SSL MAC computation

Last time we were talking about the SSL protocols and with the SSL protocols we discussed how we are going to make MAC calculation ,if you remember how we are going to make Mac calculation its not HMac ,but this look like HMac only difference is the padding ,instead of putting the HMac in to Mac secret ,we are appending it , we are going to use this Mac calculation and we need two keys one called Mac write secret for right and Mac write secret for left ,s o for different directions we are going t o use different key or calculation of Mac ,this padding is going to be done 48 times if you are using md5 and 40 times if you are using sha1, and this padding 48 for md5 and s40 for sha1, if you look at the SSL record format you can say this protocol is the protocol responsible for encrypting there message and its going to be done by record protocol and format of record protocol is this content type ,major version, minor version and compressed length and the max length will be 2 to power14 +2048 bits , the plaintext is going to be right here and it is going to be compressed and Mac will be 0 or 16 or 20 bytes , and the all of them are going to be encrypted based on the agreement on the cipher suite so inside the suite they are going to agree which algorithm they are going to use, so we will discuss it,

SSL content type

Its going to go one byte and this going to be and the responsibility of this is making the current cipher suite 1 byte so its going to make current suite , so when I got this one if the record protocol is carrying this the pending will be current state and it could be like this one byte will show this and the other is going to alert or this SSL protocol may take handshake protocol that going to be major difference and handshake protocol we are going to make types and we are going to define each one of them and this contents will be greater than 0, there is no anything that will be greater than 0 that you can use this and there is something all the time and if you are not using SSL a tall then it will be at the top of protocol. and if if you don't want to use the SSL then it will go to opaque protocol and doing nothing ,its going to just transfer the upper layer protocol like http lets have a look at the alert and record protocol

Cipher change spec protocol

One of 3 specific protocol which use SSL record protocol and it is SSL specific protocol causes pending status to current and hence updating the cipher suite and its going to be the strongest form and the strongest and less strongest one that you don't want to use will be at the bottom, first of all key exchange protocol and encryption algorithm and the Mac, and also the gives if it is necessary of the protocol and it could be something like for the key exchange im going to use rsa, for encryption we are going to use cbc. Aes this is just an example for Mac im going to yusesha1 and Iv for aes algorithm these are going to iv and do I need iv for aes if I have these are going to be the iv , doing need for sha1 I f indeed then these are going to be stuff, generally these combination corresponds to some kind of codes 001, 00111 and lets put its like this ACFBE00D this is going to be the code that im going to transfer when I transfer that code the other side will understand that I m going to use rsa for this and aes for this and sha1 for Macv calculation these quotes are standard and everybody is going to use the same code that means in the implementation of sSSL cipher quire there quotes are written , personally I used to work in a company in oregan and we were working on elliptical curve cryptography and my project was to piut

the ECC for key exchange and for this cipher suite and right now SSL protocols support ECC, the only thing is you have to get and this kind of standard and the coding for each combination and this your stronger and the other are going too be ls stronger an d the second strong is this and the third strongest is this and in the sending the order basically when you transmit same to the server or server it may be transmitting it to you you are going to select which ever is strongest for you and you are going to select it and if the other guy s supporting the same rsa and sha the its going to go with the first choice, they agreed now on strongest choice on both parties and when we make this choice then this is on accuracies then for that we are going o use this combination and exchange this cipher spec are going to make this changes and we are going to make that kind of info in that information so this guy is making the occurrence state and nothing else , as hence updating in the cipher suite in use for this related alerts to peer entity ,severity may be warning or fatal ,so two level of severity unexpected message ,bad record Mac ,decompression failure ,handshake failure ,illegal parameter ,close notify ,no certificate, bad certificate unsupported certificate ,certificate revoked ,certificate expired and certificate unknown , you name it so when you have problem you will get the warning messages and these warning messages will be something like this when you get some problems and the problems will be of two level it could be warning it will not effect the working of SSL protocol and it will ask you that you will accept this certificate and if you say you can run it is going to run , and what else compress and encrypt like all SSL data and the record protocol is also going to encrypt lets move on

SSL handshake protocol

Everything is going to be over her allow server and client to authenticate each other, negotiate encryption and Mac algorithms to negotiate cryptographic keys to be used , so in other words everything is going to be negotiated and everything is going to be decide right here and those decisions are Mac algorithms ,hash algorithm key exchange along and necessary keys for those algorithms, and compromise the series of message ,we are going to have 10 message in each phase and the first phase is called security capability second phase is server authentication and key exchange and the rid phase is client authentication and key exchange and the finish this is the last stage and these phases are stages and we are going to have 4 stages and for those stages we are going put null one is hello request is null and client hello im sending which version im using for SSL protocol , im sending a random number and it will be 32 bit time stamp and I believe it 20 bytes and 28 bytes length of random number along with the 32 bit tie stamp and we are going to use it for , we are going avoid replay attacks and into words these random numbers are used as nonce we saw that in every protocol there was random number for exchanging protocol and that random number over here is client or server random number and im sending session id, cipher suite , compression method , in other words client hello transmit this mesa to server and server will send seem to client and there is certificate and you can send those certificate to client and certificate requires it will say if you have a certificate then send that certificate and certificate verified client key exchange and these are all things that we are going to exchange so lets have a look the details the first phase is security exchanging phase the client is sending client hello message with the protocol version and session id is 0 or 1 if it si 0 that means client is requesting a new session and if it is 1then client is requesting the existing session so 0 or1 is going to be important for us, cipher suite ,now clients transferring the list of it to server so server will look at the list and compression, which

compression method they are going to use, initial random numbers so which random number they are going to use and im going to cal client and im going make 32 bit timestamp, and server will reply client with the server hello message and its say I selected this version for the encryption for the Mac, so over here you see there exchanging the necessary info and version random ,random ,session ID ,cipher suite and compression method and when we look at the risk able capability for key exchange ,SSL is going to support this key exchange and its going to be RSA and over session key will be encrypted with rsa and fixed diffehellman , you haven't heard of it I guess, this fixed differ hellman is going to use X509 algorithm but the public key and the parameters that I m going to carry will be differ helm and keys and im going to call it diffehleman certificate and why im exchanging difeh elman with certificate ?.right now im exchanging the parameter with DS ,because certificate is ds stuff so the certificate in the diffehelmna is going to exchange by using rh certificate there will be no man the middle attack, ephemeral differ helman I don't need the certificate but in need authentication still to exchange the diffehekman signature im going to use rsa or dss signature to exchange the dfeeheman parameter ,anonymous difehelma nis just a difehelman with man in the middle attack and foretzz hardware stuff s I said 1 more thing I have to say is ECC difehelman which is called elliptic curve cryptography differ hellman so that is another key exchange method that right now SSL support and these are key exchange algorithm we are residing in the first phase and the algorithm that these guy support is rc4,des,idea, aes, fortezza ,md5 and sha1, stream types block and cipher types, stream, is exportable, if was very big in there past it s not true, most probably you are not going to triple des, but you can use so that came with the os , outside of US so if you are using net scape with SSL implement you will not be using full capability of SSL you are going to use DES with only 40 bits, you will not be able to sue triple des, what I did was, before changing the SSL we were encrypting the data with 128 bit key it was double encryption I wasn't using plaintext for triple des protocol and I was actually using encrypted text as input to the SSL just to avoid this limitation it was a kind of way to solve problem right now there is no problem as there is all round the world and sa it is open source at that time its not open source and you can use any protocol for any number of time basically and whatever algorithm they want hash size is 16 byte and20 byte for 20 bytes , key material is a sequence of bytes that contain data used in generating the write keys material generate another thing im going to explain after this handshake protocol it doesn't require cryptographic, instead it requires another computation to generate the keys and iv size so these parameters are going to exchange in the first phase and with the client hello and sever hello .,lets go back ,lets go to second phase if you are supporting certificate you have certificate in both entities server will say lets decide about the key exchange based on the request and its going to request certificate from client if client support hat kind of stuff its going to send those kind of certificate and server will send im done message but if you are not supporting certificates this part is going to be secured and for the server key exchange if you are using rsa key exchange or differ ehllman key exchange you don't have to send server key exchanging what im saying is this ,when they send serve done message this server done message is calculated something like this all the server calculation done so far upto this message appended to server hello random and appended to client hello random number and all of them are going to be based this is kind of verification other side is going to same calculation if the result of the calculations are same so so far they exchanged the parameters correctly so they is no misunderstanding and on the server and client side, so we a redone with this one, so we are moving into phase 3 client authentication and key exchange it will say im going to use this key and im having this certificate and Im

going to send the certificate to server if there is not server its going to send the client key exchange and at the end if you look at the table, right here its going to send certificate verified message ,so it is going to verify al the messages sent to this message until this point , with sever hello we did this verification upto this point , over here we are including this phase as well and that verification is calculated like this certificate verify signature md5 hash they are going to calculate md5 and sha1 version of same thing pad1 appended to master secret, another concept ,SSL will use a secret called master secret and that secret will use another called pre master secret premaster is going to be decided by the client and its going to be distributed by the client and both partied will together calculate the master secret I will explain how, so everything every cryptographic computation is based on master secret.