SSL architecture

We have as you see two SSL states one is session and connection in session we are going to change the all the network connections and lets see what kind of stuffs we are changing first of session identifier ,we are going identity to the session per identity this says ssl may use x509 certification ,but we will see later on that certification comes as an option to SSL if you don't have certificate then don't worry you ca still use SSL and if you do then you surely use ,in encryption SSL use compression and what type of compression its going to use, cipher spec ,im using this kind of stuff, it is such a thing, master secret is secret for us we are going se how are doing those stuff, everything will be calculated based on master secret and other will be premaster secret and we will have premaster and master and by using theism sate secret we are going to use al the master secret key and we are going to do all the master secret things in the session and we are going to ask si it resume able or not, if its then we are going to use it, and when we come to the connection look at the parameter we are exchanging these are the parameter that we are using in the master secret , that is the major thing {question from student}, before encryption we are going to use compression like in pgp, why we are doing that?..to reduce the processing time for encryption so that encryption becomes faster and as you are providing more security and because compressed data is scrambled, using this im going to generate server client random number while key generation this random number will be very important , when two parties establish a session they are using the session by using the handshake protocol and then by using this random number and master secret two are going to calculate serve write mac key, server side mac key and client side mac key in other words when you transfer something form client server you are going to calculate some from client ,the difference will be the message that is coming from you from server to client to yourself and you are going to both direction two different hamc key and that also it may give a clue about the encryption key if you are encryption any data client its going to give any other key and clients from both the directions for encrypting is going to be different im going to call it server and client write key , depending on the algorithm that you are using you may need IV, IV are going to be generated, and also sequence number, we were controlling the sequence number in ipsec, we are going to control again in SSL protocol once you establish a session we are going to calculate its going to be based on master secret and we are going to see and these are the parameters that's going to be exchanged and calculated in session and in connecting so once we did that confidentiality will be provided using symmetric encryption algorithm and this algorithm is going to be defined in handshake protocol and al the algorithm will support ,may be you guys may not be aware of fortezza ,it is hardware based encrypting algorithm it is used in military but in commercial world you cant , just know that this hardware based symmetric encryption message is compressed before encryption and message integrity using a MAC with shared key its going to be HMAC with different padding, application data comes from application layer and we should fragment the data and these are the fragments and each fragments will g and will compress , imagine I did everything hashed ti and cipher spec word there is no are for anything and im going to record protocol im going to fragment it and this fragmentation is based on tcp fragmentation and then compress each compression technique and add mac using mac and compressed data and then encrypt the whole data, so the mac data is also encrypted and then append SSL in the record header ,so this protocol has a header , and send it to tcp and we are going to ad d the tcp header and this going to do to ip and add the ip header and its going to go something like this so we

are on top of tcp and when we finish appending the header and the whole thing will be tcp payload I got the payload right now and this is the function ality of the record and major thing will be the handshake ,we are going to come to that.

SSL mac computation

Lets have a look at the ma computation, SSL compressed fragment, appended to compressed length, whatever the length you have and appended to compressed type what type of compression zip rar, kind of things, appended to sequence number , appended to pad1 , its is byte 36 repeated 48 times ,it is hamc padding and then appended to the mac write secret and that's going to be the input to the hash algorithm and output of hash is appended to path 2 and again hamc and this si appended to mac write secret and sent to hash, but we are not calling it HMAC I said it is different than HMAC ,what is the difference between this and HMAC ,remember in hamc the key xored with the padding that pad 1 and pad 2 to crate the one block of data ,but here we are not xoring ,we aer just padding the data to the secre that way im creating the block ,this is the pad 1 that is the first block and if this block size is not required block sizr you are goingto repeat 36 for 48 times for md5 and 40 times if you are using sha1 , and pad2 5c 48 times for md5 and 40 for sha1 this way you calculate the the HMAC of the compressed fragment and fragmentation and mac of that data is calculated and slightly different data and in TLS what we are going to do is we are make it xor and that's the tls hmac calculation , we are going to use standard tls and in version 1 and 2 Mac calculating is slightly different hmac ,so this is the mac calculation and w are going to append and encrypt along with the mac calculation and that is the idea of the record protocol but before going into record protocol we have to check the details of handshake protocol as that is the fundamental part of the SSL protocol.