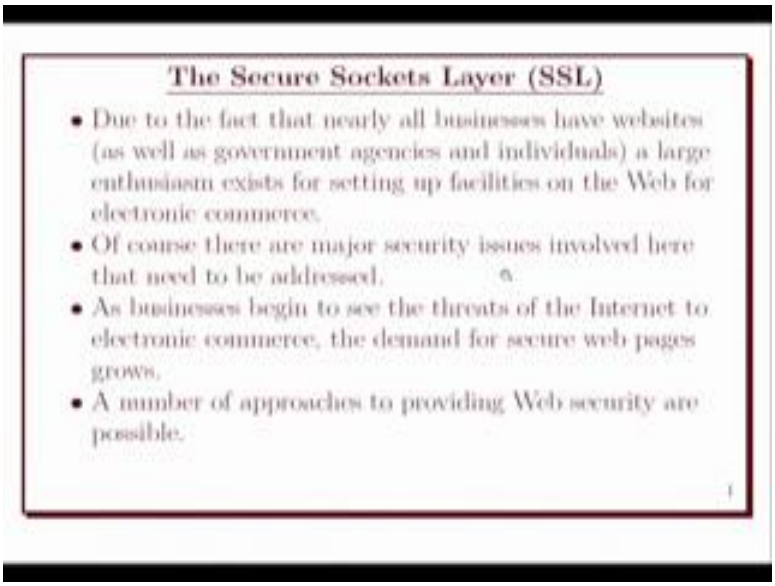
Internetwork Security
Prof. Sourav Mukhopadyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 59
The Secure Sockets Layer

So, we talk about Secured Socket Layer. This is basically the scheme to provide the security for web based application like we have, nowadays we have we are very much depending on the web based e-commerce. So, we purchase if we have a business module, build a website and I welcome all the people to buy the items through my website. And many thing like Facebook we use the website, so even this Flipkart so everything. So, nowadays we are mostly depending on the website for our online purchase e-commerce. So, we need to look over the security for this area, so web security.

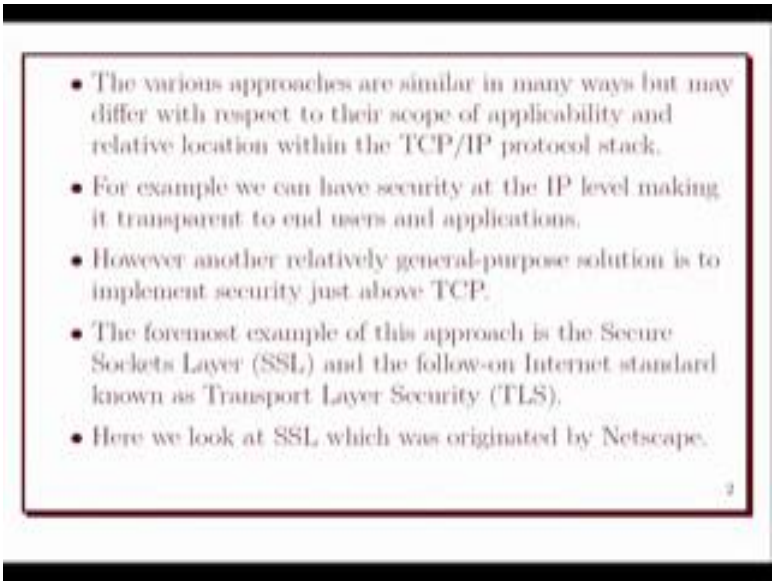
(Refer Slide Time: 01:05)



So, due to the fact that nearly all business are websites even it is government or individuals. So, we need to setup the facilities for web to for the e-commerce also. Of course, there are major security issues involved here that need to be addressed, because we just access the website by without knowing whether it is secure or not. So, we need to have that knowledge should be there; I mean that confidence should be there that the website we are accessing is the secure website. So, as the business is begin to see the

threats of the internet to e-commerce the demand of the secure web page grows. So, this is the topic is to provide the web security.

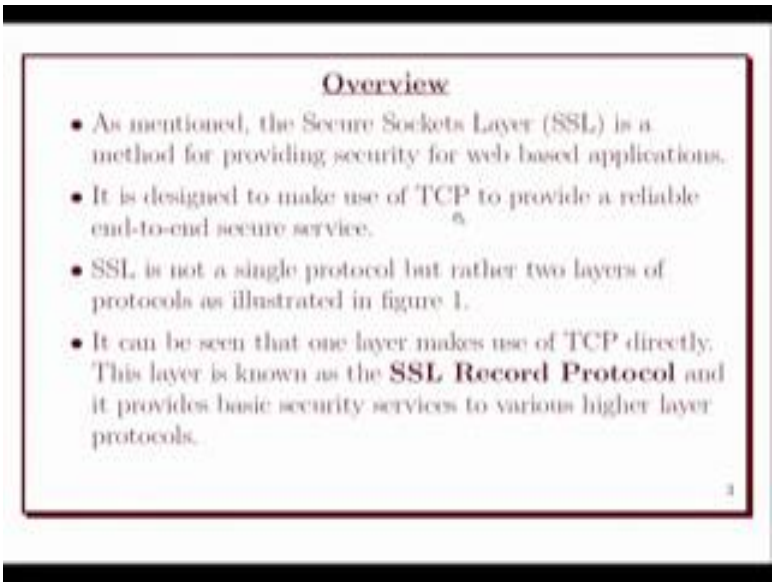
(Refer Slide Time: 02:03)



So, there are many approaches are there to providing this web security. So, few of the approaches is basically, so we have a TCP IP stack. So, basically to a IP stack and then we have a TCP stack protocol in the network protocol. Now, various approaches are similar in many ways, but may differ with respect to their scope of applicability and relative location within the TCP IP protocol stack. For example, we can have security at the IP level making transparent to the end user and applications. So, inside that is, inside the IP protocol IP layer, we can have a security we can design a security protocol layer to give the security for the users and applications.

And another approach could be which is relatively general purpose solution is to implement security just above the TCP layer. So, we have IP layer then TCP layer, so just above the TCP layer if we can have another layer of security, so that is one example of that is SSL layer. So, one example of this approach is the secret sockets layer, and which follow-on Internet Standard known as transport layer security, this is Internet Standard TLS. And here we look at the in this socket we look at the SSL which was designed by Netscape. So, this is the overview of this SSL protocol or SSL layer.

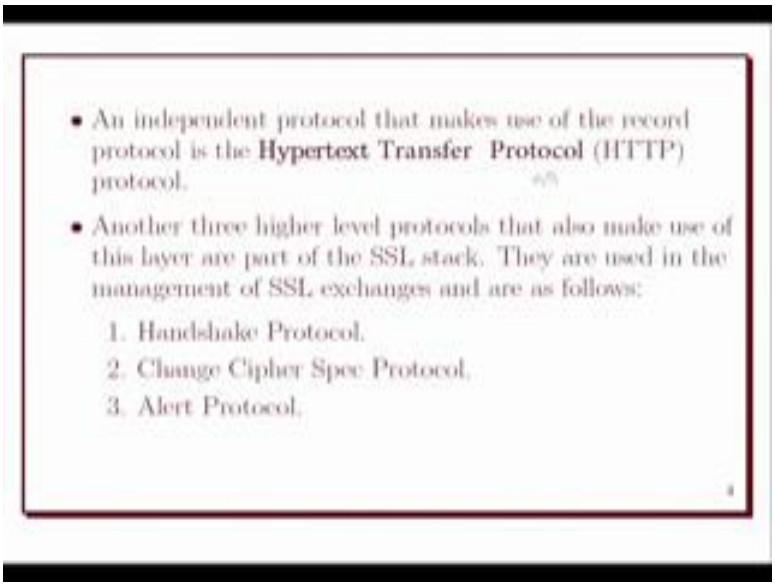
(Refer Slide Time: 03:35)



So, as mentioned this is a SSL is a method of providing security for web based applications. It is designed to make use of TCP to provide the reliable end-to-end secure service. And SSL is not a single protocol, but rather it has two layer. So, it is basically SSL is basically on top of the TCP layer and SSL is also consist of two layer; one is SSL record protocol and another layer is top layer and this has four part. We will come to that, four part is basically having this SSL handshake protocol, SSL change cipher specification protocol, SSL alert protocol and http protocol, we will talk about details on this protocol.

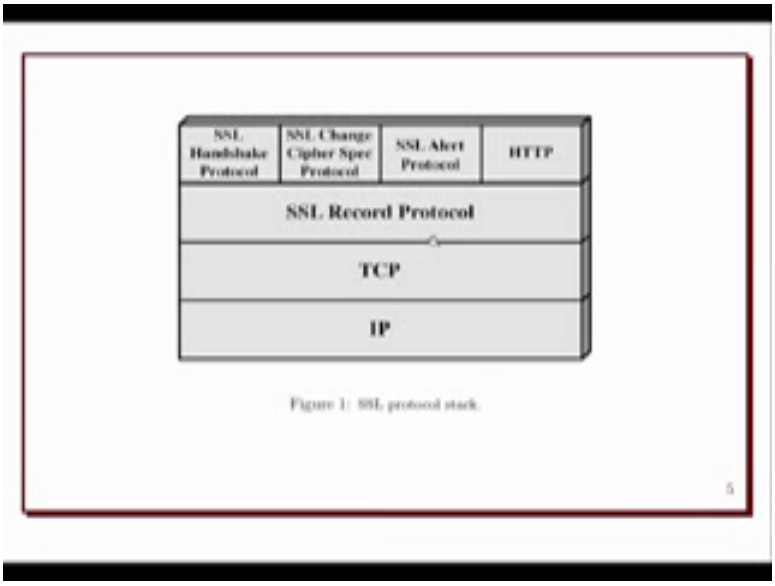
So, it has basically two layers. So, it can be seen that one layer make use of the TCP directly, because this is just on the top of the TCP layer. And this layer is known as SSL record protocol, and it provides basic security services to various higher level protocol.

(Refer Slide Time: 04:55)



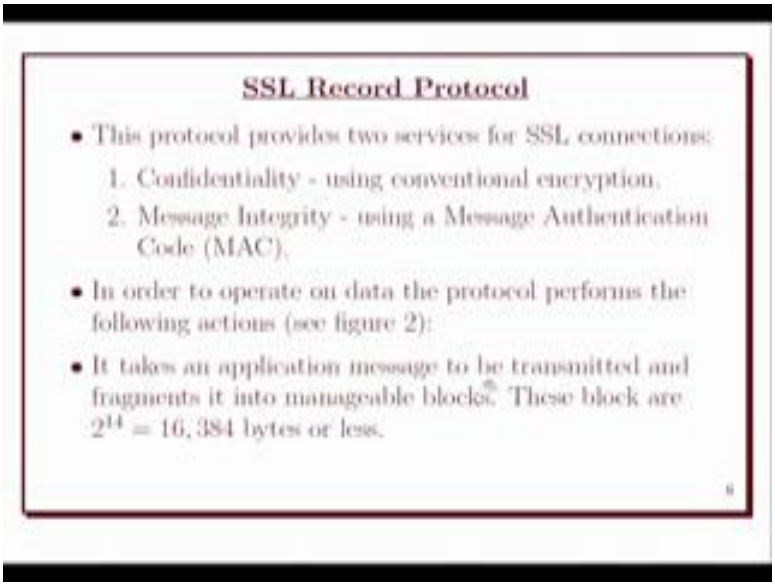
And on top of that, we have basically four protocols; one is this http protocol, and there are three mode protocol, handshake protocol, change cipher specification protocol and alert protocol. So, this is just on the top of the SSL record protocols. So, this is the part this is the top-level protocol, top-level layer of the SSL. So, SSL have two layers; one is SSL record layer; and in the top, we have four layers. So, these are the four protocol in the top-layer.

(Refer Slide Time: 05:36)



So, this is the structure of this is the SSL protocol stack. So, we have a TCP protocol, we have IP stack, we have TCP stack on top of TCP stack we have this SSL layer. So, SSL layer has two parts; it has two layers basically one is it has two stacks, one is SSL record protocol that is just on the top of the TCP layer. And another one is the top layer which is having four protocols; one is SSL handshake protocol, SSL change cipher specification protocol, SSL alert protocol and http protocol, and we will talk about details of this protocol.

(Refer Slide Time: 06:22)

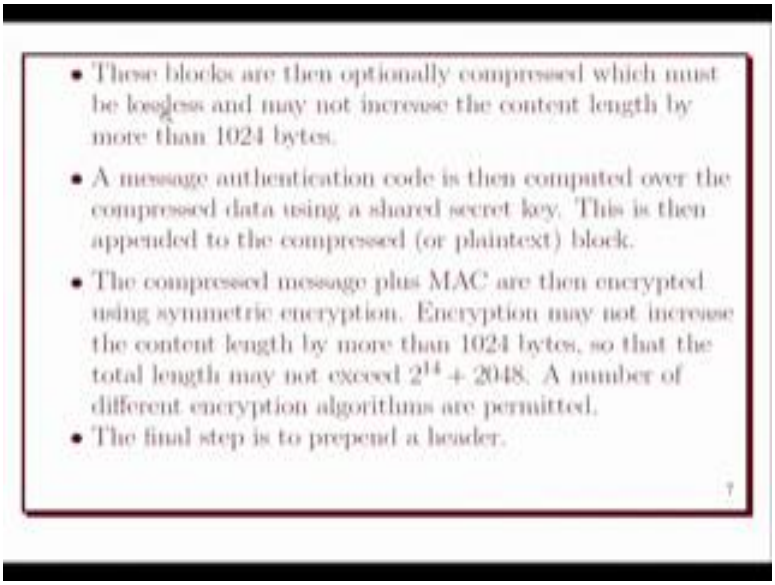


So, let us talk about SSL record protocol. So, what is the job of SSL record protocol? So, this is basically have two main services provides by this SSL record protocol or this confidentiality, another one is message integrity. So, confidentiality usually done by using the conventional encryption primitives encryption algorithm that use in this service which is called confidentiality. And we use the message authentication code or MAC for this message integrity.

So, we will talk about this in more details in figure 2. So, what we are doing? So in order to operate on the data this protocol performs the following actions. So, it takes the application message that is the data to be transmitted and the fragments, so it is blocked. So, its divides into the fragments blocks of fix size; and each block is maximum size is 2 to the power 14. So, basically we have the message or the data which to be transmitted

and this data break into the blocks or the fragments; and each block has size maximum 2 to the power 14 bytes, so that means 16,384 bytes or less. So, this is the first step.

(Refer Slide Time: 07:59)



And then after that each block is then compressed, this is optional each block is compressed which must be lossless, because this compression loss must not be lossy compression, and may not increase the content length by more than 1024. So, we have a size of 16, 34; from here, we compress this is optional compress this to a 1024 bytes. So, this is a compress operation we are doing on the each of this block. So, we have a message which needs to be sent to this data. So, we break it into the blocks or fragment; and each fragment of size 2 to the 14 bytes, and then we apply this is optional we apply the compression function on this which must not be lossy compression, this is lossless compression and we make it into less than 1024 bytes.

Now, we just apply the message authentication code or MAC. And then MAC is computed over the compressed data using the shared secret key. So, and this is then appended to the compressed block. So, we just break it into the blocks and then we just compress, and then we have a message this is the message, now this message we apply a MAC. And to apply the MAC, what we can do? We can take some of the bits say this is some of the bytes we can take. So, this is 1024 bytes, among this, we can take some of the bytes and then we can have the encryption that is using the secret key, and that is the MAC function that we can encrypt using the secret key. And that encrypted on the partial

in message is called a MAC. And this MAC value we then append to the plaintext at the end. So, basically we are adding the MAC.

So, this the compressed message plus MAC are then encrypted. So, after that, we have the compressed message and we append the MAC and then we have to encrypt this using the symmetric key encryption. The encryption may not increase the content length by more than 1024 bytes, so that the total length may not exists 2 to the power 14 plus 2048. The number of different encryption algorithm are performed, because this is a symmetric key encryption. So, after appending the MAC on the message, we just apply some encryption algorithm we can use this is symmetric key encryption we can use DES. So, we use the standard encryption technique, because DES we know is breakable. So, we can use AES or some function of AES or triple layer, so that is the standard encryption algorithm we will use. So, different encryption algorithm are permitted.

And final step we have to prepend the header. So, we got the encrypted that things, and final step we prepend the header.

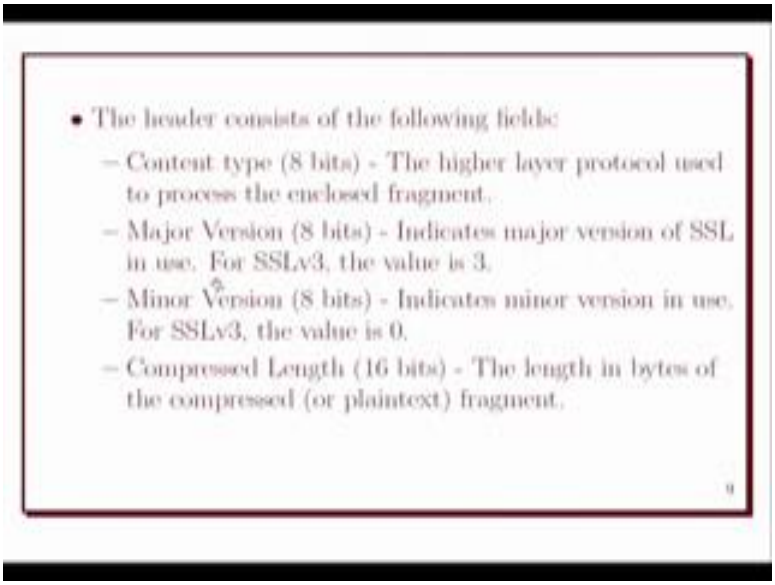
(Refer Slide Time: 11:25)

So, this is the general structure. This is the SSL record protocol operation. So, we have given a data or the message which need to be send. So, this message we break it into the blocks, so that each block will be maximum size 2 to the power 14 bytes. So, these are the fragments. So, each fragments we optionally so compress into we apply some compression lossless compression function, and we compress this blocks into maximum

1024 byte. And then we can so this is a message. So, we can take some of the bytes and we apply the some encryption algorithm, and we compute this using the secret key, we compute this MAC from this. MAC is authentication code.

And this MAC we are adding at the end of this compress message and then this is the message plus MAC, compress message plus MAC. And then these we encrypt using a symmetric key encryption; it we can use the AES or some triple layers, some known symmetric key encryption. And then after that we need to put prepend the header. So, we append the SSL record header. So, the header consists of some field, we will talk about that. So, this is the header. So, this is the operation we are doing by the SSL record protocol. So, we append the header with this encrypted message.

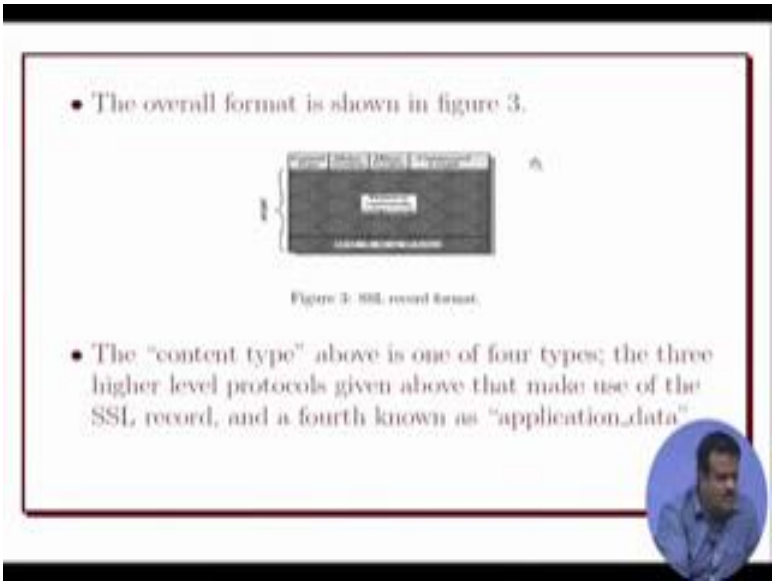
(Refer Slide Time: 13:05)



Now, this is the structure of the header; header consists of the following field. The first field is content type, which is basically 8 bits. So, we need 8 bit to represent this. So, the higher level protocol is used to process the enclosed fragment; and then another 8 bits used to indicate the major version. So, this indicate major version of SSL in use. For example, if the version of SSL is used SSLv3 then the value of this is 3. So, this 3 will be putted in this field the major version field. And also we use the 8 bits to indicate the minor version. So, if the SSLv3 then value is 0, then this 0 will be putted in the minor version, and that is also 8 bits.

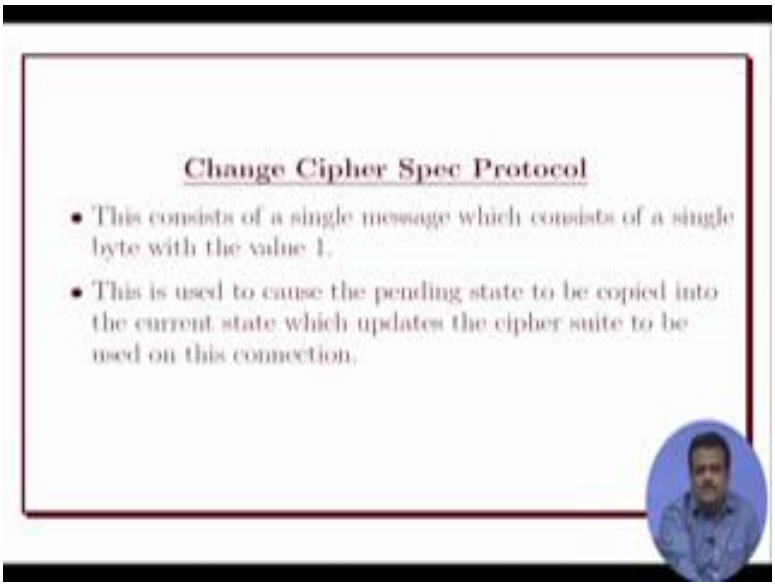
And then the composed length which is 16 bits, so this is the length and bytes of the compressed or the plaintext fragment. So, this length also we need to maintain in the header. So, header basically consist of this four field content type, which is basically 8 bit and then the major version, minor version and the length after compression. So, this four information has to be kept in the header. So, total how many 16 plus 16 plus 8 then plus 16, so this is the size these many bits are in the header. So, these we have to prepend in the encrypted message plus MAC. These we have to append here. So, this field this is the appended the header P appended. So, this field has four field like this content type, major version, minor version, compressed length, length of the compress message.

(Refer Slide Time: 15:17)



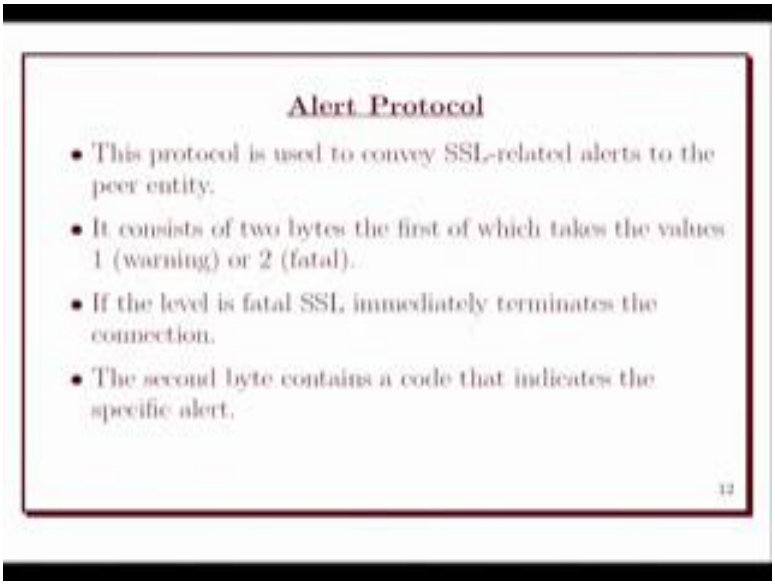
So, this is the scenario, so overall format this one is the here. So, this is our message and this is the compress message, and this compress message we then compute the MAC. And this MAC we append with is message. And then after that adding after adding the MAC, we encrypt this using the symmetric key encryption and then we prepend this type basically content type. We prepend this header and header has basically four field as we discussed content type, major version, minor version and the length of this compress length. So, this is the overall structure of this SSL record format. And the content type above is one of the four types; the three higher level protocol given above that make use of the SSL record, and fourth known as application data.

(Refer Slide Time: 16:28)



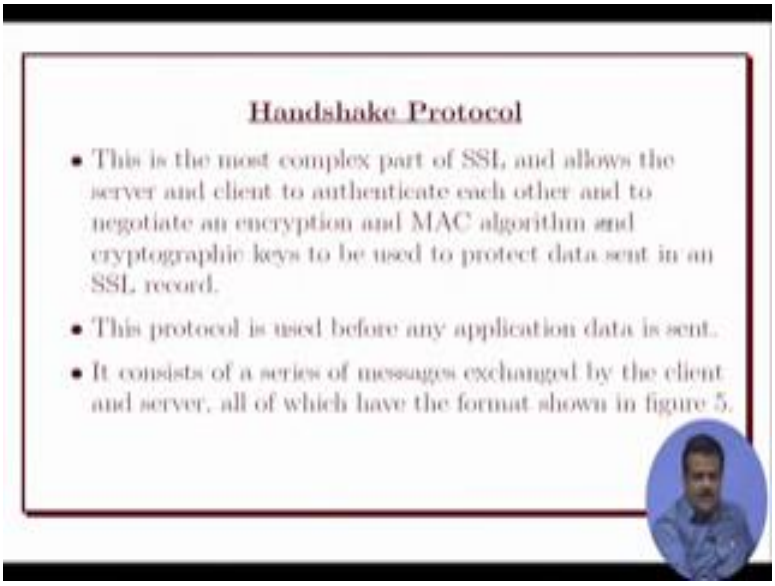
So, now we talk about the top-level. So, after this top-level protocols like we have so we already, so let us go back to this. So, this we have discussed. Now, let us talk about top-level protocol like what we are doing here. So, this is the change ciphertext specification protocol this consist of a single message, which consist of a single byte with the value 1. This is used to cause the pending state to be copied into the current state which updates the cipher suite to be use for this connection.

(Refer Slide Time: 17:09)



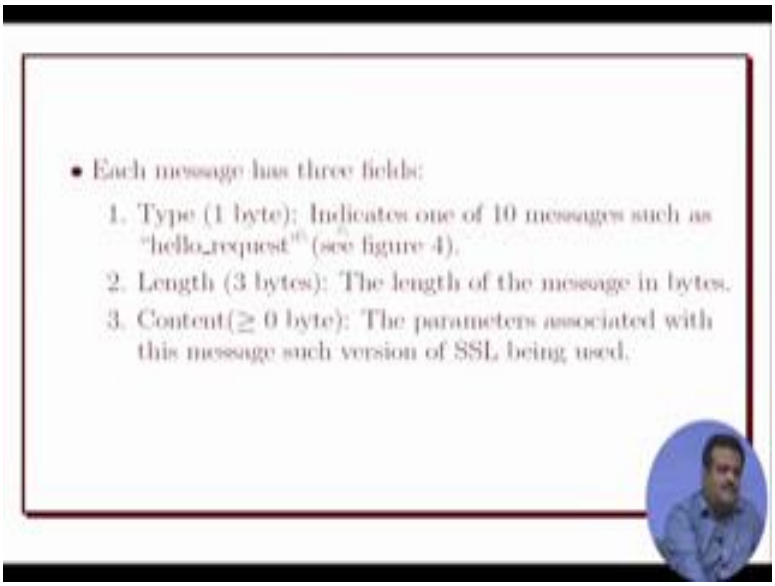
And then alert protocol. This protocol is used to convey the SSL related alert to the peer entity. Like it consist of two bytes, the first of which takes the value 1; 1 is for warning. So, it will set 1 value 1, if it is warning; and it will set the value 2, if it is fatal. Now, if the level is fatal then SSL immediately terminates the connection, because there is a danger. So, it is a fatal signal. So, then the SSL, so this is the alert protocol, alert protocol use the two bytes which takes the value 1 or 2. So, 1 means just a warning, but if it is setting the value 2, then it is a fatal signal. So, if the level is fatal SSL immediately terminates the connection. And the second byte contains codes that indicate the specific alert.

(Refer Slide Time: 18:14)



So, now we talk about handshake protocol. This is the most complex part of the SSL and allows the server and the client to authenticate each other. So, this is authentication part. So, we have this server and client has to authenticate each other by sharing some messages. So, this is few interaction will be done between them between sever and the client to authenticate each other, and to negotiate an encryption and the MAC algorithm and the cryptographic key to be used to protect the data sent in an SSL record. This protocol used before any application data is sent. It consists of series of message exchanged by the client and the server. So, we will talk about this. So, how this is happening? So, it has basically so this given in figure five.

(Refer Slide Time: 19:14)



So, each message has three filed; one is the type of the message, for this we need 1 byte to represent which indicate the 10 messages. So, there are 10 type of message which is shown in the next figure.

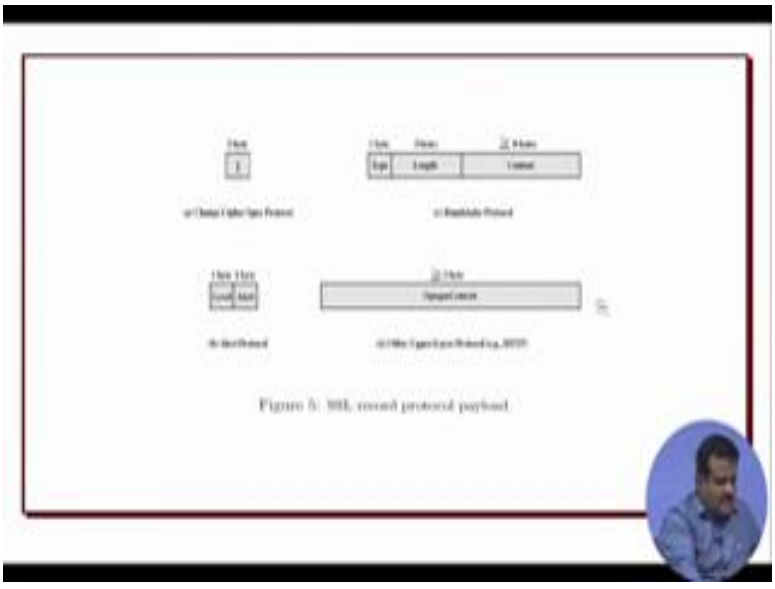
(Refer Slide Time: 19:29)

So, this is the 10 type of message. So, the message type may be hello request. So, client hello, then the server hello, certificate, server key exchange, certificate request, server done, certificate verify, client key exchange, finished. So, these are the message type. So, these are the message type. Now, these are the parameter for hello request there is no

need to have a parameter for client hello, the version, random, session id, so which session we are in. So, those has those are the parameter. So, compression method which we to be used compression method, so server hello, so here also version, random or session id, compression method, and certificate, so certificate means we should have some standard certificate, so that is the standard x dot 504 v 3, so this is chain of x certificate. And then the server key exchange.

So, here parameters are basically parameters and the signature and certificate request basically type authorities and server done parameter. Server verify is basically the parameter is the signature and the client key exchange is basically parameter and the signature and finish basically the hash value. So, this is basically 10 message type. So, these are the type of message; and for this, we need to have a few field, which is 1 bit to indicate this one of the time. And then the length, length is basically 3 bytes, length of the message in bytes. So, this is the length; length consists of two have the length use the 3 bytes and that will store the length of the message in byte. And the content the parameter associated with the message such as version of the SSL being used. So, this we have seen.

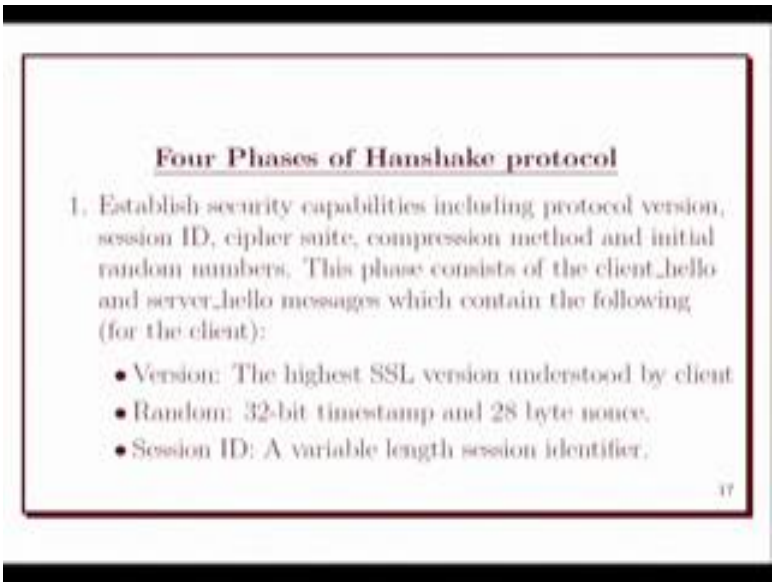
(Refer Slide Time: 21:32)



And this is the change cipher specification protocol, handshake protocol, so we have type, length and content. So, this is the type means, one of the type of the message like we have ten type. So, this is one of the type. So, to indicate that we use this field, this is 1

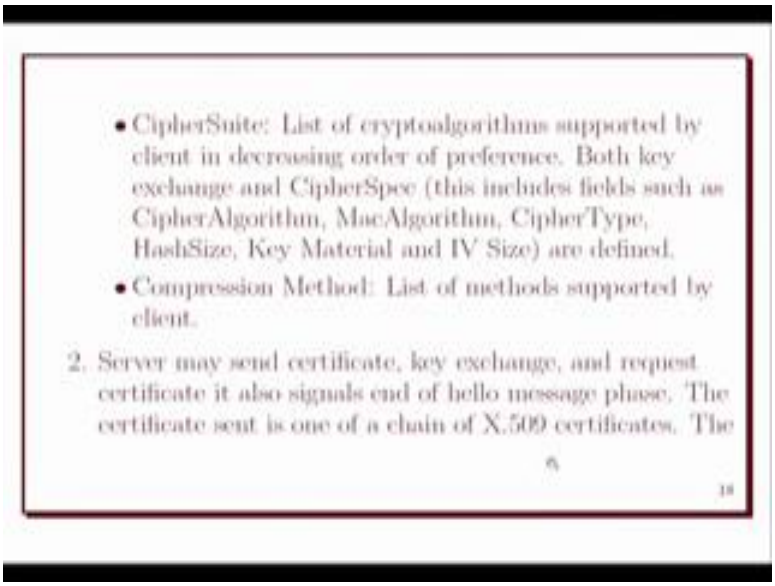
byte, and then we use this field to indicate the length of the message and this is the message. So, this is the handshaking protocol. And then we have a alert protocol in the protocol we have the label which is use 1 byte which is either warring or fatal and then we have a sorry this is the alert which is use 1 byte which is either warning or fatal. If it is fatal, then SSL terminate the connection. And then we have a http protocol, which is the other above level protocol, so which is basically used so more than 1 byte.

(Refer Slide Time: 22:32)



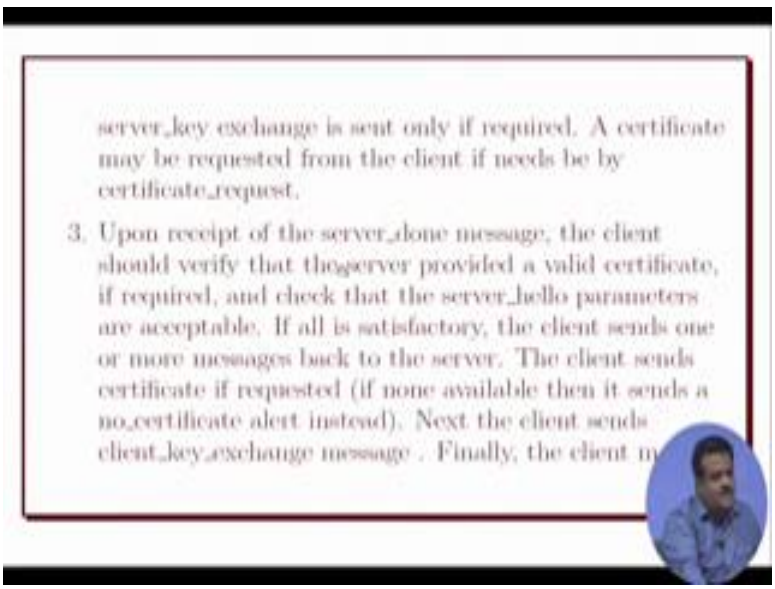
So, there are four phase let us talk about more details of the handshaking protocol. This has basically four phase. This is basically to authenticated between the client and the server. So, this is the, authenticate between so client will send some message, then server will response again client will send. So, in this four step, there are four step in this four step four phase these this handshake protocol will be working. So, this is establish security capacity this protocol establish security capacity including the protocol version session id, cipher suite, compression method and initial random numbers. This phase consist of client hello and cipher hello message, which contain the following for the client. So, version the highest SSL version understood by the client. Random, 32 bit timestamp and 28 nonce. These are used as a random numbers and the session ID, a variable length session identifier.

(Refer Slide Time: 23:38)



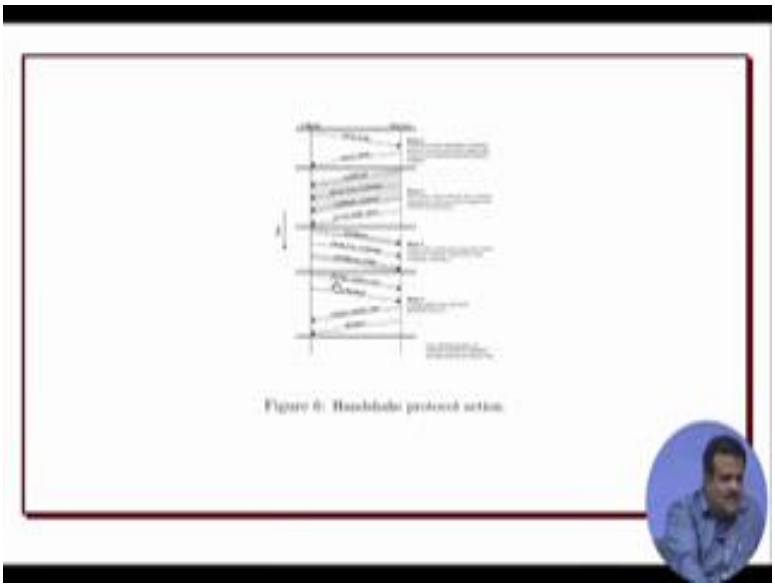
CipherSuite: list of the cryptographic support by the client to decreasing order of preference. Both key are exchange and the cipher specification this includes field such as CipherAlgorithm, MacAlgorithm, CipherType, Hash Size, KeyMaterial and IV Size - IV we can use for this, this is IV means initialization vector; are defined. So, compression method, so list of the method sup supported by the client. And this second phase is server may send certificate, key exchange and request certificate is also signal end of the hello message phase. The certificate sent is one of the chain which is use this certificate standard x 509. This is the standard of the certificate. So, this standard may be used.

(Refer Slide Time: 24:37)



Then the server key exchange is sent only if required. A certificate may be requested from the client if needs to be by the certificate request. So, upon this, this is the third phase upon receiving the; so, let us go to the picture.

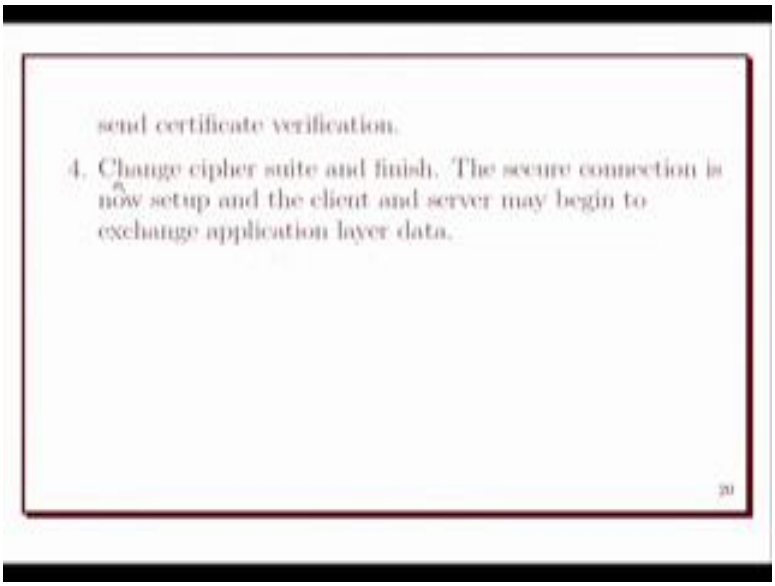
(Refer Slide Time: 24:55)



So, this is the phase one. So, client is send with the hello and server is responding to hello this is the phase one. And this is the phase two; so this is the exchange of the certificate. So, certificate is sending server key exchange, certificate request, server hello request. And this is the third phase; so client is sending the certificate this is the client side, and this is the server side phase two is the server side and phase three is the client side. So, client is sending the certificate in the phase three; client key exchange clients client verify; client is verify the certificate of the server.

Then in the final phase, the change cipher specification and the finish signal is there. So, this is basically the action of the handshaking protocol. So, detailed descriptions are here in the third phase, this is doing by the client.

(Refer Slide Time: 25:52)



And finally, in the final phase - Change cipher suite is finish. The secure connection is now setup and the client and server may begin the exchange application layer data, because there this is the handshaking protocol is basically used to authenticate each other. So, this basically using this four phase, they convince that they are authenticated. So, they are communicating with the correct person. So, server is convinced, client is convinced, now they will start sending the data.

So, thank you.