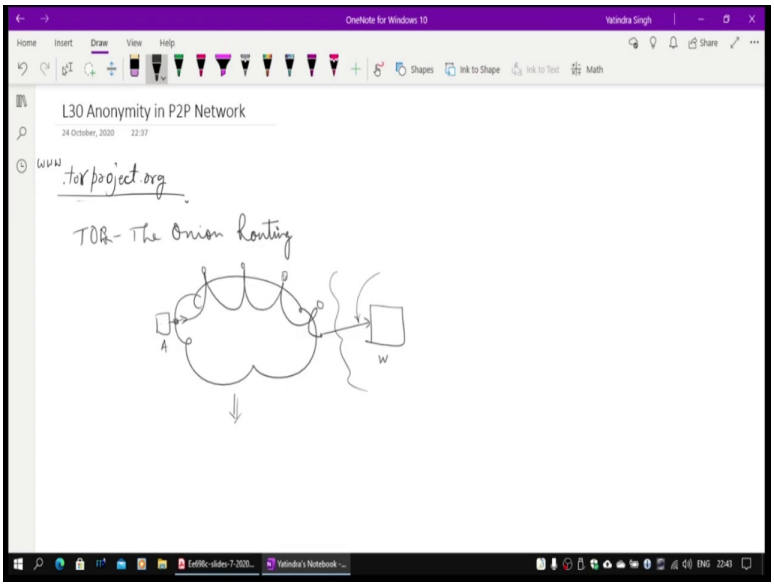
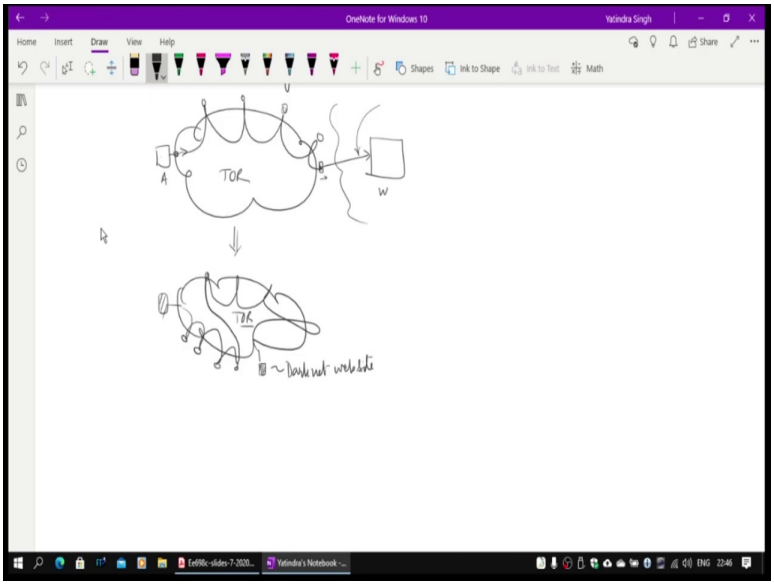
**Peer To Peer Networks**
**Professor Y.N. Singh**
**Department of Electrical Engineering**
**Indian Institute of Technology, Kanpur**
**Lecture 30**
**P2P Anonymous Communication**

Welcome to the lecture number 30, this is the last week for the mock on peer to peer network. So in this week we will be looking at anonymity, how this actually can get achieved. So, before moving on to formally to power system the onion routing system I am going to describe you one very simple thing which is very basic and this is what essentially being used in a different form when the tor routing is implemented.

(Refer Slide Time: 00:51)






But before moving on to that if the tor can be actually used by all of you, you can go to tor project dot org and you can download this particular application for yourself and use it, so this does allow what we call anonymity. The tool which is this is very popular now we call it tor browser, this is basically a Firefox built which actually has been modified to implement a tor routing.

Technically this tor routing can be implemented in any application, so both endpoints will become anonymous to everybody else. When the traffic is passing through or no intermediate nodes will be able to figure out it is almost impossible I am writing I am actually telling almost because it is technically feasible to figure out if you have access to all these intermediate node which is what makes it very difficult.

And the tor stands for the onion routing, that is a short form for this. A lot of people use it in a network if you actually are running a tor browser on your machine this normally will create use many machines in between before it exits and then it can go to the website and access the website, the website will not be able to figure out that you are accessing, who is accessing the website unless you explicitly tell him that who you are.

So, if A tells the website that I am A then only the W will know that it is A, but A knows where is W, so this is a normal internet actually, but this is what is a tor system. So, there are, they are using intermediate nodes which are set up by volunteers all across the world, so your request is getting hopped through this, none of these guys will be able to identify what you are happening,

so there is going to be heavy encryption which is being used and the only where we can somebody can figure out either if you are actually knowing this particular traffic and when you are transacting in this can be crack you can a W will be knowing who the A is.

There we can figure out, but by figuring out what is the traffic exiting from your node nobody can figure out where you are communicate. This was of it is a very powerful tool and in fact there is an extension to this we will be discussing even that at some point of time. So, where not only you actually can actually you are in this tor network, they are services which are also in the tor network.

These services you will be able to access through splicing these two kind of relay connections, so you will not be knowing where this guy is where the server is running and you can get the service, these are what we call Darknet websites, so that is a popular name. So, you not do know that where these websites are there, Facebook also runs at Darknet website so you can actually access your Facebook from there.

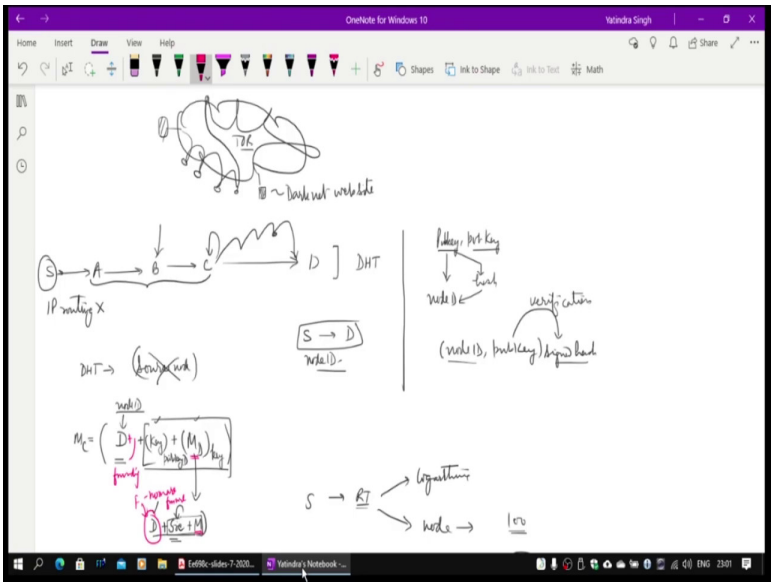
If you go and search for the Darknet URL for the Facebook I will tell you where what is that particular URL and you can try that and access it. This is a very nice tool and it can ofcourse in this browser you can see that what are these relay nodes which are being used. Now, these all set of nodes they actually all form peer-to-peer networks so you can always decipher where these nodes are and they always remain connected as overly network, they all use similar any one of the DHT routings to maintain the network connectivity.

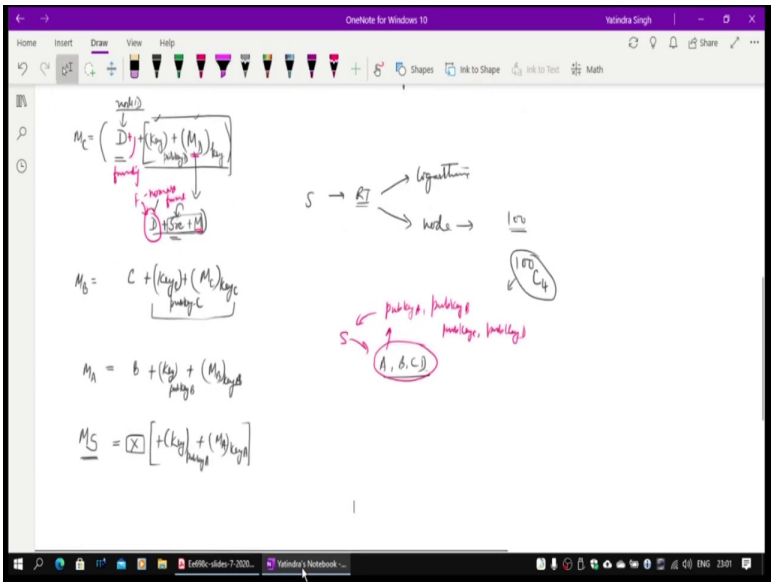
Now, but before going into these details of tor which I will be doing in the next few videos. So, let us look at the one of the most basic systems. Here what I want is now remember these then I am talking about this tor systems I am talking about TCP connections transport control protocol these are not packet based system they are likely directly TCP messages.

Though of course you can use packets in between for relaying, but this end-to-end is a TCP connection, so whatever your byte you are pumping in these bytes are going out here, you do not have to attach any message or any overhead or anything but technically as this are actually messages which are been transported among this but the endpoints, this exit point from the tor network this is what we called tor network.

And these are called tor relays and this is the exit node, so from there it goes to the internet and accesses the website. So, and when this Darknet websites and everything they are actually all part of tor network it is very difficult in fact in some of the countries this is actually illegal to use. But it is very difficult to crack, very difficult to track, this particular situation. So and now first of all describing how a packet switching system actually can be filled in this fashion, so I am basically giving a scheme we call it a telescopic encryption mechanism that is what essentially I want to explain here.

(Refer Slide Time: 06:22)




A wants to talk to some node, so we call it, S actually want to talk to B, let me have this. We also have nodes, A, B, C and D, we have 4 nodes so the messages can go through this way and ultimately to D. Now, we are not going to use IP based routing, so no IP routing, we will be using DHT routing for all messages to be transported, so it is a DHT message which will be transported.

In DHT message you do not require a source node to be there, so you can send the message without source node, so no source node. So, there is a reason why I am actually doing it, ofcourse it is possible but there is no source node to be done. When, a packet goes from source to

A. A will only know there is some destination node, A will go to B and B will go to C and C will go to D.

Now, the problem here is what I want is A and D can know, S and D can know each other, source and destination, they can know each other's node ID. But this intermediate nodes will not be knowing which are the two people who are communicating they may know the other person but they may not know the both of them at the same time, in fact B for example will never be able to figure out who is communicating to whom, it only at this may know that only A is communicating.

But A also it will not be able to figure out in the current scheme, because source node entry is not being maintained in the packet which is being transmitted. So and D will be the final goal where the message has to go. Now, let us see what the message is for C how it will look like, so there is a message which I want to send to D I call it message to D. So, now this message S has created this message which has to go to D, so what it will do?

Now, somehow C should know that we are is a destination D then only this can send the message, so I will create this particular message, but then this message is for D and most likely I can actually encrypt it with the key, so I want to do it that way. Now, one more important thing that I am actually taking a case where public key and private key pairs are there with every node and public key is used as a node ID or maybe hash value of this, this is also possible that you can compute the hash and that becomes a node ID.

But then you still maintain the public key. Any node ID will be going to have a node ID public key and ofcourse the signed hash, so you can always verify this sign hash with this public key, this verification is possible and because you can verify the public key with the signed hash, so you know this node ID is actually randomly generated it is not any arbitrary D thing. And of course this also serves another purpose to provide anonymity or security.

I will take some key and we will actually encrypt the message for D by keys. Let me write it in a slightly different way, I have $M_{D.}$ this is going to be encrypted with keys that is why I am showing with this. Now, what I am going to add here is I need to tell. D knows that it is being encrypted.  key will being encrypted key has to be added and this will be encrypted with the public key of B.

So, key is still is not been mentioned here, so D know that is going to come with me and it has to be told that there is no further destination to whom you have to send the packet, I have to also tell this guy that when D receives this packet he gets this key he gets the message, but he should also tell whether it has to be for whom. So, there should be at node ID of this is for who is the next guy to whom it should be given.

So, once I tell you are the person to whom it is, when it knows that time is not forward it is actually for me, I can decrypt the key and then get the message. So, it will now put the node ID, so I call it node ID is D, this is the node ID for this particular node, so that is what is going to be put this is the whole message. Then the message should contain that to whom it has to go back, so when it is original guy when you tell now this will be encrypted in this particular message, this will consist of message plus the source to whom you have to respond.

Nobody else will be knowing who is the source unless it goes to the D, D will only will decrypt everything and we will know it, this information will tell that it do not have it is not to be forwarded to anybody. Now, this is the message which should come out and should become available to C somehow. This is going to be the message which has to be there with C.

Now this message when will be received by C, so C has to receive this message so C somehow should actually know now this whole message to whom it has to be forwarded this message D also should so have source plus it should also have a destination also should be incorporated in $M_D$, let me just put it here.

So, I should also have a flag whether is forwarding or not forwarding I can do it this way or I can actually put a node address this a destination, this is the source, this is incorporate, so when D will be received, so it will be decrypting everything $M_D$ will be available for D, From that it will figure out that this is was generated by this has to be forward to me this is actually my message and I have already successfully decrypted this one.

This a source to whom I need to respond, so this actually is becoming part of the message itself. Once I get this I am now here also adding this node ID D, so this is the message which should be sent to C, When C will receive this, C will figure out what is the node ID, to whom I have to forward it, this it cannot do anything it will simply now forward this this particular part this

particular part it will just simply forward to the node D, this is all the anyway encrypted, so nothing will be happening only D can decrypt it.

From C it will now go through DHT hops and ultimately reach to D, so D will only know that it has come from a node which was the previous hop, it will not be knowing about C, because there is no source address anywhere except in the end where it has been put here so that D can respond back to the source.

Now, D will only be knowing at best this is the previous hop from here it comes, so it has to have access to everybody to move actually from where the massage actually has arrived. So, once it knows C and that is it, until C everything is encrypted actually now this MC also has to be further encrypted actually. So, the way it will be done is this $M_C$ will be taken, the $M_C$ is a decrypted message, so $M_C$ will also be encrypted by some key, so I can call it key C and key C will be present and this will be encrypted with the public key of C, so only C can decrypt this one nothing else.

So, this is the message which is coming encrypted and so $M_C$ will be done this way this is now what I am creating is $M_B$ now here, M plus because it has to go to C the node ID, so this is the message which will be coming to B. So, when it comes to B, B will figure out to whom it has to go, it cannot decrypt all this stuff, but it has decrypted the message which has come.

So, for example here when it is received by C this whole message $M_B$ will only send this much to the C actually. C will figure out will decrypt using this this particular key will be decrypted first using the private key and this key will then be used to figure out the $M_C$ and $M_C$ contains where the message has to be forwarded now which is D and what is the message which has to be forwarded, which is further encrypted, so there is a multiple layers of encryption which are happening.

Similarly now going further, so you will add now $M_B$ here which will be encrypted with when it will be coming it will be encrypted with, key which is so only B can decrypt this particular message that is why it is encrypted to key B, so there is this key will be encrypted with the public key of B, only B has the private key, so only this guy can actually get this $M_B$ message nobody else.

And when I add B here this is the message which will be received by A and of course when you look at $M_A$ this will be further encrypted by another some random key and now this whole exercise. I will explain where it will be done, I will add key here which will be encrypted by public key of A, so only A can decrypt using the private key this particular key and which can be used to get the message A.

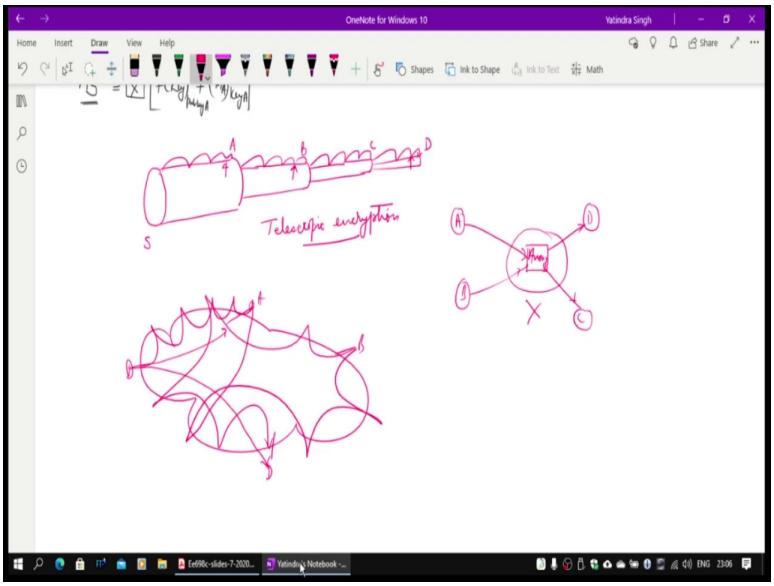
Now, this message is what is going to be generated, if this message has to be sent by whom, so you do not require anybody here. S already know that I have to give it to the A, so this is what the $M_S$ the node which is transmitting the source here, this S will generate this particular message. Important thing which you have to note down that here I do not need any other information this is not required, because S anyway knows. S knows that which are the A, B, C and D nodes and the last guy D knows about the source that is very important.

So, the procedure here is very simple that A will now the source will choose it will have a routing table and routing table is maintained using logarithmic partition again, it can have even other node list also it can be maintain, so there if they are even 100 nodes and it has to choose these four guys which are very different you will get the way the sequence you can actually do four combinatorial, so ${}^{100}C_4$ possible combinations exists.

It can choose any combination of A, B, C,D and S will then start creating this particular thing it knows what message has to go, so to pick up the message which was to be communicated to B, which will be now this message which has to go to D. So, this message is going to be this it will add its own node ID, then add D so the D knows that this is the next hop, to whom it has to or is basically going to be a flag it can simply be a flag, in no more forwarding.

And when here there will be again there will be a forwarding of flag will be there, so D can be there and D can pick it up when D is there so there is a forwarding flag ,it will forward it to D. D will be staying here. Now S will start creating this procedure this from this message onward it will keep on creating first player to some random key encrypted with the D, because now once S knows all these guys, S also knows what is the public key of A, public key of B, because it knows all the node IDs, S knows all these, S will use these actually one after another and will create encryption layer.

(Refer Slide Time: 19:45)




Now the way it looks like you are going to have a encryption layer from source, but this is going through multiple hops, so by the time it reaches A. A will not be knowing that where is S because S is not listed it is just going through multiple hops, so at best A will you knowing only this previous hop and here A will be now decrypting or removing one layer of encryption and you will have another layer it will go till B.

B again this will be going through multiple hops. A only know that it has to go B it will not it is not aware that it has to go to D actually. So, B will only know the previous hop it is not moving to know that it has come from A, the B can decrypt it, so it will there is another encryption layer gone and again C will be there again it has gone through multiple hops, another layer will be removed and ultimately it will be D, only D knows it has come from S, because that S is been mentioned in the message.

If you do not mention it, D will not be knowing that it is come from S only thing a message has come, it has to make sense out of it actually it only knows about this previous hop from where the message has come. This is what we call telescopic encryption which is used this is actually very safe because of multiple layers of encryptions which are used. Now, this is kind of anonymity because no intermediary will be knowing and for every message you can choose a different intermediate nodes. If your destination is fixed and remaining under nodes are there

with you, so 100 C3 combination, which itself is going to be still very largest not a small number.

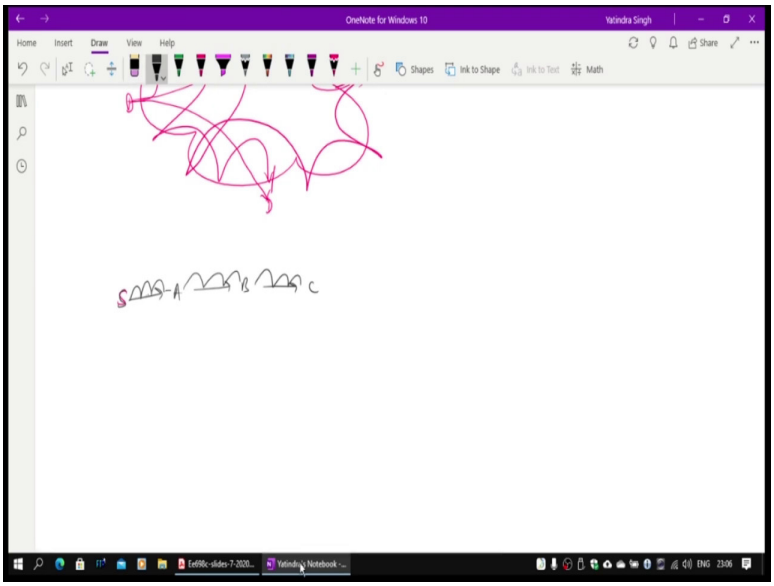
And keep on changing it every time only problem is that if you are in a network what is going to happen is you transmit if this is your A, this is your B and this is your D, normally if you remember it would have been very efficient going directly here, but now the way it is happening is you are not even going directly to A now here you are actually going through DHT hops like this.

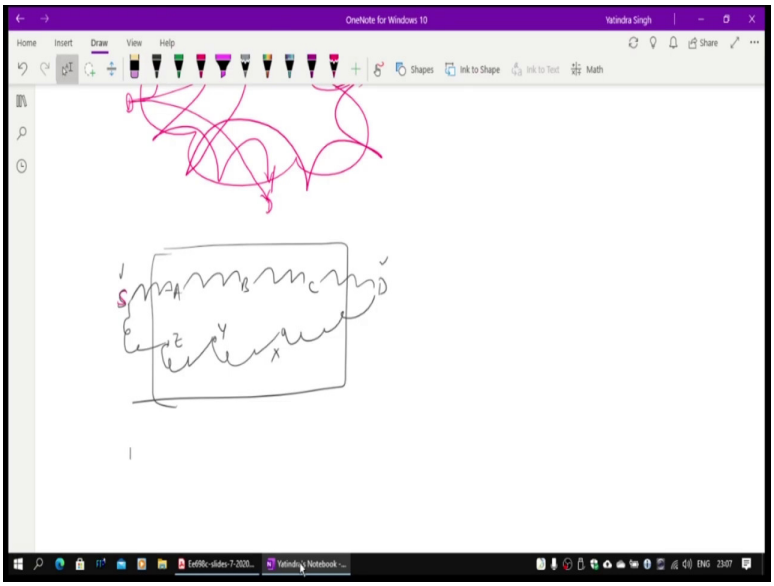
So, you will never know you are going which hops and you are reaching to B and then you are reaching D, so you are actually using creating a mishmash essentially is becomes end of extremely inefficient, but that is a penalty which you have to pay when you want to have anonymity. Another way people thought in the beginning anonymity can be done is, was using anonymizer, so if there is only one server which is there and then there are nodes, somebody wants to talk to somebody else.

Then what you do is you actually then send a message to anonymizer box, the server and this server will remove the source IP address or whatever is identifier and then send it to the destination. Everybody sends a packet to anonymizer telling that to whom the packet has to go, everybody will send it to only this is like a proxy and to remove the source. They will only know the packet has come from an anonymizer.

Only problem is if this gets compromised everything gets failed, if this fails anonymous service fails, this is not what is going to happen here in this case. This is the very basic of how the anonymity can actually be built. Now, the next question arrives, how the D is going to respond back?

(Refer Slide Time: 23:30)



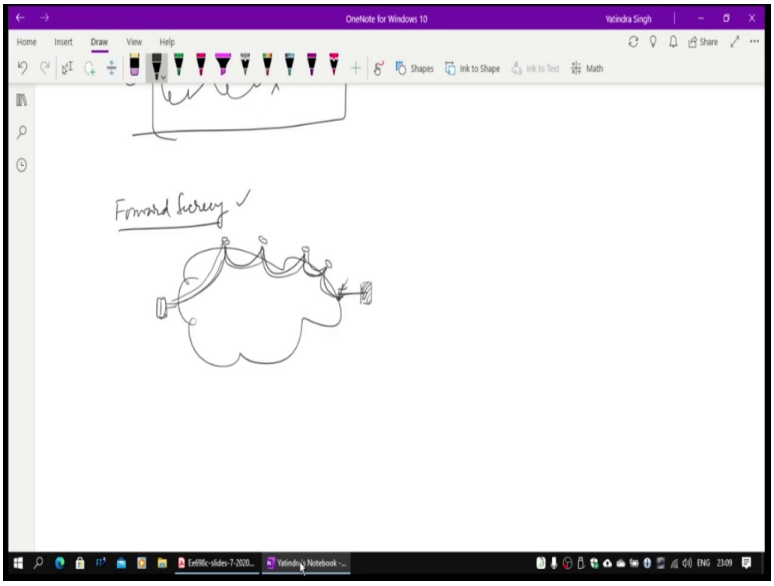




The way it is going to happen is that S is A, B and C actually it is not a direct connection I should it is a multiple hops, so that is the way I should show it, not by a straight line and removing that part it is better to show it this way, so it is better to show this way, now D knows where the source is, so it can actually now pick up another set of nodes its own maybe X, Y and Z and then the message will come.

Now, only S and D should now do have to they are the only have powers to tell in mutually do authentication. They can know each other so two preference can talk to each other but nobody can do traffic analysis on the internet and can find out who is talking to whom and your every

time can keep on changing these peers. This basically is one of the ways that anonymity can be done. So, next video will look into how the tor browser our tor routing works and it is basically trying to create what we call TCP connections. Currently what I have told you is a packet based system messages being done.

(Refer Slide Time: 25:04)




In fact now you can create variety here you have only vertical forward secrecy. When A sends to B, B will not be able to figure out it has come from A totally knows about the previous hop that is the forward secrecy. But A does know that it is going to B, now how to figure out that even forward as well as backward secrecy both can be achieved? Now, tor does a slightly different thing, in tor what we want is I am trying to create connect to a website, let us look at I am trying to connect to a website.

Now, when I am accessing a website, website knows that from where I am connecting, I do not want the website to know from there I am connecting or I am accessing the information. How to build up such a system? And it is a TCP connection, but I have to create a TCP connection through multiple relays in between and there is one node and which then on my behalf connects to the server.

This guy will always see this IP address as the person who is connecting not this particular client. The communication will still go, so none of these the traffic analysis if you are ready to cooperate and you know about the traffic you will not be able to figure out that this is

communicating to whom how these pipes are being connected, so there is a pipe which is being connected. There interesting things which can be done with this pipe, so we will discuss them in the next video.