Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #34
Basic cryptographic concepts - Part 3

In this lecture we shall be continue our discussion on cryptography concepts which we had been continuing over the last two classes. Now if you recall in the last two lectures we had talked about encryption, decryption. We talked about authentication using message, authentication code, the problem of key distribution and various kinds of encryption decryption schemes under the categories of symmetric and asymmetric key algorithms and so on. In today’s lecture we shall more talked about some applications or some security solutions that are available on the internet scenario which utilizes these basic cryptographic tools and primitives. Now when we say network security applications we refer to certain tools and techniques which people can use over the internet today to carry out secure transactions to send and receive different kinds of data and information in a secure way provide authentication and so on. So we shall basically look at some of the very common security solutions that are in use.

(Refer Slide Time: 02:18)

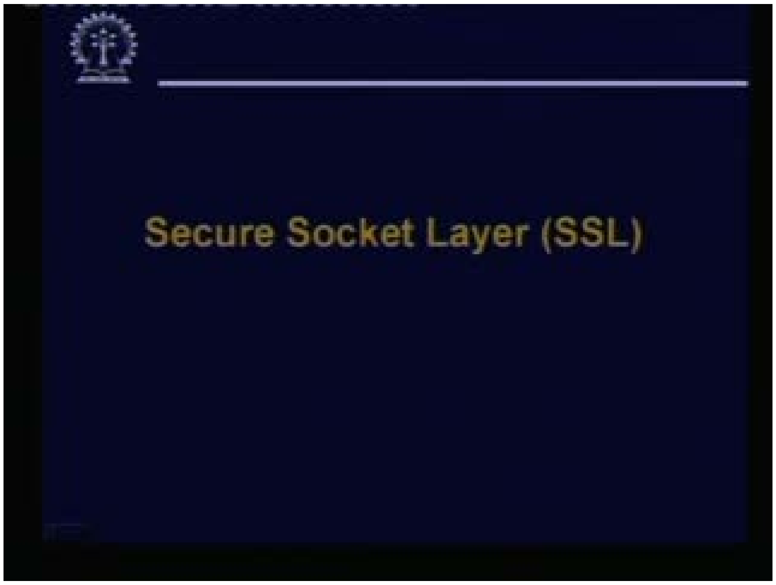
Indian Institute of Technology Kharagpur

Basic Cryptographic Concepts
Part III

Prof. Indranil Sen Gupta
Dept. of Computer Science & Engg.
I.I.T. Kharagpur, INDIA

So today’s lecture title basic cryptographic concept part three.

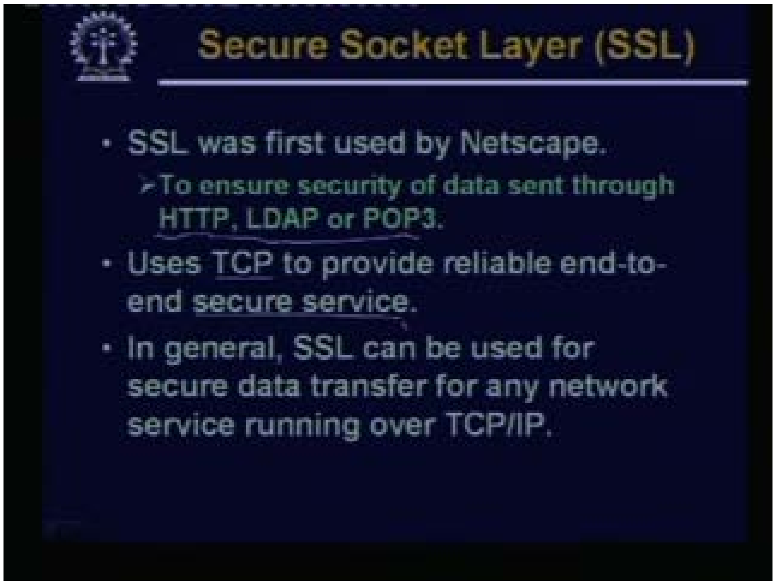
(Refer Slide Time: 02:23)



Secure Socket Layer (SSL)

First we shall be talking about a protocol called Secure Socket Layer or SSL.

(Refer Slide Time: 02:33)

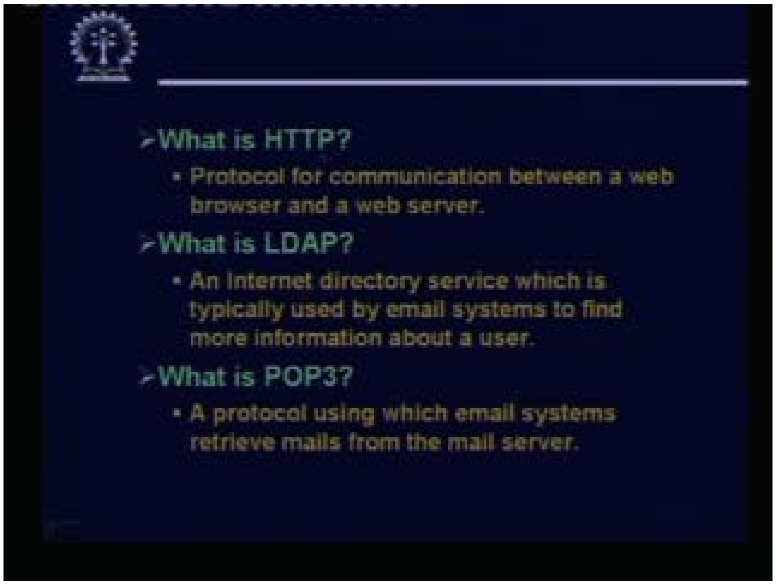


Secure Socket Layer (SSL)

- SSL was first used by Netscape.
  - To ensure security of data sent through HTTP, LDAP or POP3.
- Uses TCP to provide reliable end-to-end secure service.
- In general, SSL can be used for secure data transfer for any network service running over TCP/IP.

Secure socket layer protocol was first proposed and used by the company named Netscape who are more commonly known for the browser. No SSL was initially proposed by Netscape to ensure security of data which you have sent through certain kinds of protocols which were very popular at that time. Well HTTP, LDAP, POP3, these were the three very popularly used protocols when this SSL first came into being. We shall be looking at these protocols very shortly. But the basic idea was that all these protocols they sometimes send some information across the network which may be deemed to be sensitive. So in order to hide or protect this kind of sensitive information that are flowing through the internet through a public communication network. This new protocol SSL was proposed. Now in SSL we basically used TCP to provide reliable end to end secure service. So the transport layer protocol that is used for SSL is TCP it runs on top of TCP. So basically since it runs on top of TCP, so if you have any kind of network service or application running on the top of TCP/IP, SSL can be used on top of that to provide secure data transfer. So what this means is as follows that SSL is not restricted to any set or classes of applications any applications which you are running on top of TCP can take help of SSL to provide security in the data transmissions.

(Refer Slide Time: 04:34)

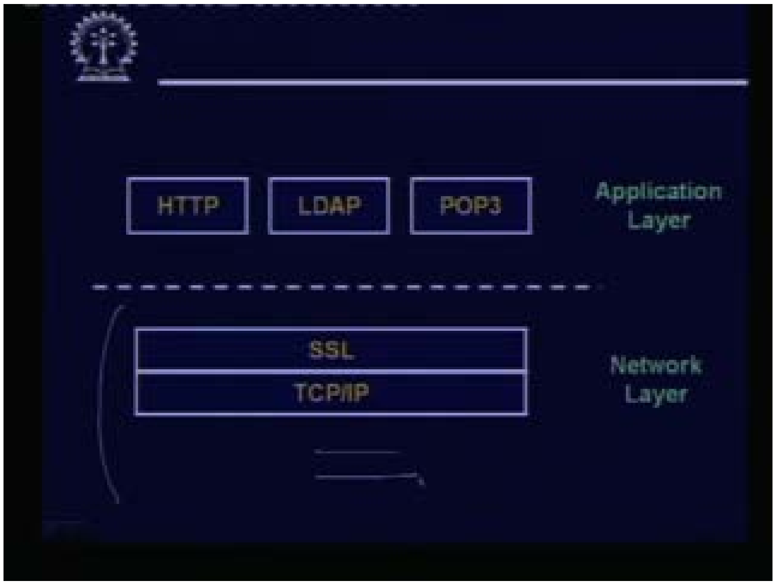


- What is HTTP?
  - Protocol for communication between a web browser and a web server.
- What is LDAP?
  - An internet directory service which is typically used by email systems to find more information about a user.
- What is POP3?
  - A protocol using which email systems retrieve mails from the mail server.

Now the three applications that we talked about, HTTP, LDAP, POP3 which were the initial targets of SSL. Well HTTP, and POP3, you already know. HTTP is the protocol which is used for communication between a web browser and a web server. This you already know. Similarly POP3 also you have studied. This is used in email systems. This protocol is used to retrieve mails from the mail server. Well however we have mentioned that, well POP3 is an older version. Nowadays we use an improved version of protocol called IMAP. Talking about LDAP, this is called Light-weight Directory Access Protocol. This is basically an internet directory service. Directory service means some information is located in some central place.

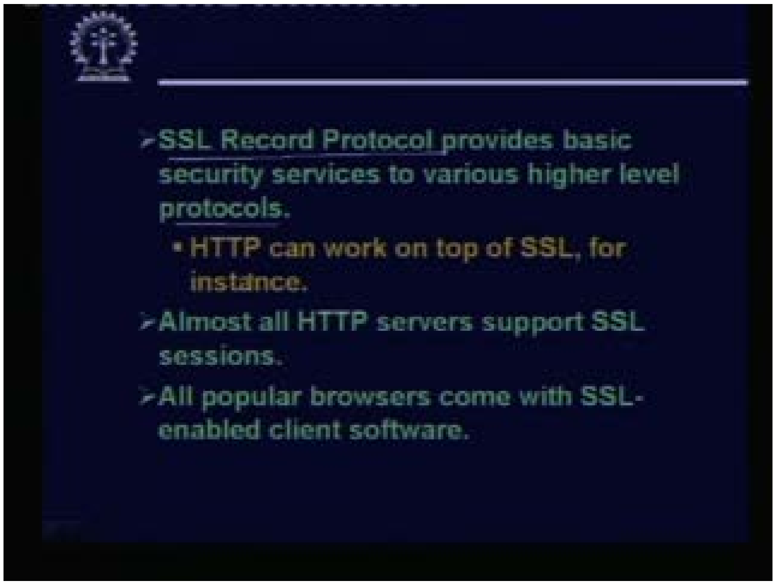
On some servers whenever I need information I can send a request to that directory server and the directory server sends me back with the requested information. Now basically the directory server stores information about people, about users of the internet about users of electronic mail system. So this LDAP service is this directory service this is used typically by email systems to find some information about a particular user. Now sometimes the information that you are trying to retrieve, this may be some sensitive information which you do not want. The other people who may be snooping on the net might be having access to. So for this kind of communication you need some kind of security or secrecy while the data is being transferred or communicated.

(Refer Slide Time: 06:36)



So the picture looks somewhat like this. While here we are showing only the partial TCP/IP protocol stack as modified because, below TCP/IP you will be having physical or the network layers. So the TCP and IP I am showing in a single block. But actually it will be the TCP layer on top IP layer on bottom and SSL is an additional layer we are inserting on top of TCP. It is like this. This SSL is an additional which you are inserting on top of TCP and the application layer is assumed to be on top of this. So as I said HTTP, LDAP or POP3, these are the typical application layers for which SSL was initially designed. But again I had mentioned that any application which runs on top of TCP can utilize the functionalities of SSL if it so desires but this is the modified protocol stack. Now essentially we have added one more layer between TCP and the above application layer that SSL layer can provide certain security related services to the application.

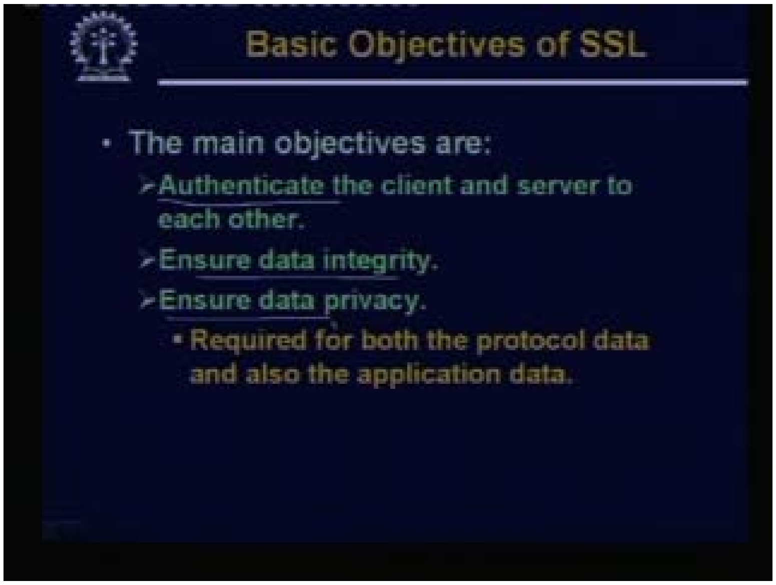
(Refer Slide Time: 07:55)



- SSL Record Protocol provides basic security services to various higher level protocols.
  - HTTP can work on top of SSL, for instance.
- Almost all HTTP servers support SSL sessions.
- All popular browsers come with SSL-enabled client software.

Now in SSL there are several different component protocols. Those are protocols whatever you say, there is one protocol called SSL record protocol this particular protocol provides the basic security services to various higher level protocols at the application layer. Now as I had shown in the previous example HTTP, this you can regard as an application running on top of TCP this can work on top of SSL also. So the SSL record protocol provides basic services to HTTP with respect to security. Nowadays almost all HTTP servers which we see they can support SSL based sessions. Which means I can have certain web pages or web sites which must be accessed or viewed or the data transmitted to it through SSL layer; others you can bypass SSL and go to TCP directly and also on the other side all popular browsers. Nowadays explorer, Mozilla, whatever they come with SSL enabled client software that means in your browser you can also access some information on a server using the SSL protocol if you want or desire.

(Refer Slide Time: 09:38)

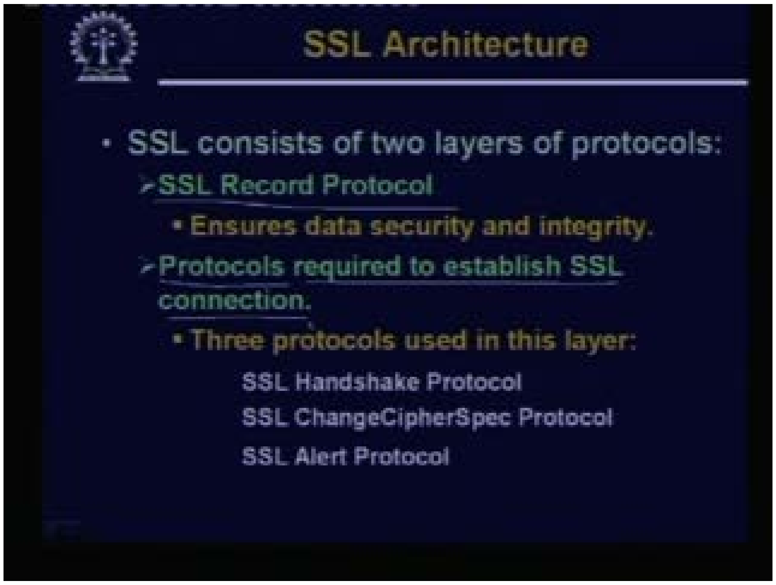


Basic Objectives of SSL

- The main objectives are:
  - ➢Authenticate the client and server to each other.
  - ➢Ensure data integrity.
  - ➢Ensure data privacy.
    - Required for both the protocol data and also the application data.

Now let us look back into the basic objectives of SSL. What is the kind of services SSL aims to provide to the upper layers? The main objectives are these three. First is authentication the client and the server must authenticate to each other so that the client knows that with whom it is interacting with. Second is to ensure data integrity to make sure that an intruder sitting in between is not modifying the data. Or due to some network error some data is getting modified such things should not happen. That is data integrity and third thing is of course you are protecting data from prime eyes of intruders ensuring data privacy. Now when we talk about data privacy, I mean as a user, as an application I would like that the data that I am trying to transmit should be kept secure. But there is another level to it. Data privacy is required from the application data point of view as I mentioned also from the protocol data point of view. This means that the SSL protocol itself might be using certain parameters which it would not like to disclose to the intruder. So those parameters the information about them should also be kept secret in the actual communication.

(Refer Slide Time: 11:22)



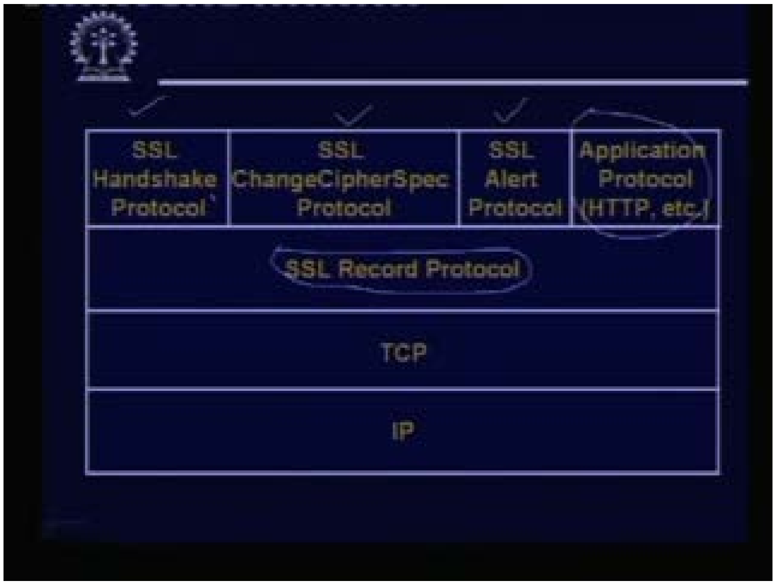
SSL Architecture

- SSL consists of two layers of protocols:
  - SSL Record Protocol
    - Ensures data security and integrity.
  - Protocols required to establish SSL connection.
    - Three protocols used in this layer:

      SSL Handshake Protocol
      SSL ChangeCipherSpec Protocol
      SSL Alert Protocol

So talking about the SSL architecture broadly speaking there are two layers. The lower layer is the so called SSL record protocol layer. The primary purpose of the SSL record protocol is to ensure data security. And integrity as I mentioned just now and above SSL record protocol there is another layer. Actually there are a number of protocols here which are required to establish SSL connection and also required during an SSL communication. Now at this second or higher layer, there are three protocols that are used. First is called the SSL handshake protocol. Second is called the SSL ChangeCipher protocol. Third is called the SSL alert protocol. So what I have said is that in SSL, there are two layers of protocol. At the lower layer we have the SSL record protocol. At the upper layer there are three different protocols. They have certain well defined functions as we shall see shortly.

(Refer Slide Time: 12:43)



SSL Handshake Protocol

SSL ChangeCipherSpec Protocol

SSL Alert Protocol

Application Protocol (HTTP, etc.)

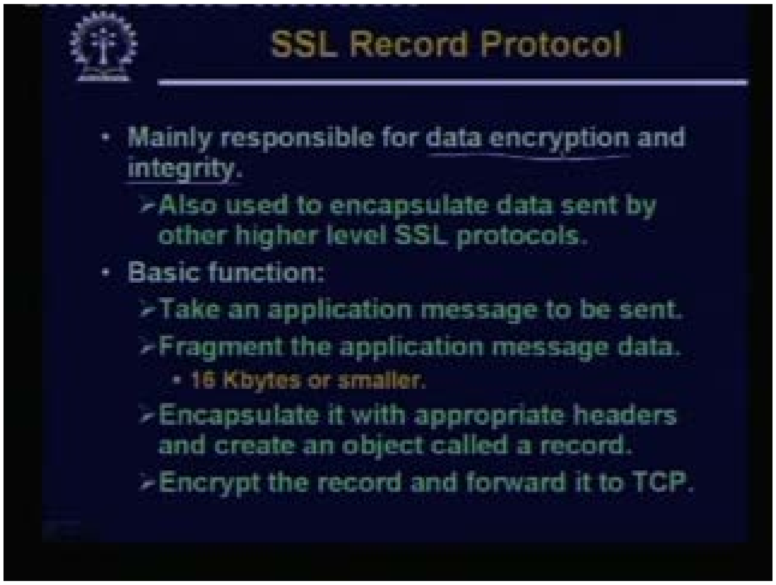
SSL Record Protocol

TCP

IP

The picture now looks like this. This is a refined diagram in the earlier diagram we have simply said on top of TCP. There is SSL and top of it we have applications. But now there is a refinement on top of TCP. We have a part of SSL. This is the SSL record protocol and the applications whatever are running it can be HTTP, LDAP, POP3 or anything else. They actually require the SSL record protocol only. And the other SSL related protocols which are mainly required during connection establishment within the two end parties. The handshake protocol, ChangeCipherSpec and Alert, these three also run at the application layer above the SSL record protocol. See the idea is like this. The SSL record protocol provides a mechanism to transmit some packets in an encrypted fashion so that intruders cannot hide. On top of it there are protocols which are used for negotiating a connection. Now the parameters that you are using for negotiating that may also be confidential or you would not like them to be disclosed. So those protocols also use the SSL record protocol to send those data in a secret fashion. So this is the basic principle. The SSL record protocol is mainly responsible for secrecy of the information transmitted.

(Refer Slide Time: 14:23)



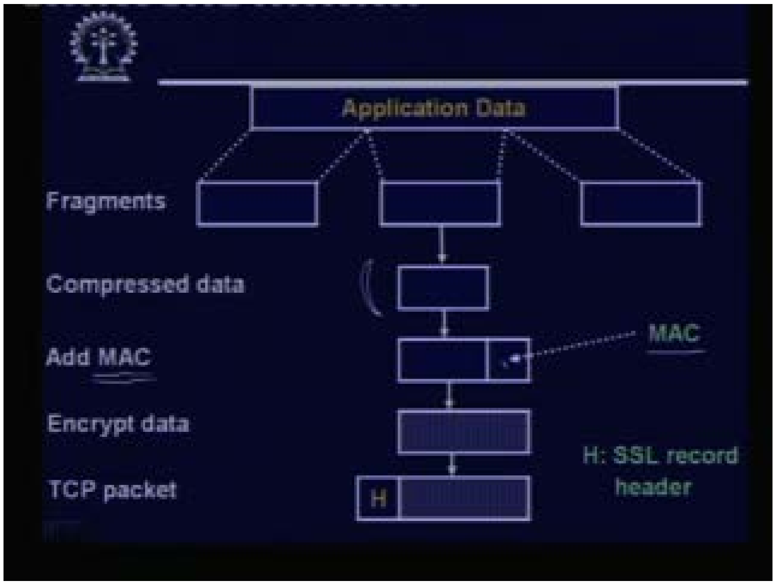
SSL Record Protocol

- Mainly responsible for data encryption and integrity.
  - Also used to encapsulate data sent by other higher level SSL protocols.
- Basic function:
  - Take an application message to be sent.
  - Fragment the application message data.
    - 16 Kbytes or smaller.
  - Encapsulate it with appropriate headers and create an object called a record.
  - Encrypt the record and forward it to TCP.

So some details about the SSL record protocol here. As I said earlier this layer is responsible for data encryption and integrity data encryption because I do not want or the applications may not want that the data that is being transmitted in the packet should be viewable by an intruder. Should be retrieved and decoded by an intruder and integrity means no one should be able to modify the data during the transmit. And as I just mentioned this particular layer is also used to encapsulate the data sent by the other high level SSL protocols. Basic function of this record protocol is as follows. It first takes a message to be transmitted. This you are calling an application message. The first step it does is that if the message is too big it breaks or fragments the message into smaller junks.

Now the junks are 16 kilobytes or smaller in sizes. For each of this junks proper encapsulation is done and appropriate headers are added to create an object which is called a record. Now this whole record is encrypted and the encrypted data is sent to TCP for actual transmission. So what the basic concept here is that before we actually give the data to TCP for transmission we do certain preprocessing on it. If it is too big, we break it up, we append some headers; we do some encryption. After we do everything we give it to TCP. TCP does not know what is happening about it. TCP simply takes the data as a message unit and it transmits the data as a packet to the other side.

(Refer Slide Time: 16:47)

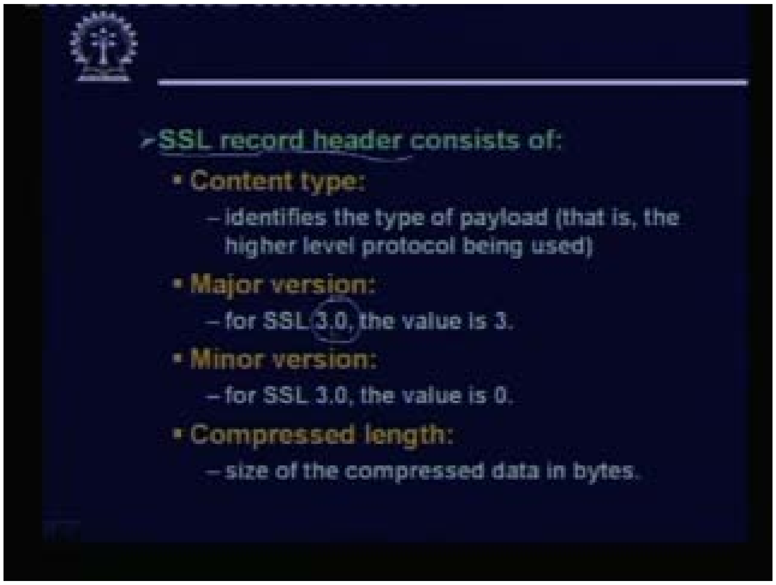


This diagram summarizes what are the kinds of transmissions or mapping that are carried out on the data before being sent to TCP layer for transmission. It starts with the application data at the top. You see in this diagram we have shown that the application layer or the application data is too big, so that it is broken up into three fragments. Here we have shown the flow for one of the fragments. Similar things would happen for all the fragments. The first step is in order to reduce the volume of data transmitted. The first step is to compress the data into it become smaller in size. So compression is the first step in the SSL record protocol.

It takes a message fragment, it compresses it using a standard compression algorithm and after compression it computes a message authentication code MAC and it adds this MAC to the compressed data. So now you have a data in it, where you have the compressed data and an authentication code check what kind of a thing signature kind of a thing which is computed on the compressed data. In the next step you treat this whole unit including MAC as a single data unit and perform encryption and you get an encrypted version of this thing. Lastly you appended so called SSL record headers which will contain relevant information so that the SSL layer at the other side will be able to decrypt or decode it in a proper way. So here we have the headers added H to this and you are transmitting.

So this is how the SSL record protocol works. Now at the receiving side just the reverse thing would take place. The TCP packet would be received. The header analyzed and stripped off. Decryption will take place. MAC will again be recomputed and verified. Then decompression will take place you get back the original data. So as you can see here we are using a combination of compression, message authentication code and encryption. Compression is used for two purposes. One to reduce the size of the data to be transmitted because since we are anyway adding this MAC, so in order to reduce the overrate of the data transmission you try to reduce it to the extent possible. MAC is used to ensure integrity and encryption is used to ensure data security.

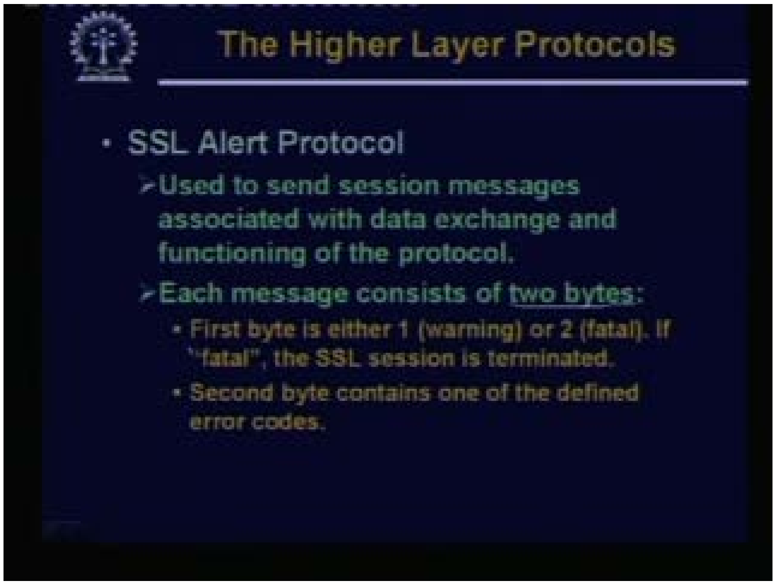
(Refer Slide Time: 20:05)



- SSL record header consists of:
  - Content type:
    - identifies the type of payload (that is, the higher level protocol being used)
  - Major version:
    - for SSL 3.0, the value is 3.
  - Minor version:
    - for SSL 3.0, the value is 0.
  - Compressed length:
    - size of the compressed data in bytes.

Now if you look into the SSL record header while here we are not showing the detail header fields but these are the basic fields which are present in the header. There is a field called content type. Content type mentions what is the type of payload or the higher level protocol which is being used. Means is it HTTP? Is it LDAP? Is it the SSL alert protocol? Or what kind of application or program on top of it has given some message to it for transmission? So it identifies the so called type of the content there is a concept of a major and a minor version. Now SSL comes in various versions. For example 3.0 can be the current version. So if it is 3.0, then 3 is termed as the major version and 0 is called the minor version and the header consists of the compressed length of the data in bytes. Since we are compressing before trying to transmit, so instead of storing information about the original message we store information only about the compressed version of the message.

(Refer Slide Time: 21:29)

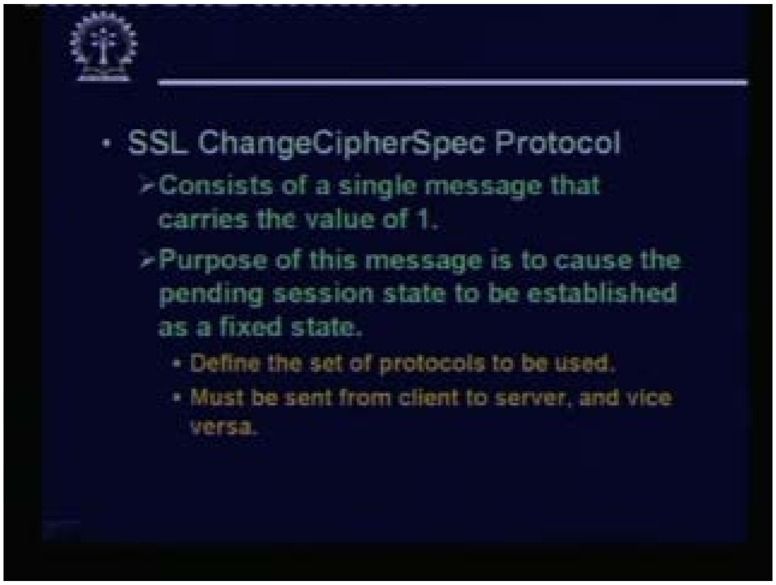


The Higher Layer Protocols

- SSL Alert Protocol
  - ➢Used to send session messages associated with data exchange and functioning of the protocol.
  - ➢Each message consists of two bytes:
    - First byte is either 1 (warning) or 2 (fatal). If "fatal", the SSL session is terminated.
    - Second byte contains one of the defined error codes.

Now talking about the high layer protocols SSL alert protocol, now alert protocol as the name implies it provides some kind of an alert to some entities. This protocol is used to send session messages associated with data exchange and proper functioning of the protocol which means essentially the time of connection establishment. And also during the progress of the session during the time the SSL session is active if there is some anomaly some error some warning this particular protocol can be used to transmit. This kind of warning or error message to the entity concerned. Now SSL alert protocol is very simple. Here the message consists of only two bytes. The first byte will be 1 or 2. One will mean that it is warning message which is some kind of warning message but the operation will continue with respect to the session or two means fatal. If it is fatal then this session is terminated. It cannot proceed any further the kind of error that has taken place is designated or classified as fatal. Second byte contains one of the defined error codes. So when there is either a warning or a fatal error the second byte will indicate that what kind of a warning or message has occurred. There is a table; there is a list that each number will indicate the kind of error or one which has occurred. So this protocol is used just for sending error messages.

(Refer Slide Time: 23:40)

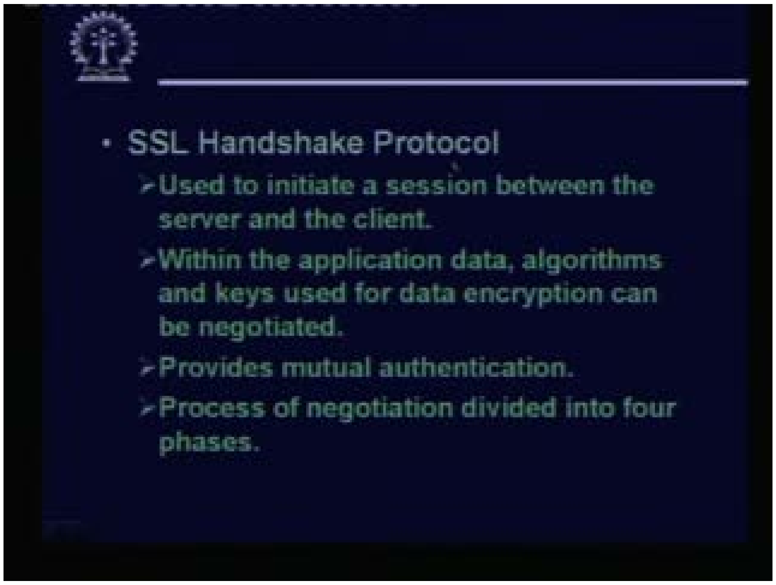


- SSL ChangeCipherSpec Protocol
  - Consists of a single message that carries the value of 1.
  - Purpose of this message is to cause the pending session state to be established as a fixed state.
    - Define the set of protocols to be used.
    - Must be sent from client to server, and vice versa.

There is another protocol this is also very simple ChangeCipherSpec protocol. This is more like an acknowledgement sending protocol. This protocol is even simpler. It consists of a single message that carries the value one. This message is used more like an acknowledgement. The purpose of this message is to cause the pending session state to be established. Like I am giving the example why this is needed? Suppose initially when two parties, client and the server are trying to negotiate on some parameters like what cryptographic algorithm should be used. What is the key size? What will be the secret key that means if we you use public key cryptography?

They will be sharing the public keys and so on. But these are some information which are exchanged between the two parties. But until and unless the ChangeCipherSpec protocol message is transmitted, this information which was transmitted will not be registered and will not be started to be used. So once this message goes from client to the server and server to the client, both the parties will not start using the parameters which have been agreed upon. So this is more like an acknowledgment protocol. For definition of set of protocols and as I said this kind of a message must be sent from client to server and also server to the client.

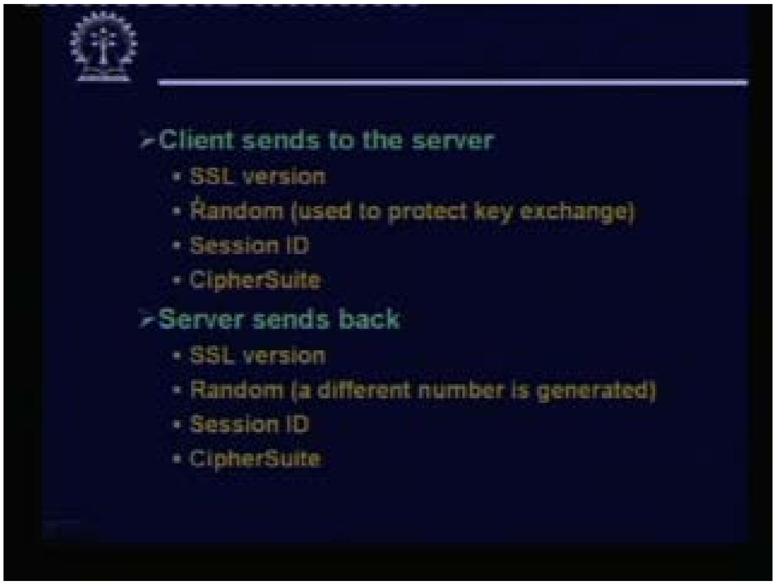
(Refer Slide Time: 25:21)



- SSL Handshake Protocol
  - ➢Used to initiate a session between the server and the client.
  - ➢Within the application data, algorithms and keys used for data encryption can be negotiated.
  - ➢Provides mutual authentication.
  - ➢Process of negotiation divided into four phases.

SSL handshake protocol is actually used for sending the parameter details between the client and the server. This protocol is used basically during the creation of a session for initiating a session between the server and the client. Now under this protocol as I said algorithms which encryption which compaction which MAC algorithm will be used. What keys will be using what size? So all this things can be negotiated and SSL handshake protocol because it is negotiating between two parties. It must do so only after mutual authentication. Because this is something important which is going on between two parties because, after this step possibly they will be starting to exchange some very confidential information. So during this step mutual authentication must be carried out. This is a very important step. And during this phase the process of negotiation, whatever is called we can divided it into four different phases. Negotiation means negotiation between the client and the server on the different parameters to be used.

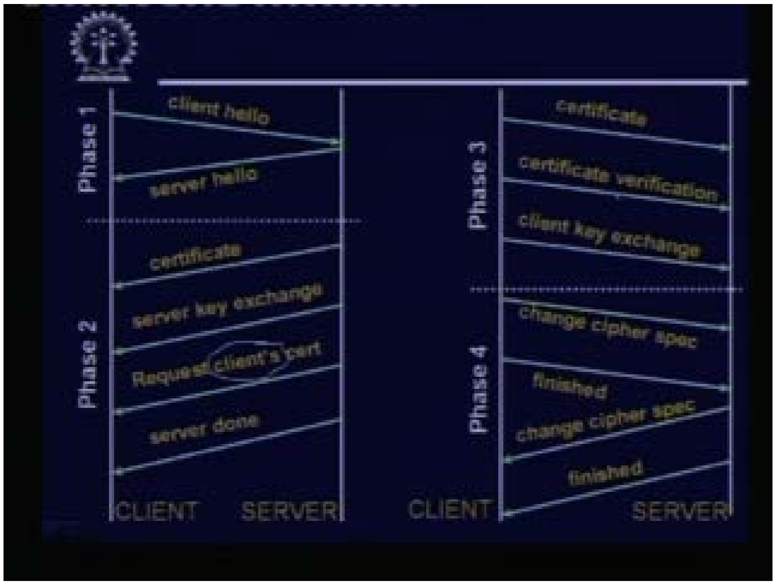
(Refer Slide Time: 27:00)



- Client sends to the server
  - SSL version
  - Random (used to protect key exchange)
  - Session ID
  - CipherSuite
- Server sends back
  - SSL version
  - Random (a different number is generated)
  - Session ID
  - CipherSuite

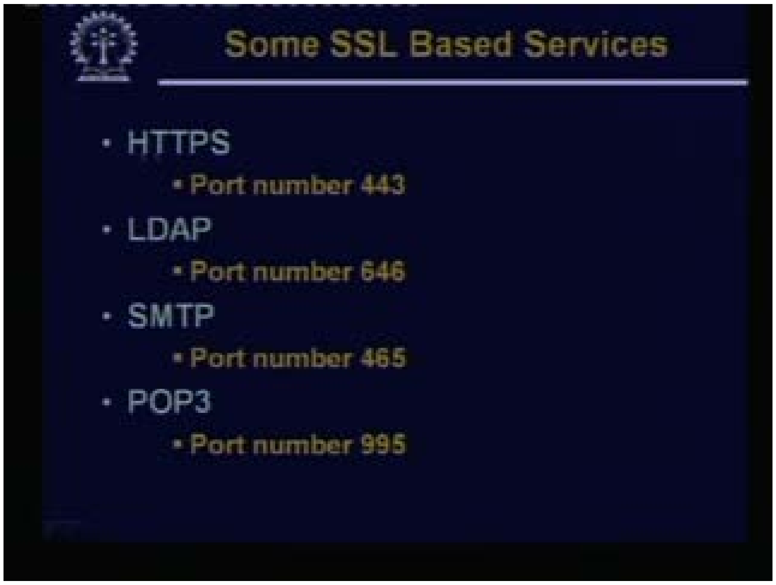
Now we shall be seeing the different phases. But first let us see what are the information the client requires to send to the server during this step the version of the SSL that it is using because it is quite likely that the client and the server are not using the same SSL version. So the side which is using the higher version will be downgrading itself into the lower one. But the reverse is not possible in general. The client will also send to the server some kind of a random number which is used for key exchange. Some session ID which is generated and the detail CipherSuite which have to be used. Like the encryption algorithm, other algorithms size of the key and so on. Server must also send back similar information like SSL version again some random number which the server generates. Typically this is a different number as compared to the clients Session ID, CipherSuite. So both the side has to agree on certain things. So other than the random this SSL version session ID CipherSuite these are some things which have to be agreed upon before the active communication can start.

(Refer Slide Time: 28:13)



Now this diagram will show you the four phases of this negotiation process. First phase; the client and the server exchanges are hello message just to tell each other that we are alive and running. Client sends a hello message to the server the server acknowledges that with a hello message. So both the side knows that the other side is active and up. Now in phase two the server first coactively sends certain things to the client; first is the certificate. Now we shall be talking about certificates later in some future lecture. Certificate is something which is used to verify the authenticity of the server. There are some certifying authority certificate granting authority who among other things also grants a public to a particular company or entity. Suppose I want to contact that particular entity I will first ask the certificate from that particular party. So after I get this certificate I can verify it again through the certifying authority if I want then I can start the actual communication. So the certificate is being sent. Then server sends some keys to the client, requests client certificate because server also needs to know the authenticity of the client. Their client is a legal party in this transaction and finally a done kind of a message indicating that there is nothing more to send. So after the client receives all these things, it starts responding back. First it responds back with its own certificate. It also sends back information that it has verified the certificate which you have sent by the server. So the certificate verification information is also sent back and some information about key exchange. So the server sends its own public key the client sends its own public key to the other side. So now both the sides have each others public key with the help of which they can start communicating. And in the fourth phase, this is the final acknowledgement. This comes in the form of the ChangeCipherSpec message. The client says ChangeCipherSpec then a finished message, server does the same thing. So these are the four steps which are followed by the client and the server in order to negotiate the parameters to be used for the connection. So after this step we can regard that the connection has been established, it is now working.

(Refer Slide Time: 31:25)

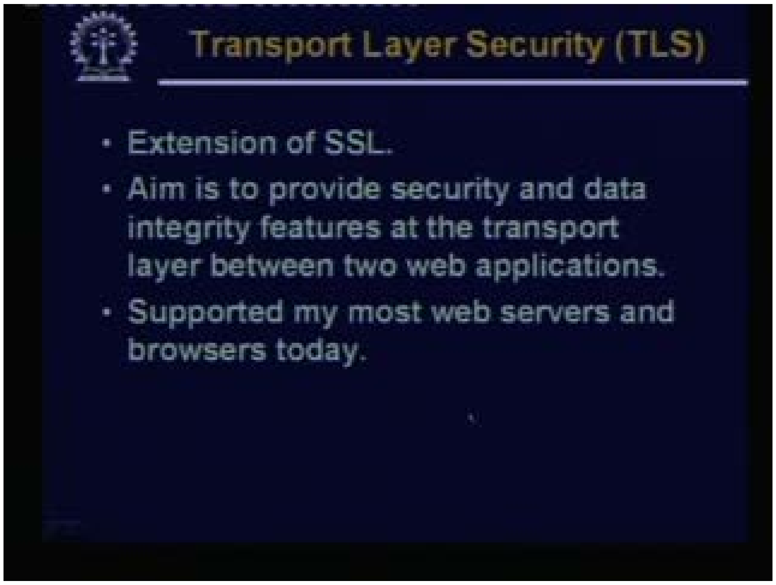


Some SSL Based Services

- HTTPS
  - Port number 443
- LDAP
  - Port number 646
- SMTP
  - Port number 465
- POP3
  - Port number 995

Now some of the popular applications of services which runs on top of SSL. They use certain well defined port numbers. Now some port numbers are as follows. Well HTTP which is running on top of SSL. This is called HTTPS it uses port number 443, LDAP uses 646, SMTP uses 465, POP3 uses 995.

(Refer Slide Time: 31:55)

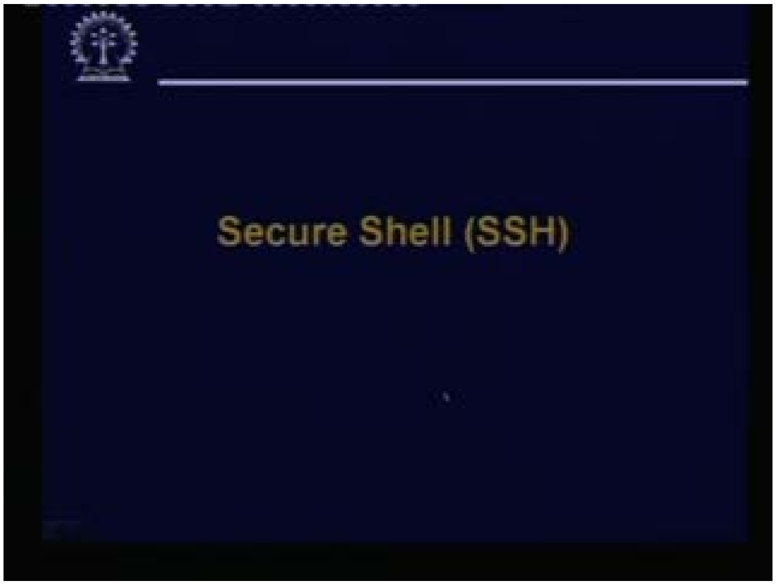


Transport Layer Security (TLS)

- Extension of SSL.
- Aim is to provide security and data integrity features at the transport layer between two web applications.
- Supported my most web servers and browsers today.

SSL, as I said is a protocol which runs on top of TCP. Now there is another protocol which is also proposed called transport layer security. I am not going in to detail of this. This is basically an extension of SSL. Here the aim is to provide security and data at the transport layer between two web applications. This is supported by most web browsers, servers and browsers today. See here the idea is that in SSL each and every packet are encrypted independently and sent. But here what we are talking about is that at the level of the transport layer the two applications on two sides. One side wants to a send a very large message to the other side. They can start continuously sending the message without breaking them into packets and handling each packet independently as SSL does. So TLS is a solution which aims to do just that and TLS is also supported by most web servers and browsers today. So both TLS and SSL are in use nowadays.

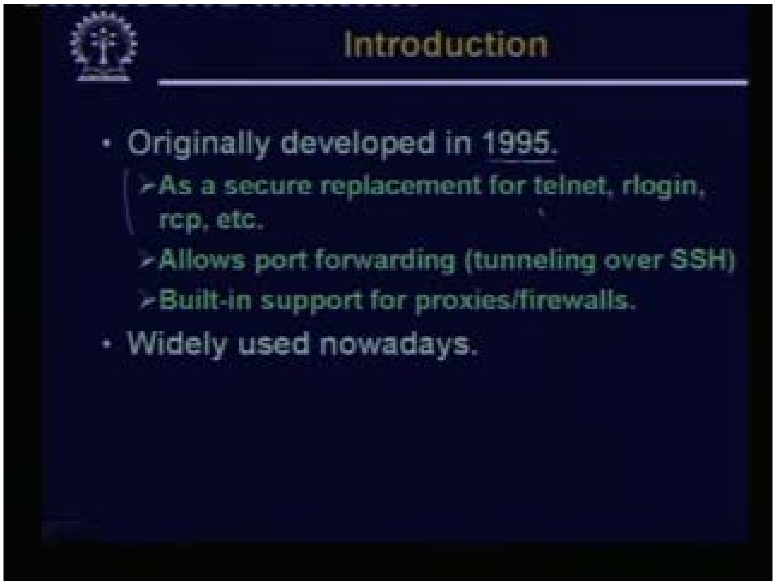
(Refer Slide Time: 33:10)



Secure Shell (SSH)

Now let us talk of something else something related secure shell or SSH. See the things that we would be talking about. See we have talked about secure socket layer. We have talked about transport layer security. Now we are talking about secure shell. Now these security solutions are evolved at different periods of times based on certain different sets of requirements. It is not that at any given point in time you need to have all of these to be used to work in a secure way with respect to certain applications. Sometimes these are supplementary. You can use any one of these two. So let us see what this secure shell really is?

(Refer Slide Time: 34:03)



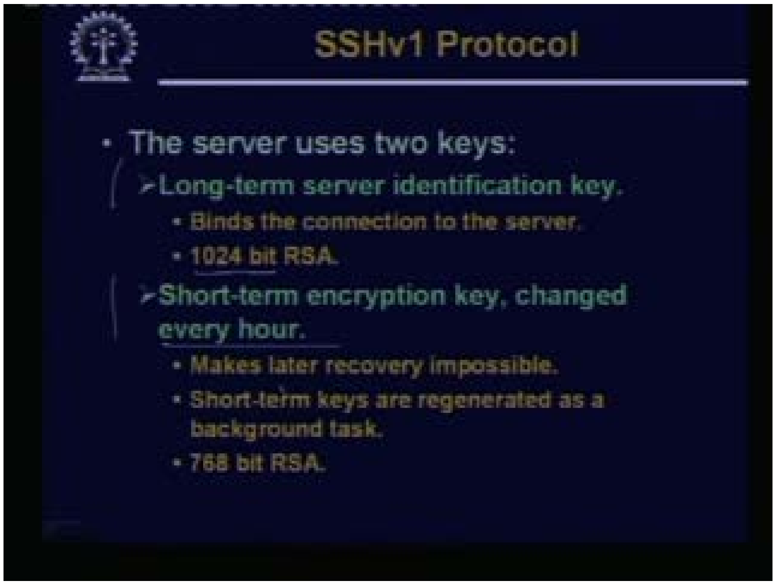
Introduction

- Originally developed in 1995.
  - As a secure replacement for telnet, rlogin, rcp, etc.
  - Allows port forwarding (tunneling over SSH)
  - Built-in support for proxies/firewalls.
- Widely used nowadays.

This again was originally developed long back in 1995. See you know at that time one of the very common kind of network applications that people used to use were some kind of remote login either TELNET or remote login rlogin or remote copy RCP. These were utilities which were over the network. Sitting here I can copy my file over the network to another machine. I can do a telnet, I can do a rlogin, but for each of these operations you need some authentication in terms of username and password. But in the simpler version of this application whenever you type user name and password they were transmitted to the other side.

So if someone was running some kind of a snooping program a packet cipher. They would very easily get to know your username and password. Now if you have SSH versions of these protocols TELNET, RCP and so on there the basic idea is the same. But the difference is that whatever is getting transmitted they are encrypted before transmission. So that an eaves dropper even if it snoops on the net would not be able to get the confidential information which earlier it was so easy to get. This uses so called port forwarding or tunneling. This is a built in support with most proxies or firewalls used today and as I said this is also widely used nowadays.

(Refer Slide Time: 35:56)



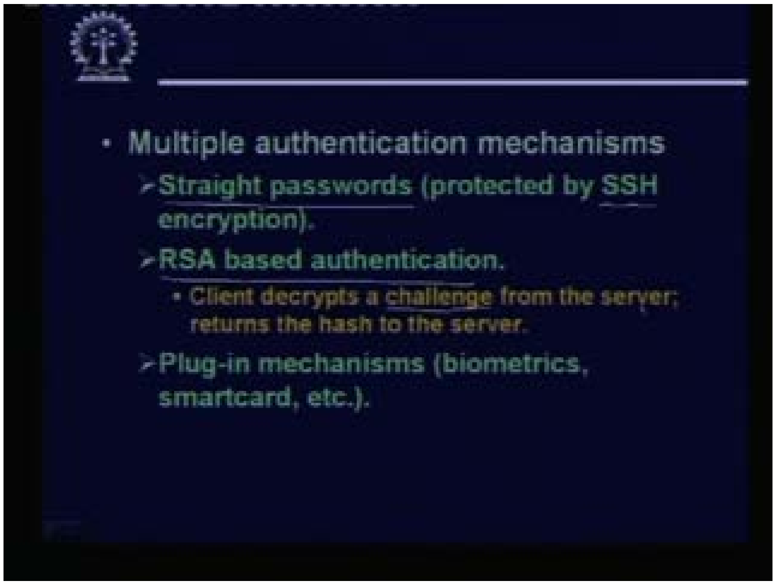
SSHv1 Protocol

- The server uses two keys:
  - Long-term server identification key.
    - Binds the connection to the server.
    - 1024 bit RSA.
  - Short-term encryption key, changed every hour.
    - Makes later recovery impossible.
    - Short-term keys are regenerated as a background task.
    - 768 bit RSA.

Now in SSH version one the server uses two keys. One is long term other is short term. Both these keys are based on public key cryptography. Long term server identification key, this is a key which binds the connection to the server. This is a 1024 bit RSA, actually this is the public key. I am a server. I am sending out the public key to you. That public key will be binding yourself to me in the sense that now you will be able to send some information to me encrypted by that. This is the long term server identification key and there is another short term encryption key which is changed every hour. This is used because if an eaves dropper somehow can break it next time some other key will be used so that the latter recovery will not be possible.

Here we use a 768 bit RSA which takes relatively much time for encryption decryption as compared to 1204 bit RSA, short term keys are regenerated as a background task. Now this secure shell, this I am not going in to detail of this protocol. But in this protocol there is need for these two kinds of levels of encryption. One is used on a per session basis you need to refresh your keys at the end of every session. And the other one is a long term key where for example you want to change the key. The short term key you can use the long term key for the purpose or you are verifying the username and password. For all those things you can use the long term key.

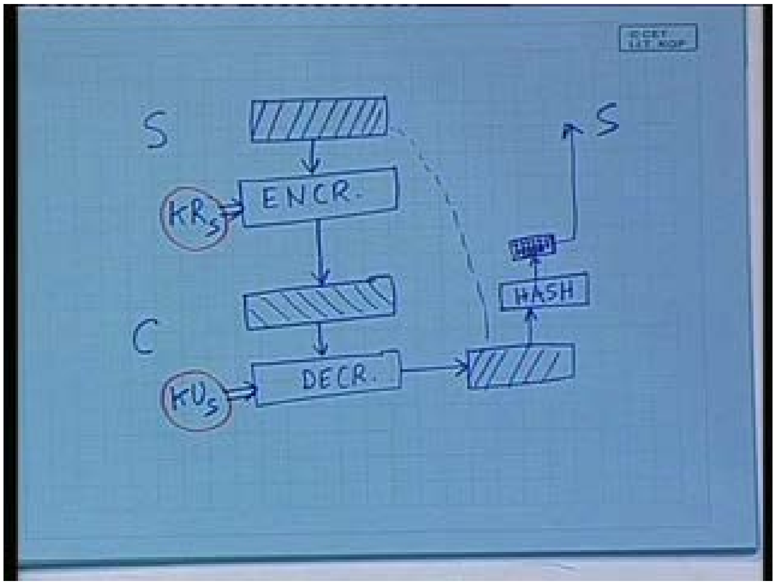
(Refer Slide Time: 37:54)



- Multiple authentication mechanisms
  - ➢Straight passwords (protected by SSH encryption).
  - ➢RSA based authentication.
    - Client decrypts a challenge from the server; returns the hash to the server.
  - ➢Plug-in mechanisms (biometrics, smartcard, etc.).

There are multiple authentication mechanisms support which is there. Some are built in, some you can plug in. Like you can have straight password based authentication. However the login name and password that you are typing they will be protected by SSH encryption. So whatever will be flowing over the network they will be the corresponding encrypted values. Secondly you can have RSA based authentication. See now you have an environment where public and private keys are shared by the two parties. Now we can also have authentication using RSA like this. Here the concept is based on some kind of a challenge. Challenge is nothing but some kind of a text or a data; a block of data which the server sends to the client. See it is like this, the server suppose I am the server I send you a block of data. So the block of data will be encrypted you can decrypt. It is like this.

(Refer Slide Time: 39:13)



Suppose here we have the server here we have the client. So the server has a block of data. This is any arbitrary data it can generate. This can be a random number also. This can be a random number also which is generated by the server. Now what the server will do? It will carry out an encryption process. It will encrypt this data and will send encrypted version to the client. So the client will be receiving this. So what the client will do, the client while this encryption was carried out by client's public key. So client would be able to decrypt it. So client will be using a decryption mechanism using its own public key to decrypt it. So whatever you get back here this must be same as this.

But now what happens is that, on this you run a hash and you get a small hash code. It is this hash code which is now transmitted back to the server. Now it is possible for the server to verify just by looking at the hash code whether the client was able to successfully decrypt the message or not. Now since we are talking about authentication typically the server will carry out encryption using its own public key and the client will carry out the decryption using the server's public key. So the process will be as follows. When this encryption is going on, this will be carried out by the private key of the server and when this decryption will be done this will be carried out by the public key of the server.

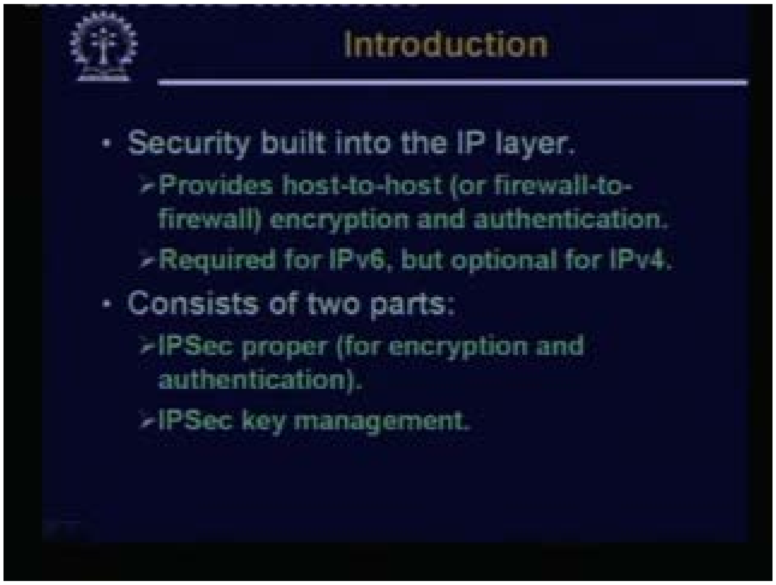
So recall when we have applications like authentication encryption is done by private key decryption is done by the public key. So we do this. Since client has access to the public key of the server you can do it. But if it was normal data transmission secrecy of it then the order of the keys are different. But since we are talking about authentication here, the order of the usage of the key should be like this. And of course it has some plug in mechanisms based on the state of the art techniques like biometrics, finger print identification, detection of the eyes and other thing smartcard. So there are a number of techniques which you can use as the plug in.

(Refer Slide Time: 42:21)

IP Security (IPSec)

Now let us talk about protocol called IP security, IPSec. This is also there as a standard. Some people do use it.

(Refer Slide Time: 42:40)

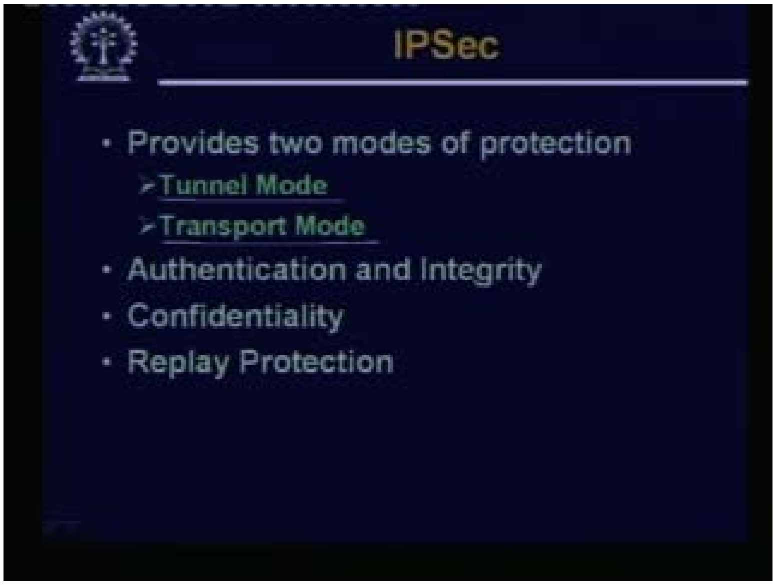


Introduction

- Security built into the IP layer.
  - Provides host-to-host (or firewall-to-firewall) encryption and authentication.
  - Required for IPv6, but optional for IPv4.
- Consists of two parts:
  - IPSec proper (for encryption and authentication).
  - IPSec key management.

Now IPSec is different. IPSec here some security is built in to the IP layer. Now if you recall all the other techniques they do something either at the TCP layer or above. But here at the IP layer level with respect to each IP packet that is flowing some kind of security is put in. Now there are a number of ways in which you can use this IPSec. You can use host to host encryption or firewall to firewall or router to router and also authentication. So this protocol can provide you encryption and authentication service between two hosts or computers or two routers or firewalls. If you are using IP version 4 then IPSec is optional you may or may not have. But if you have already migrated to IP version 6, then IPSec is an essential service. It consists broadly of two parts; one is the basic IPSec or the IPSec proper. This is used mainly for encryption and authentication. And the other part is IPSec key management; how to manage the keys.

(Refer Slide Time: 44:21)

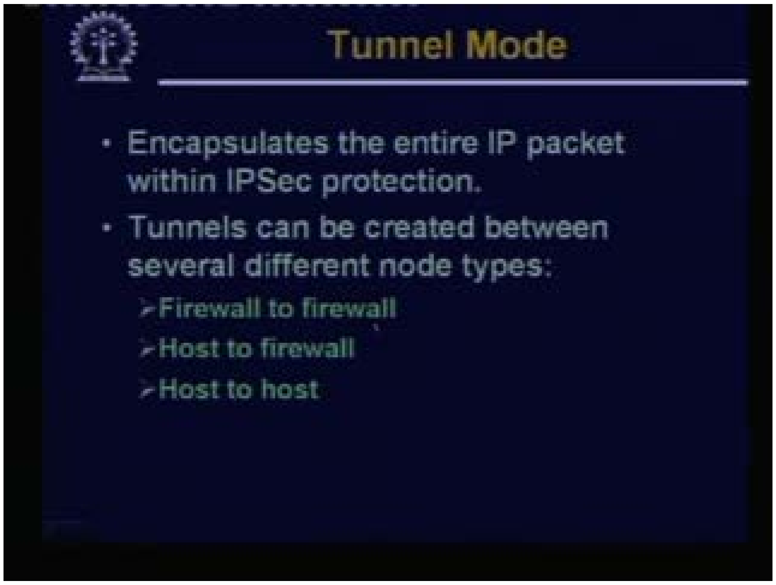


IPSec

- Provides two modes of protection
  - Tunnel Mode
  - Transport Mode
- Authentication and Integrity
- Confidentiality
- Replay Protection

Now IPSec provides two modes of protection. The first is called the tunnel mode; other is called the transport mode. We shall be briefly looking at this. Now in addition to this as I said it provides authentication and also integrity of the data. You cannot modify the data. It provides encryption, decryption, confidentiality, replay protection. Means it avoids or it prevents this kind of attack that an intruder is silently listening to our conversation and is simply replying or resending the packets at a later date. This kind of an attack is also taken care of by adding some kind of a sequence number to the packets which cannot be reproduced later. Because they itself are encrypted along with the data. The intruder cannot decrypt them.

(Refer Slide Time: 45:20)

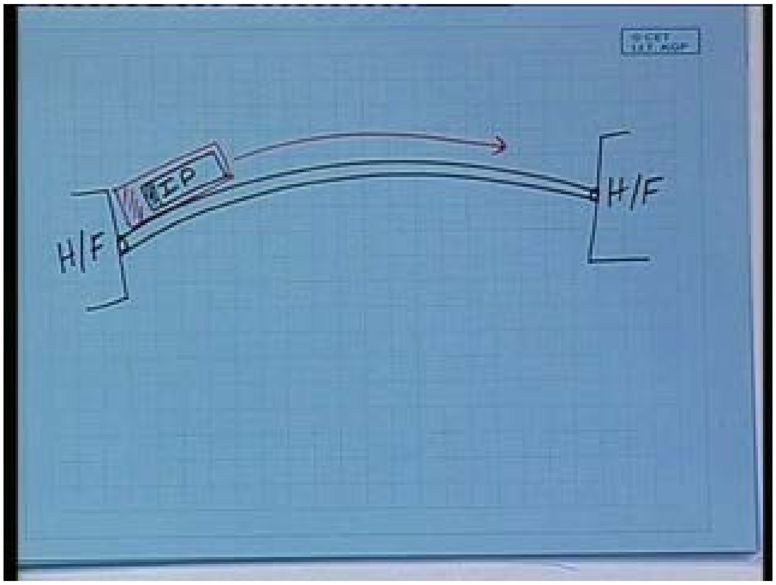


Tunnel Mode

- Encapsulates the entire IP packet within IPSec protection.
- Tunnels can be created between several different node types:
  - ➢Firewall to firewall
  - ➢Host to firewall
  - ➢Host to host

Now in the tunnel mode what you do is that the entire IP packet is encapsulated within IPSec tunnels can be created between several different node types. Firewall to firewall, host to firewall, host to host. See the concept is like this. If you are using say SSL, then finally you are using the IP layer for the delivery of packets. In the IP layer, in the IP header, you have the source IP address, destination IP address as usual. But here what you are seeing is likely different.

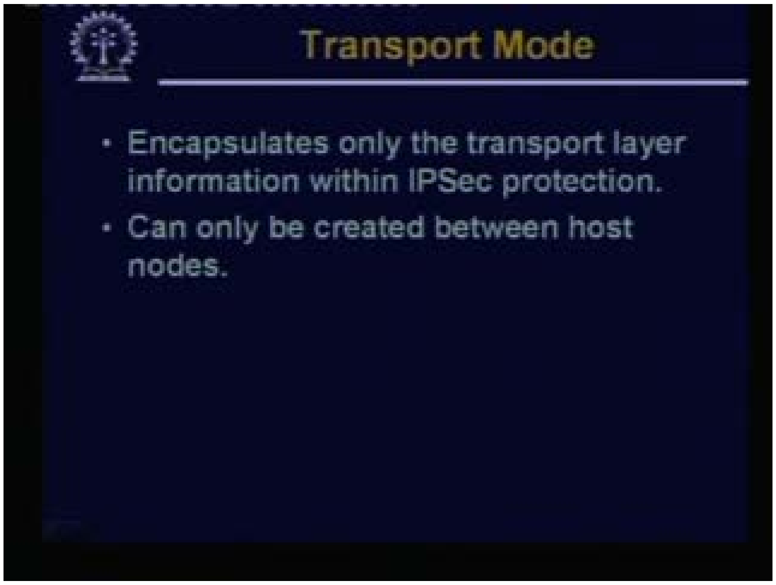
(Refer Slide Time: 46:09)



We have two nodes on two sides. Now these nodes can be either a host or a firewall on either side. Now here it says that if you have IPSec instead of sending the IP packets independently, you first create some kind of a connection between these two at the IP layer level itself. This is some kind of a tunnel connection and the entire IP packet. This is an IP packet complete with headers there is a header portion also. Here is a header portion also. This whole IP packet is encapsulated in IPSec. So IPSec will be putting in some more headers on top of it like this and it will send the whole thing over this tunnel to the other side.

What this means is that with respect to the IP packet, both the contents of the data and also the IP packet header both of them are protected and intruder cannot have any access to either of these two. Now if you recall earlier we have mentioned that for passive kind of an attack, an intruder can know what data you are sending. It can also know some statistics about the packets that are flowing. Now if you have this kind of a tunneling, then the statistics information will not be revealed to the intruder. Because statistics you can know only if you can read out the headers. What is the source address? What is the destination address, port numbers, etcetera. But if the entire thing is encapsulated within IPSec intruder will not be able to know exactly what is there. So this is what the tunnel mode is.

(Refer Slide Time: 48:04)

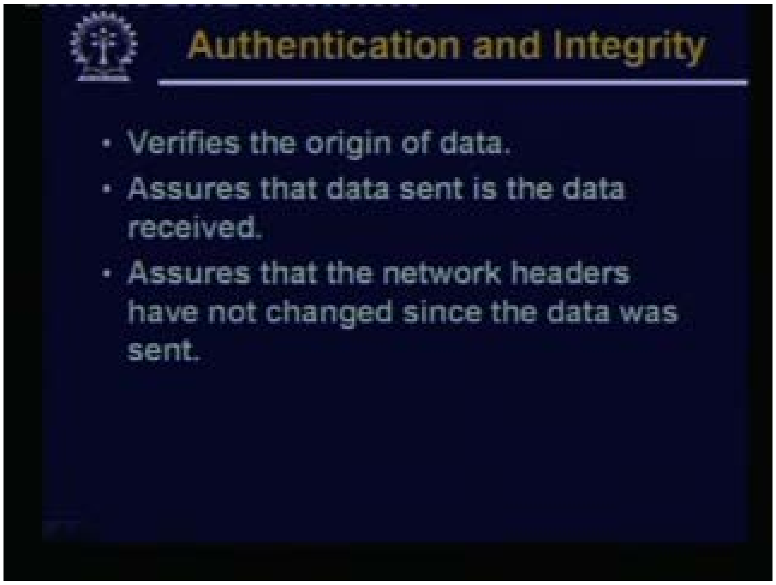


Transport Mode

- Encapsulates only the transport layer information within IPSec protection.
- Can only be created between host nodes.

The other one is the transport mode. Transport mode says that you encapsulate only the transport layer information, not the IP headers. This can only be created between host nodes because typically between two ordinary computers you cannot create that kind of a tunnel you need to specify the source and destination IP addresses. So you can encapsulate the transport layer pay load not the IP layer pay load. So this is the so called transport mode.

(Refer Slide Time: 48:40)

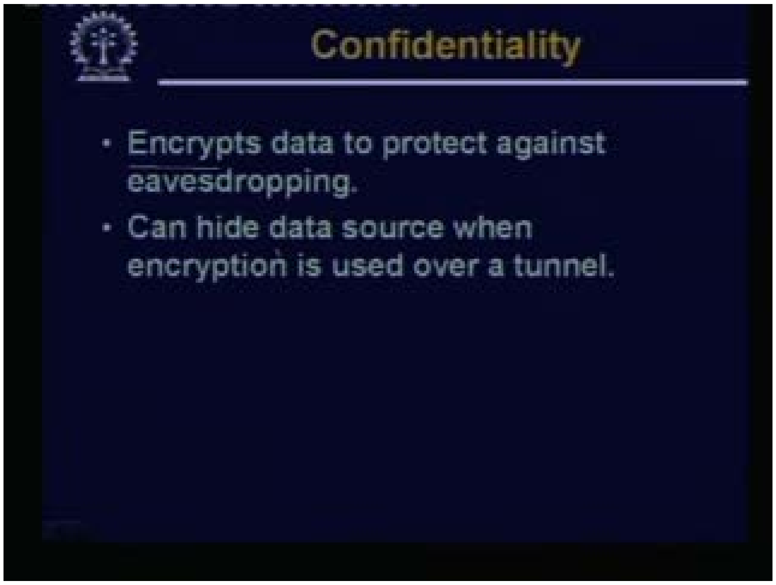


Authentication and Integrity

- Verifies the origin of data.
- Assures that data sent is the data received.
- Assures that the network headers have not changed since the data was sent.

Regarding authentication and integrity verification of the origin of data is there. So that it can be assured that what data you are sending is the data you are receiving and more over the network headers. Because of this encapsulation of the header also the network headers also you cannot modify. Because there are many kind of attacks where a packet is flowing intruder is changing the contents of some of the headers before letting or allowing the packets to flow. So these kinds of attacks are also prevented.

(Refer Slide Time: 49:17)

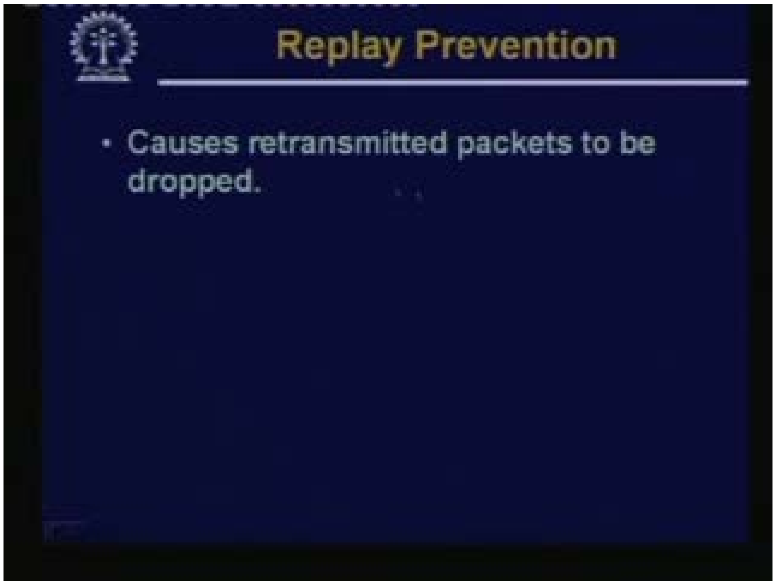


Confidentiality

- Encrypts data to protect against eavesdropping.
- Can hide data source when encryption is used over a tunnel.

Confidentiality is ensured by using encryption decryption. So since the IP packets including headers are encrypted as I said, so the data source and destination can also be hidden when you are using in the tunnel mode.

(Refer Slide Time: 49:40)

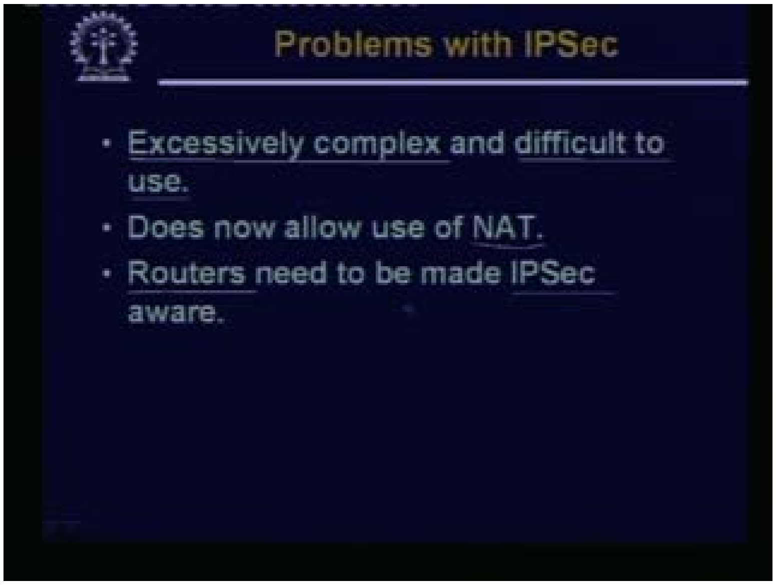


Replay Prevention

- Causes retransmitted packets to be dropped.

Replay prevention, I have mentioned I am not going in to detail how it is done. But by using some kind of sequence numbers which are also part of the encrypted pay load. The intruder will not be knowing what these sequence numbers are. So the intruder cannot play or regenerate the same packets at a later date because at the later date if he wants to do so, it must generate the correct sequence numbers. It does not know what is the last sequence number that was used.

(Refer Slide Time: 50:13)



Problems with IPSec

- Excessively complex and difficult to use.
- Does now allow use of NAT.
- Routers need to be made IPSec aware.

IPSec has a number of very good features. Everything seems to be very good, but the problem is that in order to incorporate so many things the protocol has become excessively complex and is rather difficult to use. Secondly nowadays if you look at the internet or the network infrastructure the use of network address transmitters are pretty wide spread. But in IPSec since you are creating tunnel and hiding the entire IP packets you can easily imagine that NATs will not work here. Because NAT is the basic low level hardware box through which all packets are flowing. But if you are hiding the IP addresses, how will NAT take a decision whether to send? How to receive? So NATS will not work under this changed IPSec scenario and also if you are using routers.

You need to use routers which are compatible with IPSec. Because IPSec needs some very special supports like tunneling as I said. So if the router does not have this kind of a support you cannot have IPSec. So IPSec is a very general and complex kind of a protocol which is good there are a number of features and facilities you can have. But the problem is that you need to gear up your network. So have to IPSec, if you have IPSec the plus point is that certain kinds of attacks which otherwise are possible, will not be possible anymore. Like gathering statistics about the servers that are used most heavily in a network for example. So if you do not disclose the information about the source and destination addresses this is something we will not be able to do very easily.

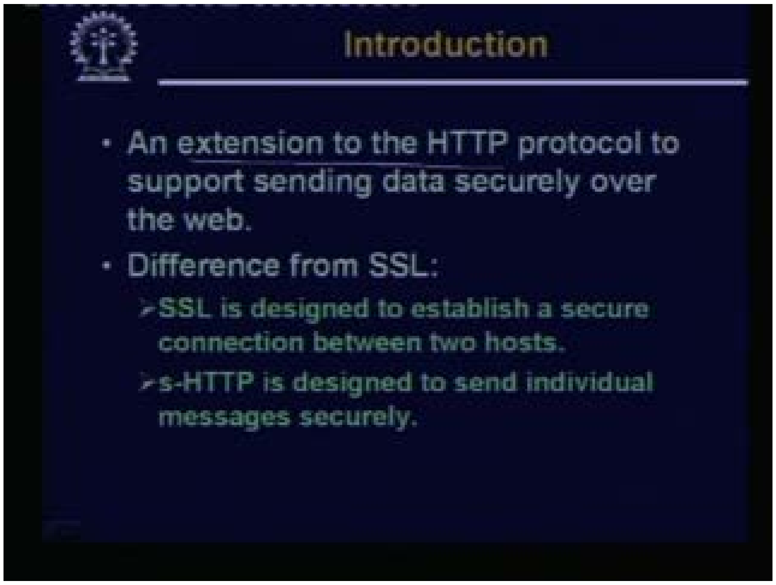
(Refer Slide Time: 52:12)

Secure HTTP (S-HTTP)

So the last class of protocol we would be talking about is with respect to the web protocol HTTP, secure HTTP or S-HTTP. Now s-HTTP is basically an extension to the HTTP protocol. Because in the HTTP protocol if you recall there are some basic comments like GET, PUT, which the client sends to the server and the server sends back my encapsulated response after that. But both these things are going in the clear. If an intruder wants to listen what is going and what is coming back, intruder can easily find out and can also change modify if it requires, if it wants.

So this s-HTTP this allows you to send data securely over the web. Now there are many scenarios where you may want this. The most common scenario which may come to your mind is while when you are carrying out some confidential transaction over the net like you are typing in your credit number, you are typing in your user name and password over the network. These are certain applications where you need that whatever you are typing on your browser should go to the other side in an encrypted fashion and this feature should be built into the HTTP protocol.

(Refer Slide Time: 53:52)

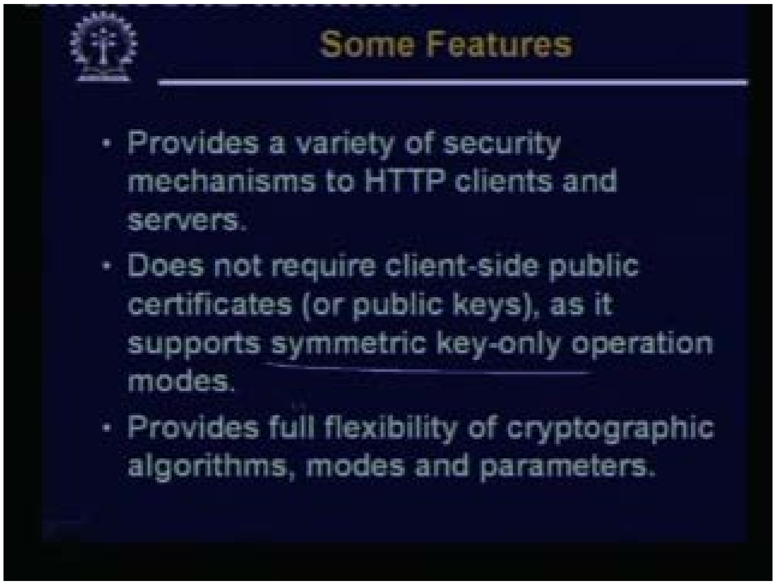


Introduction

- An extension to the HTTP protocol to support sending data securely over the web.
- Difference from SSL:
  - SSL is designed to establish a secure connection between two hosts.
  - s-HTTP is designed to send individual messages securely.

Now SSL you can argue also provides this kind of a facility. But the difference is as follows. In SSL you are establishing a secure connection between two hosts. And all packets which are flowing will get encrypted and decrypted. But s-HTTP is used on a per message basis. So every HTTP request to a sending they are encrypted and similarly the responses which come back they also come back encrypted. So this s-HTTP is used for a per message basis. But SSL is used for the entire duration of the session. All the messages which flow will be using the same keys, same set of protocols and parameters to send and receive data.

(Refer Slide Time: 54:44)

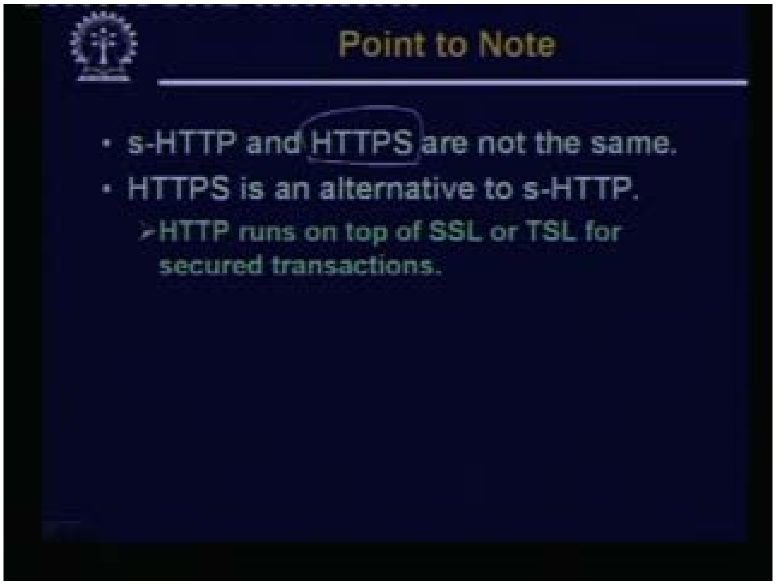


Some Features

- Provides a variety of security mechanisms to HTTP clients and servers.
- Does not require client-side public certificates (or public keys), as it supports symmetric key-only operation modes.
- Provides full flexibility of cryptographic algorithms, modes and parameters.

Some of the features of s-HTTP provide a variety of security mechanisms for the clients and servers. This does not require client side public certificates because here we can use symmetric key only operation modes also. Here again the client and server can negotiate. You have full flexibility of which algorithm to use and the parameters key sizes of the key which key to use and so on.

(Refer Slide Time: 55:17)

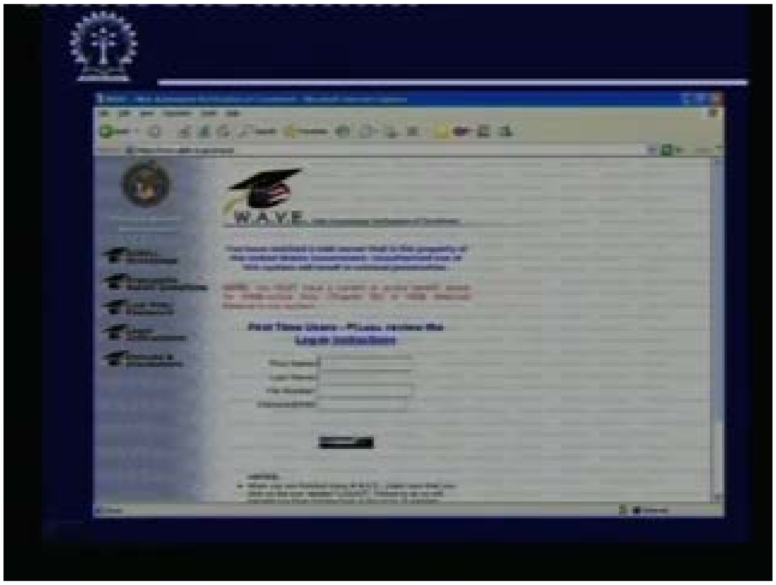


Point to Note

- s-HTTP and HTTPS are not the same.
- HTTPS is an alternative to s-HTTP.
  - HTTP runs on top of SSL or TSL for secured transactions.

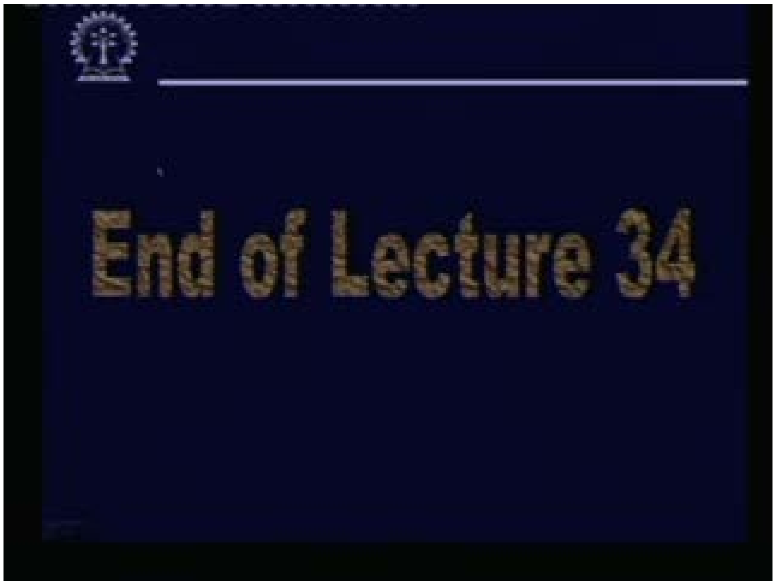
Now there is one point you should remember s-HTTP is there of course. But if you look at the web sites nowadays you will find that HTTPS which is an alternate. This is more widely available and used but s-HTTP and HTTPS are not really the same. HTTPS is basically the version of HTTP that is running on top of SSL; that is HTTPS. But s-HTTP is a separate security protocol itself. HTTPS is something which is running on top of secure socket layer.

(Refer Slide Time: 56:01)



Here I have an example website. See nowadays you see websites like this where you are asked to type in or key in some information which is confidential. But if you look at the URL or the address you will find that the address starts with HTTPS and not HTTP. Just by looking at the first few characters of the URL, if you see that its start with HTTPS you will know that this is a HTTPS transaction that is going on and everything that will go will go encrypt. So you can be sure about the transaction. Even if you type in something secure no one will be able to know it. So with this we can come to the end of the present lecture.

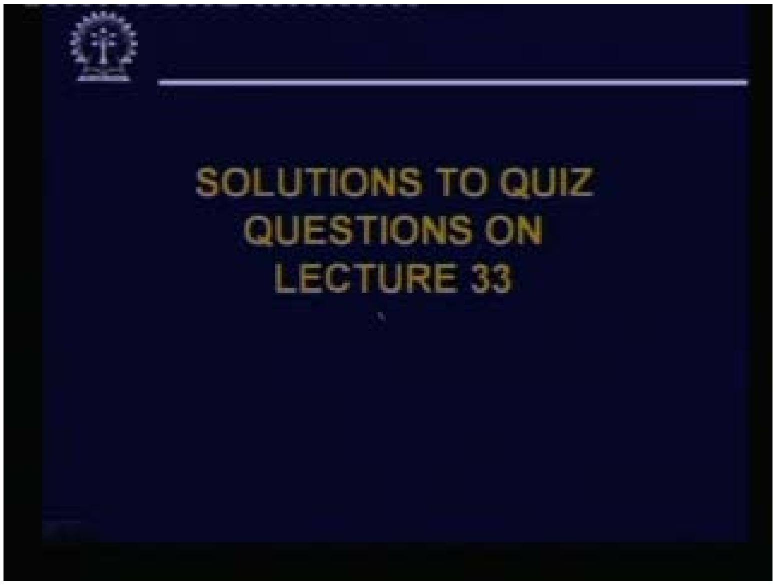
(Refer Slide Time: 56:54)



End of Lecture 34

Lecture number 34.

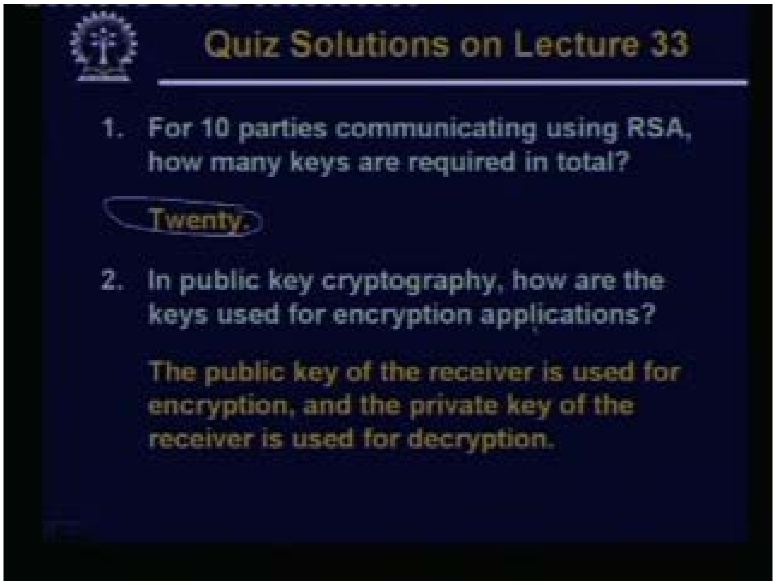
(Refer Slide Time: 56:56)



SOLUTIONS TO QUIZ QUESTIONS ON LECTURE 33

So first we quickly look at the solutions to quiz questions of previous lecture.

(Refer Slide Time: 57:01)



Quiz Solutions on Lecture 33

1. For 10 parties communicating using RSA, how many keys are required in total?

   Twenty.

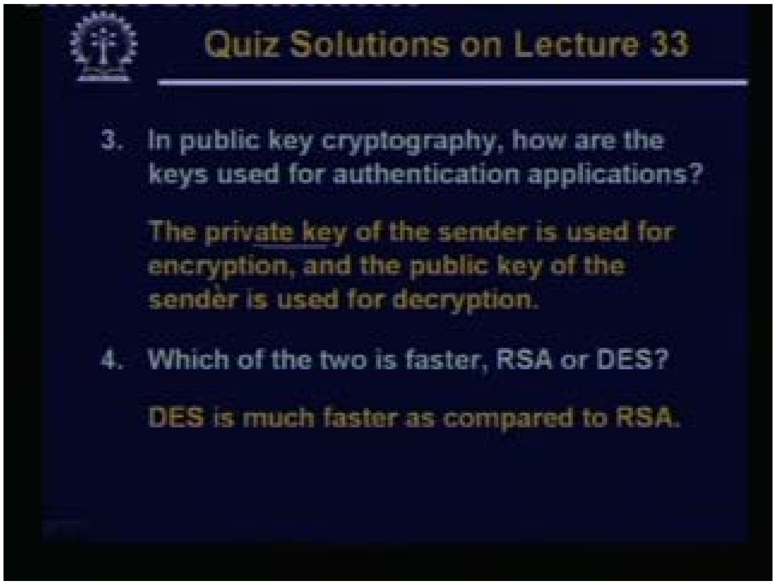
2. In public key cryptography, how are the keys used for encryption applications?

   The public key of the receiver is used for encryption, and the private key of the receiver is used for decryption.

The first question was for 10 parties communicating using RSA. How many keys are required?
So I said for each party you need two keys; one public and one private. So it will be 20.
In public key cryptography how are the keys used for encryption applications?

For encryption if you recall the public key of the receiver is used for encryption and the private key of the receiver is used for decryption.

(Refer Slide Time: 57:35)



Quiz Solutions on Lecture 33

3. In public key cryptography, how are the keys used for authentication applications?

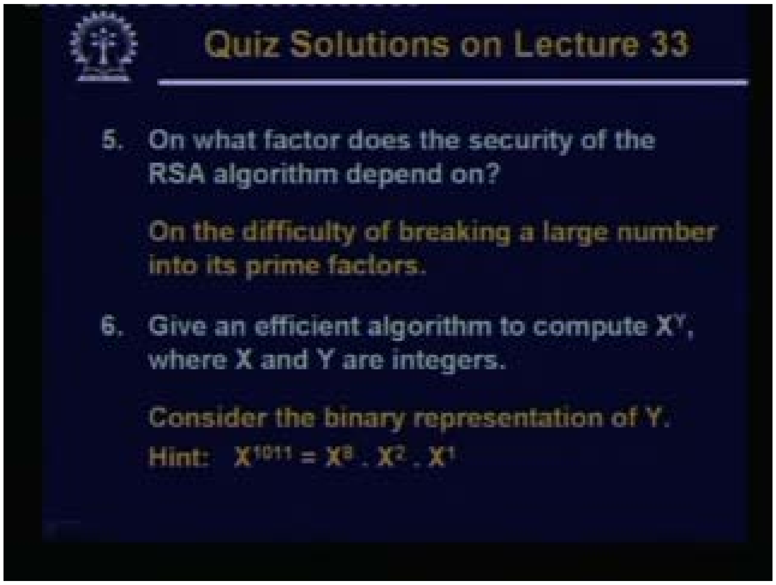
   The private key of the sender is used for encryption, and the public key of the sender is used for decryption.

4. Which of the two is faster, RSA or DES?

   DES is much faster as compared to RSA.

In public key cryptography how are the keys used for authentication application?
So here the usages are reversed. The private key of the sender is used for encryption and the public key of the sender is used for decryption.
Which of the two is faster RSA or DES?
This we have mentioned DES is much faster as compared to RSA at least by 100 times or even more.

(Refer Slide Time: 58:04)



Quiz Solutions on Lecture 33

5. On what factor does the security of the RSA algorithm depend on?

   On the difficulty of breaking a large number into its prime factors.

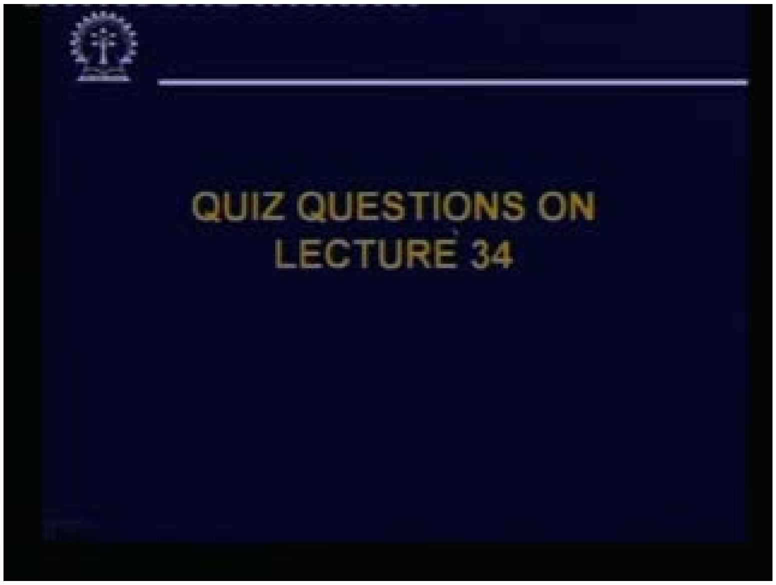
6. Give an efficient algorithm to compute X^Y, where X and Y are integers.

   Consider the binary representation of Y.

   Hint: $X^{1011} = X^8 . X^2 . X^1$

On what factor does the security of the RSA algorithm depends on?
It depends on the difficulty of factoring or breaking a large number into its prime factors.
Give an efficient algorithm to computer X to the power Y where these are integers.
Now the outline is as follows. If you break the number Y in binary then you can write it like this.

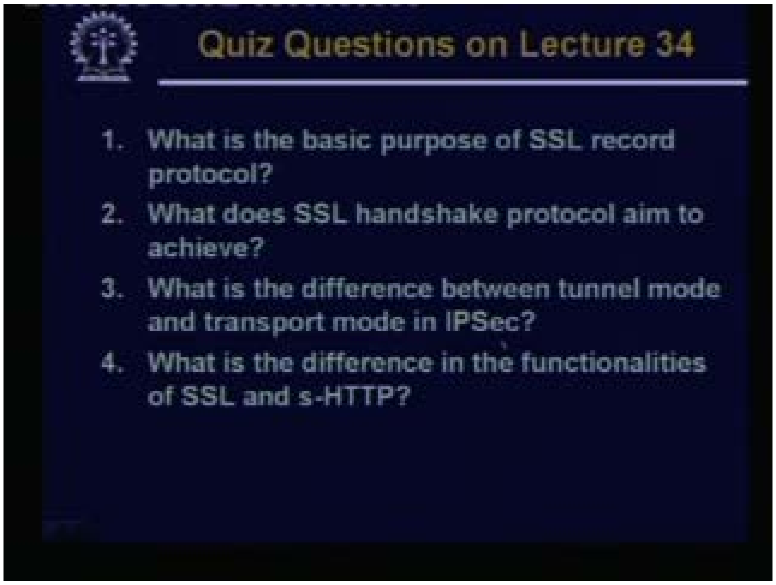
(Refer Slide Time: 58:34)



QUIZ QUESTIONS ON LECTURE 34

These are some of the quiz questions from this lecture.

(Refer Slide Time: 58:39)



Quiz Questions on Lecture 34

1. What is the basic purpose of SSL record protocol?
2. What does SSL handshake protocol aim to achieve?
3. What is the difference between tunnel mode and transport mode in IPSec?
4. What is the difference in the functionalities of SSL and s-HTTP?

What is the basic purpose of SSL record protocol?
What does SSL handshake protocol aim to achieve?
What is the difference between tunnel mode and transport mode in IPSec ?
What is the difference in the functionalities of SSL and s-HTTP?
So with this we come to the end of today’s lecture. Thank you.