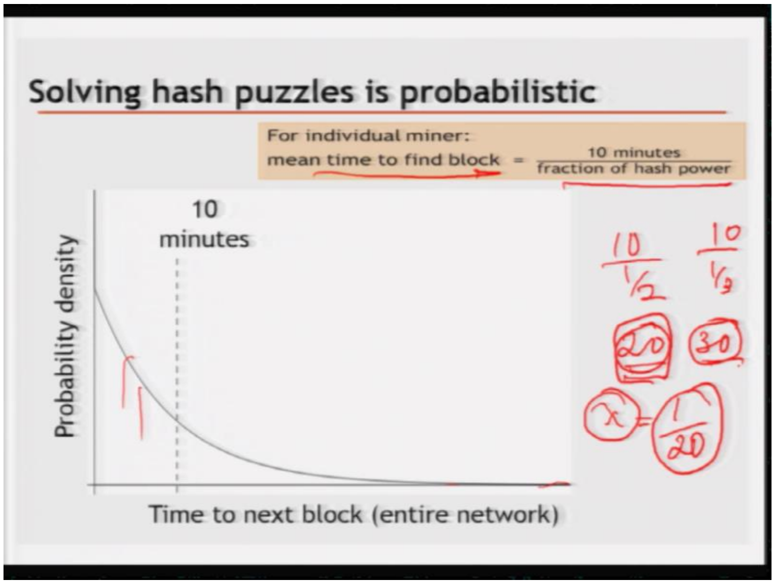
Key security assumption

Attacks infeasible if majority of miners weighted by hash power follow the protocol

So therefore they attacks or 51% attack is not possible. If majority of the miners as weighted by hash power, follow the protocol so it is the number not the number of miners that we are worried about, we are worried about the total number of hash powers they have. So if somebody has less than one person hash power, there may be, you know, you will need 51 of them to get 51%. If some, people have like 10%, and then you need about 5 to 6 of them to get 50% hash power.

So, it is not the number of miners, but how much hash power or how much computational power they have together, if they collude to become 51% then we have a problem. So, we assume that so far, it has not happened that 51% are colluding and therefore, we are we are kind of safe. So as I said that solving hash puzzles is probabilistic. So when you will solve the next one is actually follows the exponential distribution with that average of 10 minutes.

**(Refer Slide Time: 19:56)**



Solving hash puzzles is probabilistic

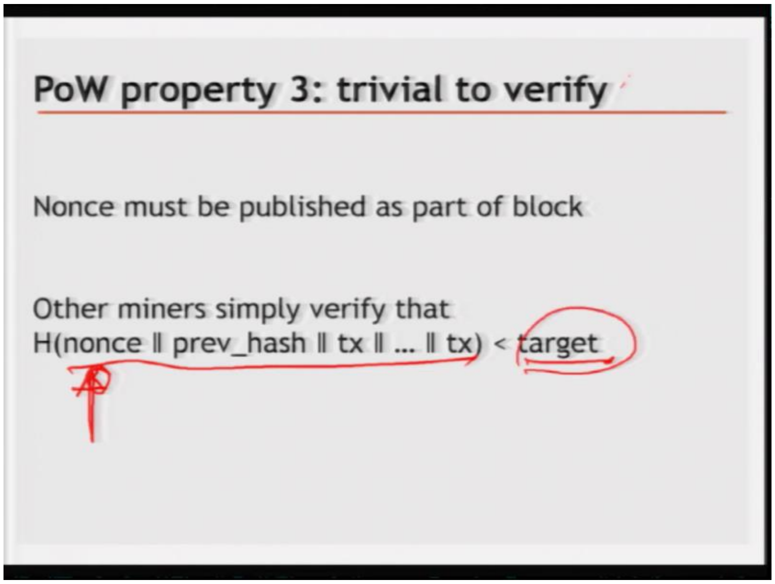
For individual miner:
mean time to find block = $\frac{\text{10 minutes}}{\text{fraction of hash power}}$

But sometimes you may need a lot more time and sometimes you can do we went faster than 10 minutes. So if it so happens that it becomes faster than 10 minutes for quite a few blocks, then we say that, the hash puzzle has become easier if people have thrown in a lot more resources. So let us go and adjust it so that it becomes again, average becomes 10 minutes. That is the idea. And for each individual miner, the meantime to find a block is 10 minutes divided by the fraction of hash power.

So if you have let us say, half of the hash powers, then your time to get this is your average will be 20 minutes, if you have one third of the hash power, then you have the average will be 30 minutes,? So therefore, the probability will accordingly also be adjusted. So if your, mean of your exponential distribution is 20, then your probability of solving one in within a fixed interval is also less than if you are more than if you are going to do it in 30.

Because the exponential distribution, the average is actually reciprocal of the lambda of the distribution is reciprocal to the mean of the corresponding exponential distribution. So your probability will be better if you are doing it in 20 minutes, if your average is 20 minutes versus if your average is 30 minutes.

**(Refer Slide Time: 21:20)**



PoW property 3: trivial to verify

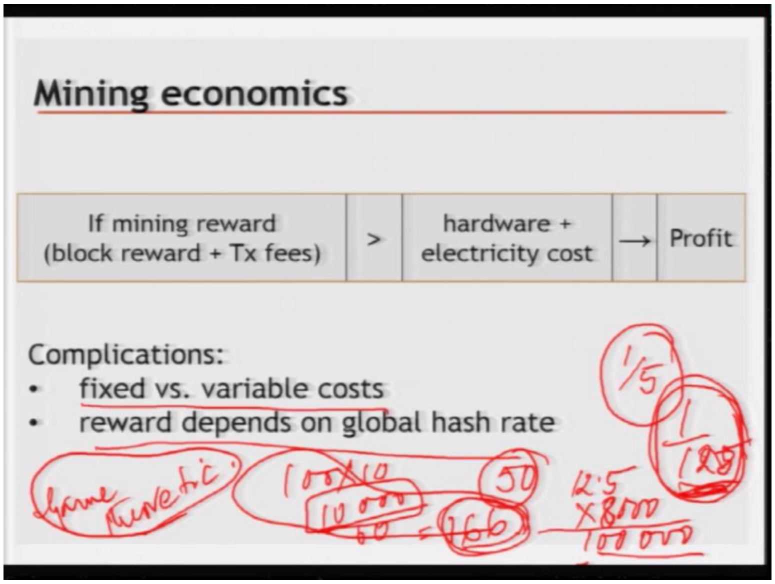
Nonce must be published as part of block

Other miners simply verify that
H(nonce ‖ prev_hash ‖ tx ‖ ... ‖ tx) < target

The other thing that must be true is that the, it must be trivial to verify when somebody claims that I have solved the hash puzzle for all other miners. If they have to verify it because before they add that block, to their copy of the blockchain, they need me to know that if it is indeed one of the winning blocks. And it should be easy because they know the target. And all they have to do is the compute one hash that is the one this nonce becomes part of the block.

So the winner has to not only broadcast his block with all the transactions and the hash of the previous block, but also has to put in the nonce in the block. So once you get that very, quickly can compute a single hash see the guy who has did the mining, he has computed probably millions of hash to in order to reach to that particular nonce. But if you are just very fine, then you have the nonce, you just plug it in. And you see whether this is below the target. And that that is why it is it is trivial to verify.

**(Refer Slide Time: 22:28)**



## Mining economics

| | | | | |
|---|---|---|---|---|
| If mining reward<br>(block reward + Tx fees) | > | hardware +<br>electricity cost | → | Profit |

Complications:

- fixed vs. variable costs
- reward depends on global hash rate

Finally, the mining economics, if I did not get enough reward, that will be my investment, that is hardware and electricity costs that are incurred throwing in so much computational resources, hardware, software, cooling costs, electricity costs, and so on. Then I will do it so I have to have enough high probability of winning, if I am throwing in a lot of resources, so there has to be a profitability in mining now. There are a couple of things in this.

That is the fixed versus variable costs. So hardware cost is fixed, but the electricity cost is variable. So you have to accordingly do some amortization level computation, you have to see out of how many subsequent blocks you have, you usually make a block. So if your probability of being one of the winners is one over 5, then every 5 blocks, you will be making 1 block. If your probability is 1 over 100, then every 100 blocks, you will be probably making 1 block. Now, 5 blocks takes about 15 minutes, 1015 minutes if you can win 12.5 bitcoins, and then each of them let us say cost  8000. So, you are talking about $100,000 in 15 minutes.

**(Refer Slide Time: 23:48)**

Putting it all together

So, if you can win $100,000 in 15 minutes, then you have to see that to obtain one fifth of the hash power, how much hardware you have to invest in how much we will be your electricity cost over a period of 50 minutes. And our cooling costs over a period of 50 minutes, you will subtract this if your probability is 1 over 100, then you can only win 100,000 in 100 times 10 that is 10,000 minutes. So 10,000 minutes is a long time.

So that is about we divided by 60. So we get 400. So, 16 and others 160, so that is a 166 hours. So if you if you take 166 hours, so you have to see whether the hardware cost and the electricity cost that you incur in 166 hours, obviously, you have a much less hardware, so much less electricity cost and much less cooling cost. So you have to see whether that makes sense the other otherwise you will not do this.

So the; reward, whether you get the reward or not, it depends on the global hash rate. It is not only your rate, so your probability let us say is one of our 100, suddenly somebody invests a lot more? Suddenly your fraction of hash power reduces. And then let us say it becomes 1 over 125. Then again, you have to recalculate all this? So it is reward is not only dependent on you, or what is the probability of you are getting a reward is not only dependent on you, but also what investment others are making.

So that becomes a game theoretic problem. And we will not discuss this in this class but there has been a lot of work on applying game theory to see whether the strategy of each miner forms a Nash equilibrium or not? If they do then it makes sense that they will stick to their strategy. Otherwise, they may not stick to the strategy and shift their strategy. So we will actually end this section here.

And then in the when we come back next session, we will just basically summarize what we are discussing with respect to the Bitcoin consensus and how the consensus works and also talk about what we are going to see in the in the future in this class.