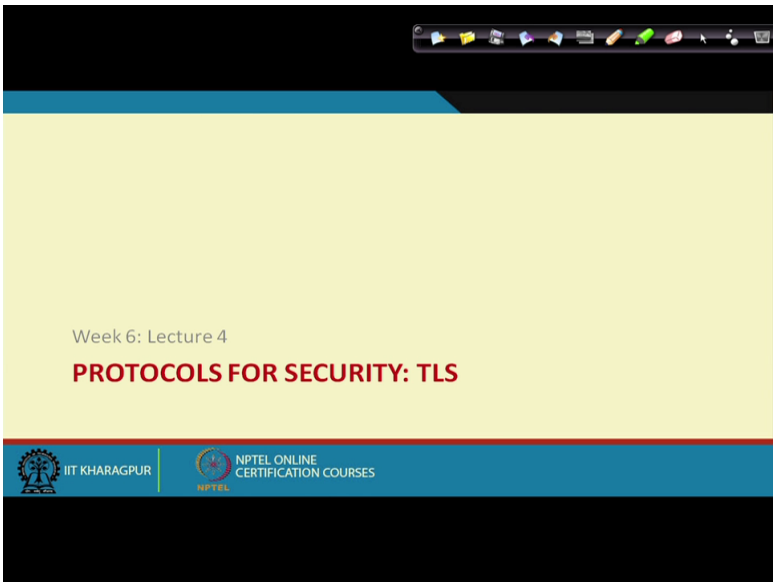
Course on E-Business
By Prof. Mamata Jenamani
Department of Industrial and Systems Engineering
Indian Institute of Technology Kharagpur
Lecture 31 Protocols for Security: TLS

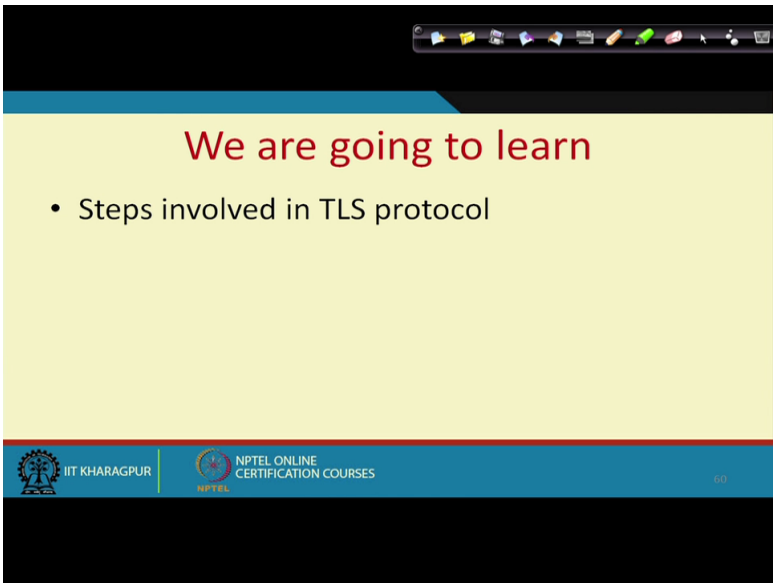
Welcome back and we continue our discussion on security protocols

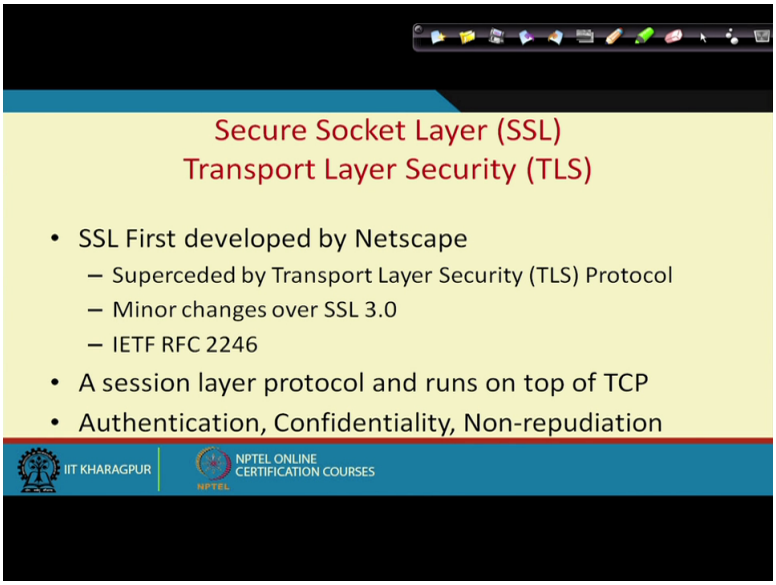
(Refer Slide Time: 0:23)



And in this regard last class we saw about what is a digital certificate and that time I told you that digital certificates are used when somebody establish a secure secure communication with a server so let us try to know about a protocol which we typically otherwise use unknowingly which is https which is a which is a protocol which is which is inserted within the TCP IP protocol stack so this protocol is called TLS protocol.

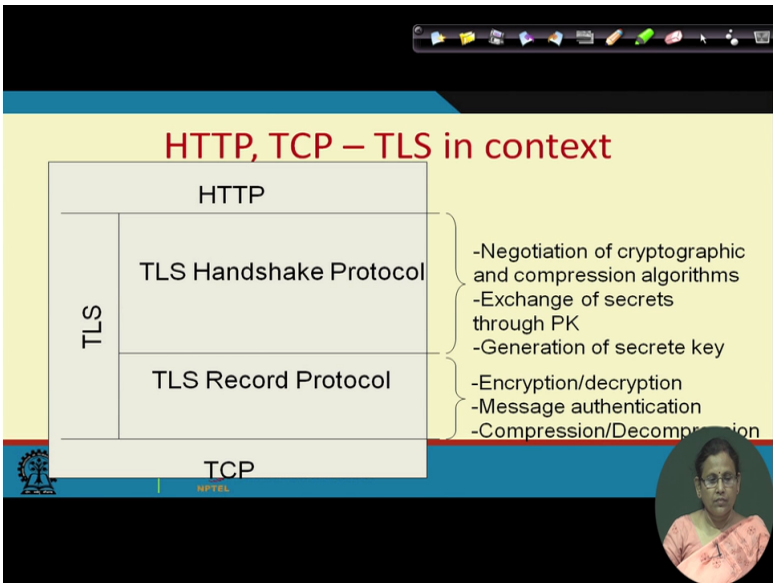
(Refer Slide Time: 1:05)





So this TLS protocol originally was known as SSL protocol first proposed by Netscape later it is it is called it is called as transport layer security protocol and used for secure communication over the web. So this protocol basically resolve this authentication confidentiality and non repudiation issues.

(Refer Slide Time: 1:36)

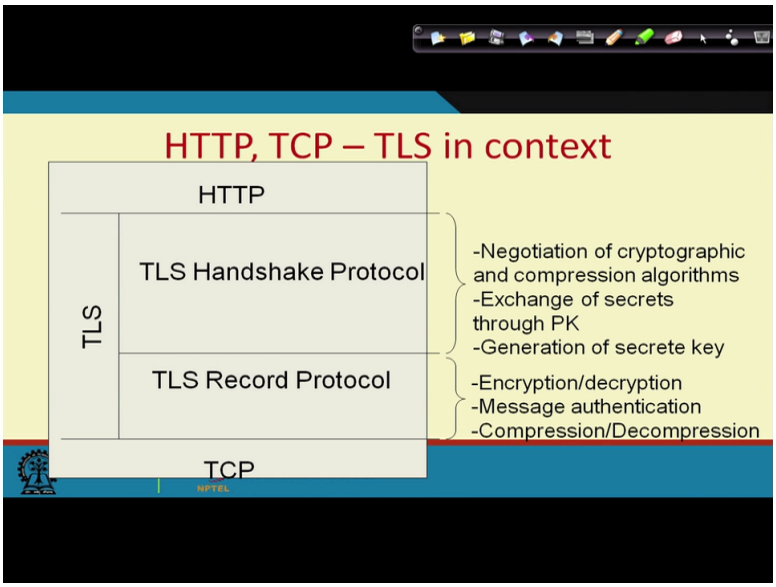


Let us see that how this protocol works? This put protocol if you remember the TCP IP Protocol stack TCP IP is at the top and above that you have application layer so above TCP and below http http is one application layer protocol so above TCP and below http lies your TLC transport layer security protocol.

This protocol has 2 parts first is TLS hand shake protocol and second one is TLS record protocol if the work of the TLS handshake protocol is to to negotiate about the Cryptographic and data compression algorithm exchange of secret public key generation of secret key and before I proceed about this record protocol.

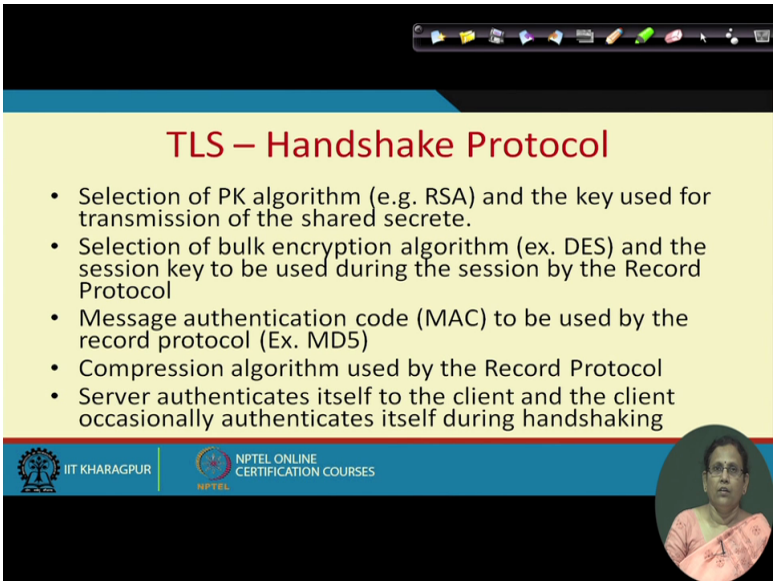
Let me tell you while discussing about symmetric and asymmetry algorithm we learnt that asymmetric algorithm are very costly compare to your symmetric algorithms but due to problem in key exchange the keys are first exchanged using asymmetric algorithm and once the keys are shared by both the (())(3:10) then the bulk transfer can take place using symmetric algorithm.

(Refer Slide Time: 3:17)



This is this idea is implemented in the TLS protocol so during TLS handshake procedure this key exchange takes place and during record protocol actually bulk data transfer takes place so let us try to see what all happened in detail what all happened during this protocol execution.

(Refer Slide Time: 3:41)

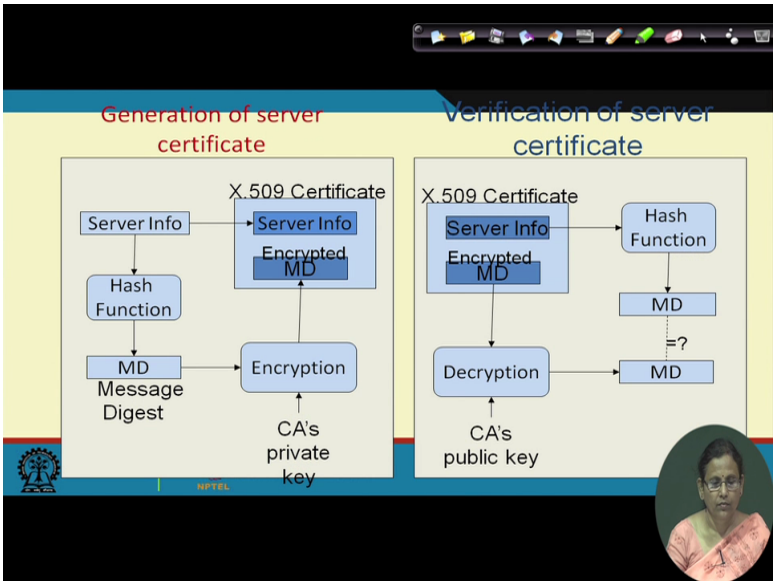


So let us first learn about TLS handshake protocol here we have already learnt about how this public and private key pairs are generated by by certificate certifying authorities and all and we are not going to discuss about it anymore I expect that you have already gone through that lecture now it is the work of the TLS handshake protocol to select the public key cryptographic algorithm.

And in this case RSA is used the key used for transmission and share of and transmission of shared secrete so the key is required here it also helps in selection bulk encryption algorithm DES and the session key to be used during the session by the record protocol then the message authentication code to be used for the record protocol then compression algorithm used by the record protocol.

And server authenticates itself to the client and the client occasionally authenticate itself to the server during handshaking procedure.

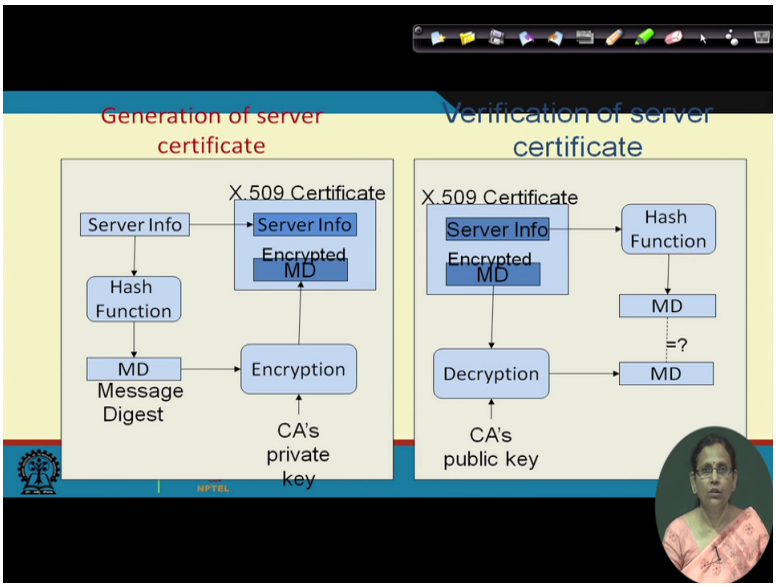
(Refer Slide Time: 5:07)



So a basically at this stage during this handshaking protocol the authentic the server authenticates itself and the client occasionally authenticate itself to the server but during this authentication process the key exchange has to take place as well as both the parties must decide see if some suppose first the key exchange has to take place so both the parties must have to have the same algorithm to decrypt the information about the key.

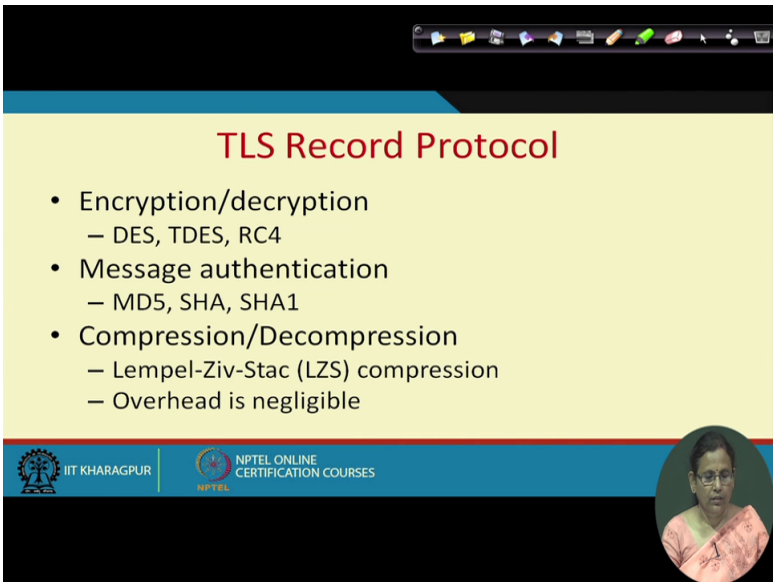
So some negotiation has to be has to happen between client and server similarly when the bulk in encryption takes place the same algorithm must be present in both server and the client where as suppose DES is present in server and client has been then if something gets encrypted with DES algorithm same DES algorithm with the shared key which is use shared using this PK using the digital you know server information exchange procedure can be actually used for decrypting the bulk data.

(Refer Slide Time: 6:41)



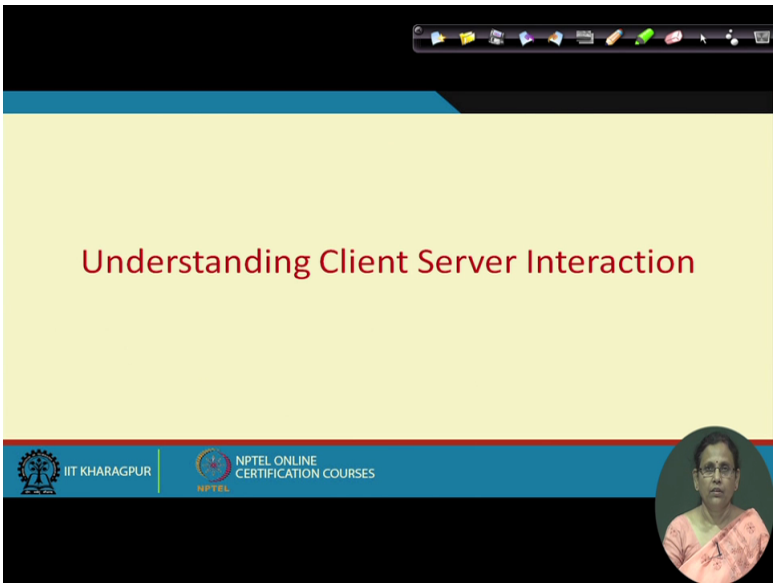
So this is about the digital certificate we already know and the server show this certificate to the client to get itself authenticated.

(Refer Slide Time: 6:50)



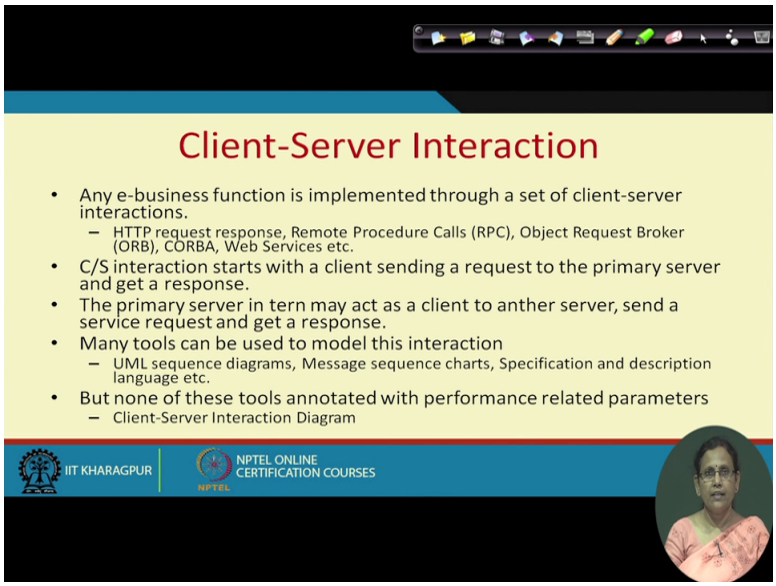
Now look at the TLS protocol details for encryption and decryption for bulk encryption and decryption it uses DES, TEDS or RC4 for message authentication MD5, SHA, SHA1 and so on similarly for data compression or decompression it uses certain algorithm as well .

(Refer Slide Time: 7:15)



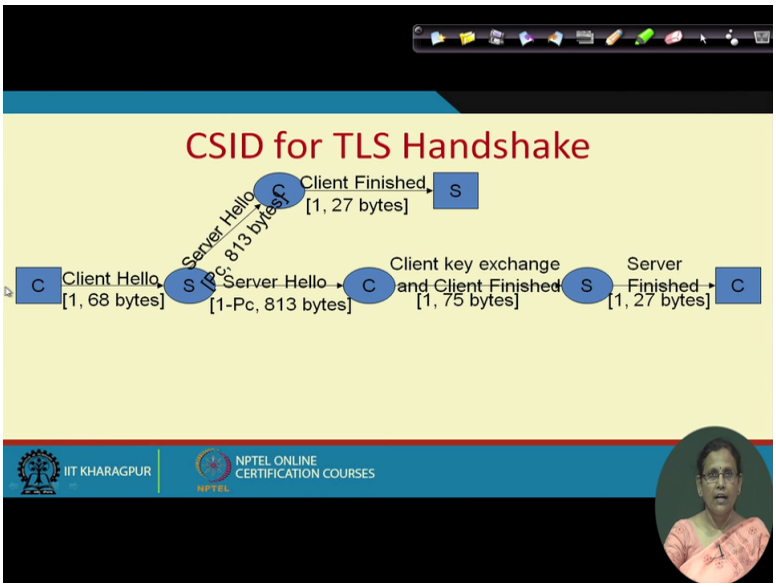
Now let us see this steps in little bit more detail using something called a client server diagram which we are going to use subsequently.

(Refer Slide Time: 7:27)



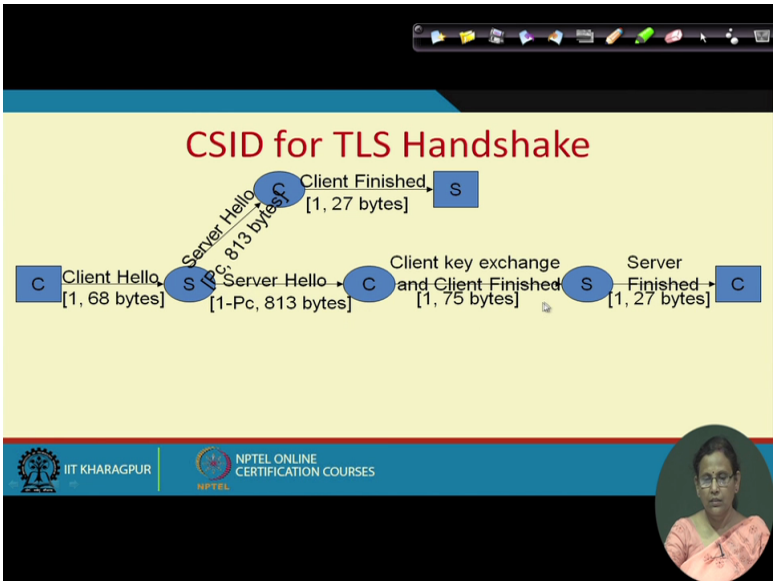
In a typical client server interaction diagram we will be we will be having two nodes one we call as a client node.

(Refer Slide Time: 7:39)



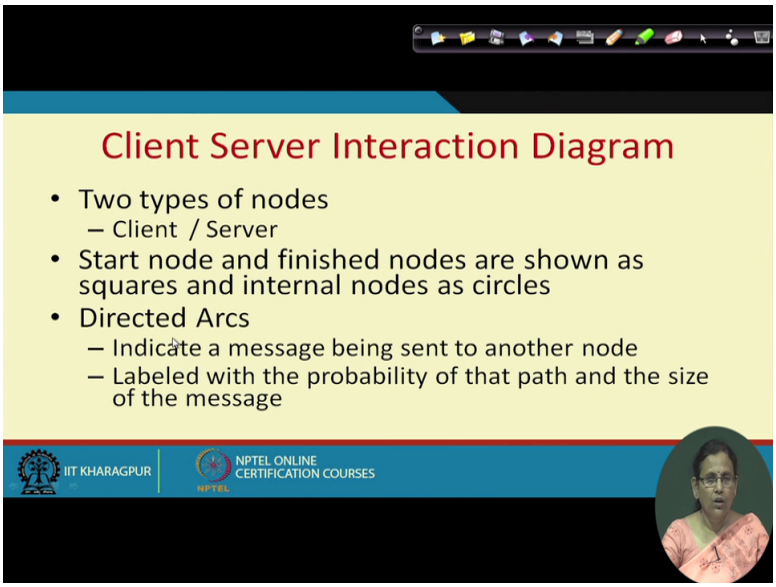
This will be calling as a client node and this will be calling calling as I mean the then the other nodes which will be used for various operations in between so you have two types of nodes one is the client node and another is the server node.

(Refer Slide Time: 7:58)



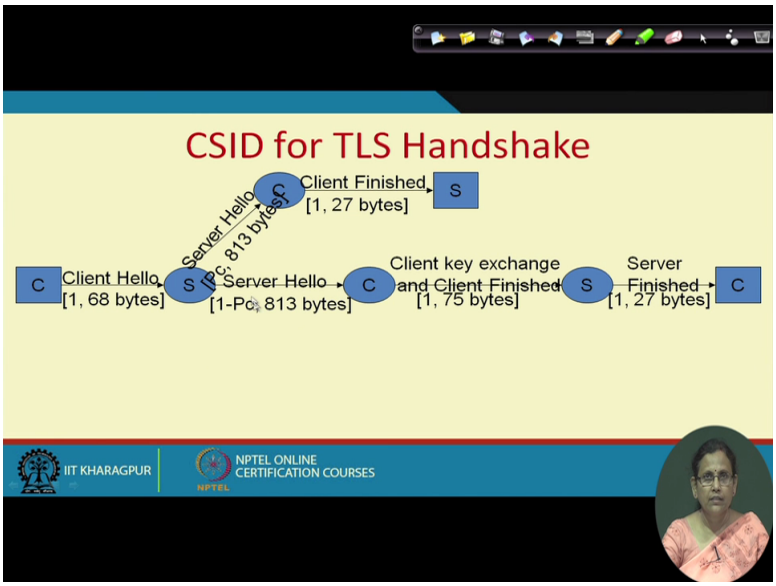
So this client node which is used using this which will be no using this looking at the C and server node will be represented with S and when the operation start there will be a start node which will be represented as a square and all these things will be in a circular shape.

(Refer Slide Time: 8:28)



So when the message goes from one entity to other will have a directed age connecting both the entities.

(Refer Slide Time: 8:43)



Now look at this this is a client server interaction diagram for TLS handshake protocol. So first the client when you will be sending a message to the server you will first start the interaction as a client through your browser you are sending some message to the server so first you have to start the interaction so first the client sends a client hello message in response server sends one server hello message.

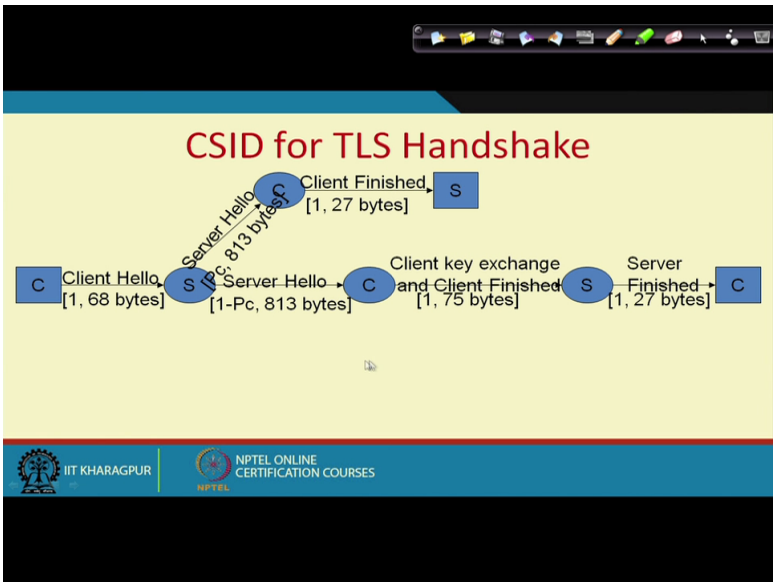
And this with response to this servers message once again the client sends some other message and finally the server finishes it with acknowledging the client finiseting finished it by acknowledging the servers message if you look at this diagram we have 2 paths usually during during TLS transaction the as we have already told you HTP is a state less protocol so in order to.

So which means if every time you make a connection over secure TLS protocol you have to be going through http because ht TLS protocol is in between TCP and http so the message has to pass through TLS layer as well. Now in during when this TLS operation takes place after later may be in next class we are going to talk about what is the implication of using the security protocol on server's performance.

But right now let me tell you it is actually a costly operation so therefore every time getting the server authenticated if consecutively we are going to have a number of interaction every time genere creating and sharing the symmetric key using private key public key operation is a pretty costly activity so therefore at the server and at the client send some session will be maintained.

And within that session within that time period if another secure message comes then the old symmetric key which was originally generated can be used.

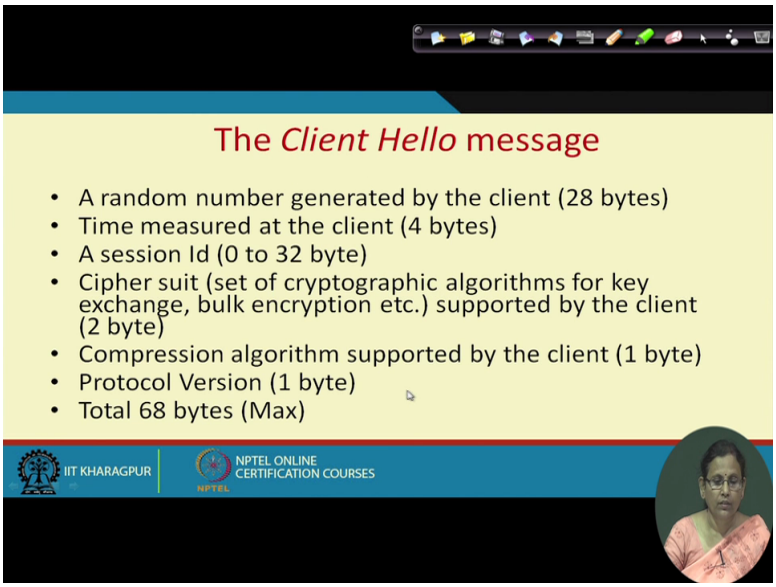
(Refer Slide Time: 11:49)



So here if that particular session is still there then there is no need for this key exchange again so therefore this during handshake procedure 2 path can be followed if the session is

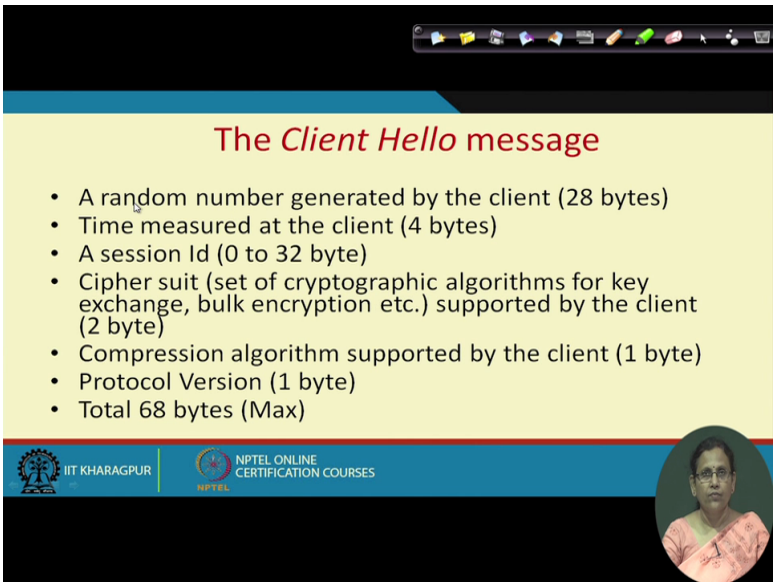
maintained then this can be directly send if the session is not maintained the key exchange once again has to take place.

(Refer Slide Time: 12:25)



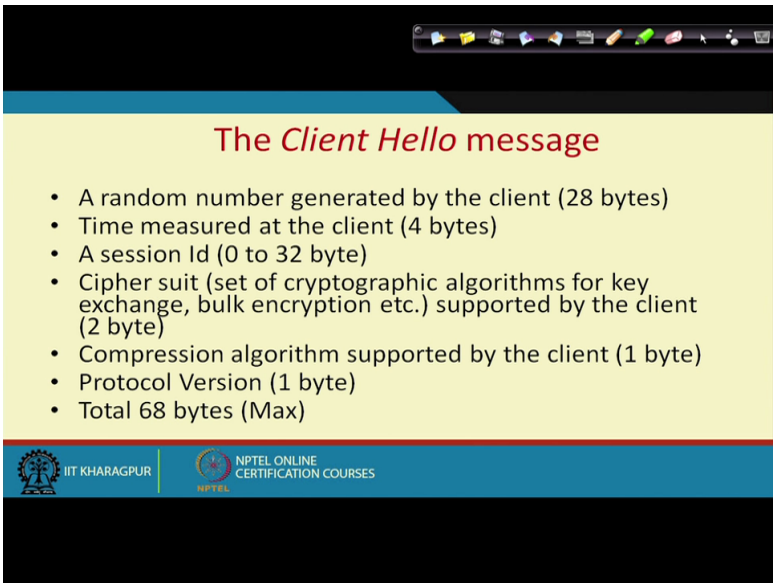
Now each of these messages that we saw just now client hello server hello etc let's see what all is included there.

(Refer Slide Time: 12:40)



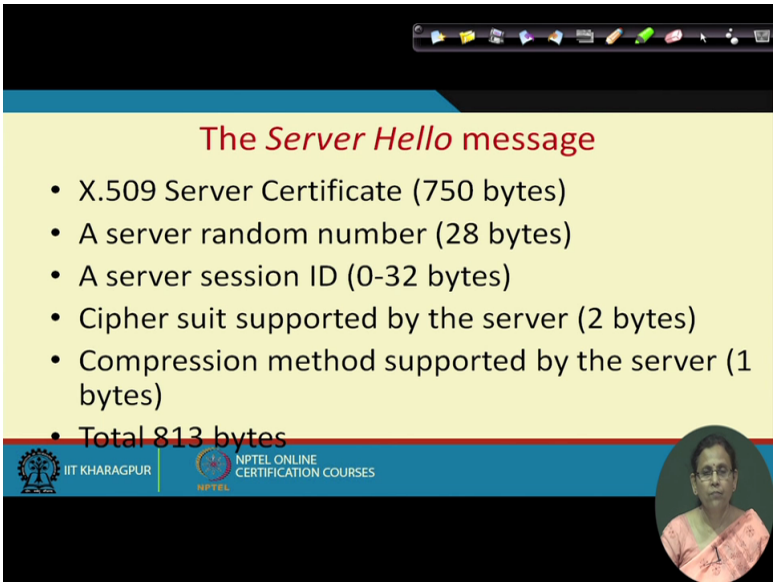
Client hello message through the client hello message what you send is a random number generated by the client the time measured at the client or session ID or cipher suit because now both the client and server are negotiating because during symmetric operation they are going to use the same cryptographic algorithm same what you say that data compression algorithm etc.

(Refer Slide Time: 13:12)



So therefore the cipher suit which is which is a name for the set of encryption algorithms and the other stuffs which is supported at the client side is sent to the server client now says I have this this this facilities so therefore while sending the data in encrypted form you have to use one of this one of this algorithm so that at my end I can use the same algorithm to decrypt it so it also send some protocol version and total 68 bytes it sends.

(Refer Slide Time: 14:01)

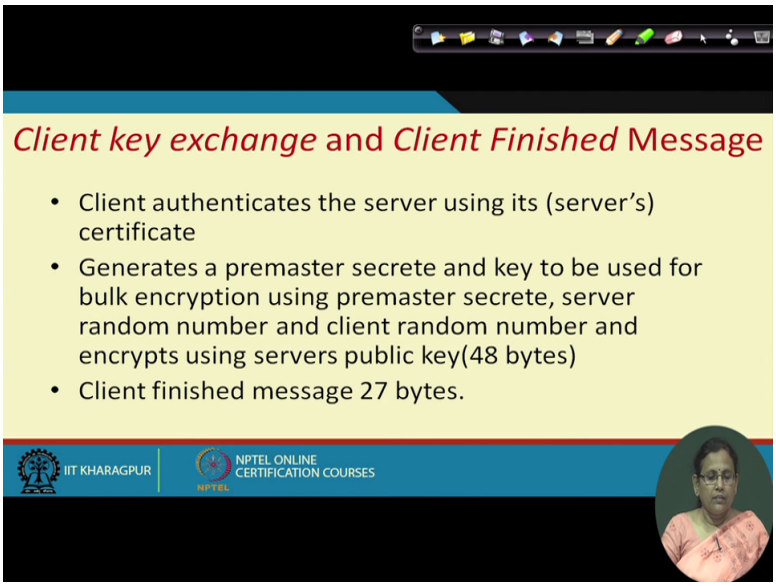


But that is not important but again from during server hello message what the service does server will be sending server will be authenticating itself and while talking about this digital Certificate we know that digital certificates are used for issued by certain certifying authority

and they are used to authenticate a server so server will be sending its digital certificate digital certificate contains many details about the server including its public key.

Then client send one random number server will send another random number to the client so client will have his own random client is now going is now having his own random number plus server random number similarly server is having clients random number which is sent during client hello message and his own random number and it also let the client no the cipher suit supported by itself and the compression method supported by itself so once the server sent this message.

(Refer Slide Time: 15:24)



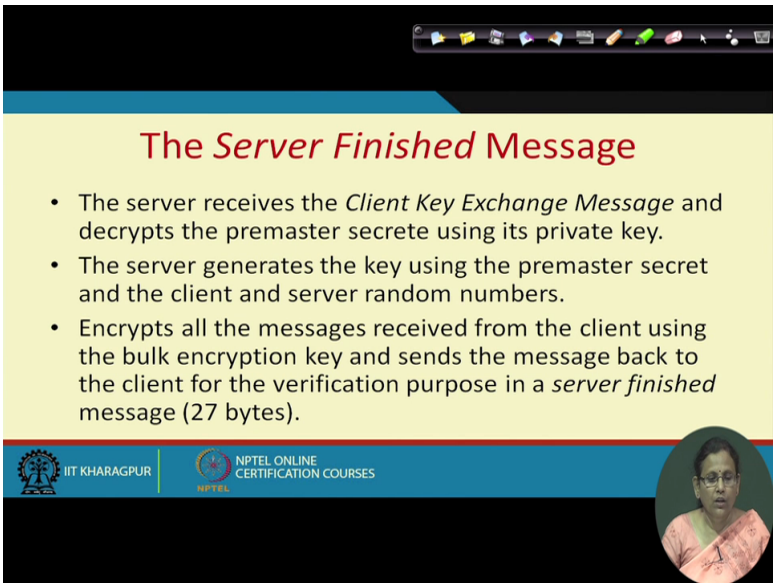
Then client will authenticate the server using the server certificate so what it will do? It will verify the certificate and about digital certificate verification we have already discussed last class that how the digital certificate gets verified so now this after the client verifies the digital certificate the client generates a pre master secret and key to be used for bulk encryption.

And how this key gets generated at the client side using some pre masters secret server random number, client random number and this pre masters secret is available at both the end so this pre master secrets server random number and client random number are shared then it encrypts it using servers public key.

Where from it get servers public key because server has already sent his public key during certificate while showing his certificate during server hello message so using all these three

things it will now it will now create a key and this key will be subsequently used by the during the bulk record TLS record protocol which is for bulk transfer.

(Refer Slide Time: 17:04)

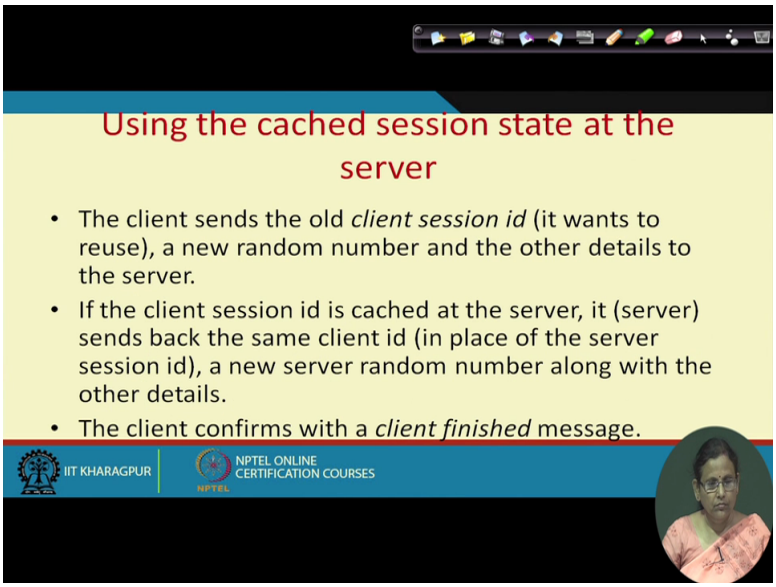


Now server receives this client key exchange message and decrypts the pre master secret using its private key because it is now encrypted with his public key. See public key is already sent the pre master secret which is generated at the client side see previously both the random numbers were shared now a pre master secret which is about which algorithm etc to be used is now shared with the server.

So now this pre master secret nobody can see because it was encrypted with servers public key which was already sent to the client so therefore using both the random numbers and this pre masters secret now the server creates the bulk transfer key at its end because both the entities use the same information for creating the key the key need not now to be explicitly transferred What got transferred transferred 2 random numbers.

Then certain using this private key operation some pre master secret which is about the algorithm etc is shared to the server so the key gets generated at the at both the end both at client and server end.

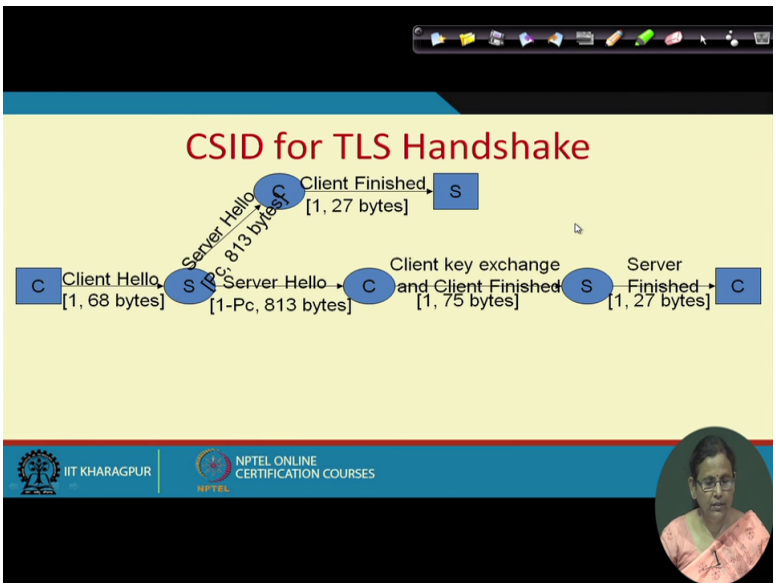
(Refer Slide Time: 18:41)



As I was telling you this whole process involves many cryptographic operations many security operations so which is bit costly so if see during this transaction some session ID was getting generated that I told you so if the session ID that a station ID has a life within that time period.

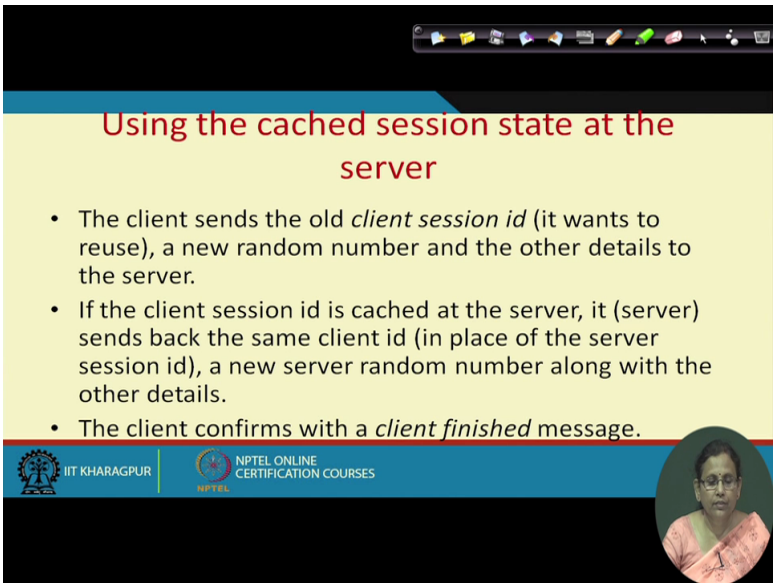
If further transactions continue them then the same session continue then the same key which got generated previously can be used this will actually decrease some load for encryption load from both client and server side so the client send the so in this particular path.

(Refer Slide Time: 19:36)



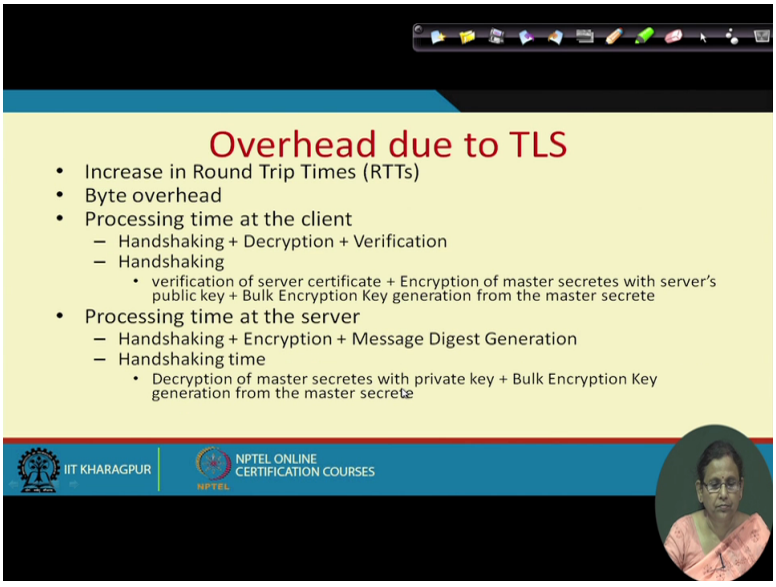
If you if we are going back we were talking about 2 paths per hand second so far we have been talking about this path and just now I told if the the session IDs are maintained by both client and server then this particular step which is for key exchange can be bypassed.

(Refer Slide Time: 20:06)



So if session IDs are cached this can benefit the transaction in terms of time the client sends the old client session ID if its wants to reuse it a new random number and other details to the server if the client session ID is cached at the server then server sends back the same client ID in place of the server session ID a new server random number along with the other details client confirms with the client finish message and both now can use continue using the same you can continue with the same session.

(Refer Slide Time: 20:47)



Now what is the overhead due to TLS this overhead due to TLS is because of the increase in the round trip time by round tip trip time time we mean when see look at the time of TCP

interaction you had one three way handshaking procedure so client connects to the server server gives some acknowledgement then client starts sending the data during TLS again you have another another.

Once again you are doing carrying out some kind of hand shaking procedure for generation of the symmetry and server for the server certificate verification and generation symmetric key so for this some you know the some overhead is incurred by the server as well as the client and the processing time increases so next class we are going to see how exactly processing time increases through one example.

But let us try to understand what is the traditionally time taken at the client sent client takes time takes time for hand shaking for decrypting the message for verify at the server it is handshaking encrypting the message and generating the message digest this thing again this hand shaking time hand shaking which again involves this encryption decryption is what?

And de compression and verification means what this is for the bulk data but hand shaking is for Key generation so key generation takes place plus bulk data transfer takes place for both these now some time is consume this handshaking is again has many steps verification of server certificate, encryption of master secretes with servers public key, bulk encryption key generation from the master secretes.

Then the processing at the server which is about handshaking at the server side and then encryption. Encryption of what encryption of the bulk data encryption of the bulk data along with the message digest generation this is for your record protocol but this is for your handshake TLS handshake protocol part.

And D here the what is the hand shaking time? Decryption of the master secrets with private key bulk encryption key generation from the master secrete at the server end. So all this time taken together adds overhead to TLS now at what factor each it actually increases that we going to see in the next class. Thank you very much!