Kerberos version 5

Kerberos version 5 , will address all these issues, in other words technical deficiencies and environmental shortcomings, is going to be solved and version 5 is a standard and standard documentation can be found in RFC1510, you can see all the details, lets have a look at version 5 that way we will clearly understand the things.

Kerberos version 5 message exchange

In the first packet, as in version 4 I m going to have options and those options are forward able removable options , like in here , initial, software authentication, hardware authentication, renewable, may postdated ,invalid, proxy able, non proxy able, forward able, forwarded these are options ,if you use this you will this is issued by protocol and not by TGT, and this one initialization and hardware authentication may require smart card authentication , may be finger printing may be voice recognition, or may be retina or something like that, we have to pass those tests or else it could be hardware decision also , in the authentication we will say hardware authentication then the AS will ask if your implementation is different thing, it will ask let me look at your finger print ,that kind of stuff or put the smart card on you card reader or something like that ,renewable tells TGS that this ticket can be used to obtain a replacement ticket that expires at a later date. That means your credit cards are renewable ,may post date ,it is not that famous in US it is very famous in Europe ,you can write post dated checks , if you want to pay some person some money, if you want to pay it by june you can write check with the date june and that guy will cash that check in june, but here its not like that if you write the check then they can go and cash it immediately and if you write the post date and they are going to cash it later basically , so it's the same thing here so you got the ticket post dated and you have to be ready for application in april for example and you are going to store those ticket and you solved those problems basically , invalid this is invalid and must be validated if it is not renewable ticket if it is expired that is going to be invalid ticket tby checking that one the system will ask the user whether the ticket is valid or invalid , if the ticket is renewable form the system but if something happens and some kind of error then check the system, what they are doing is they putting in some kind of list ,like in certificates they have some certificate allocation lists expired or certificates are invalid we are keeping everytime we get the certificate then I am going to check the lists so we are going to have those kind of lists ,proxy able they may carry your credentials to someone else one behalf of you ,proxy is the same thing and forward able is almost the same thing, but this time for different network address so proxy able addressable are almost the same thing so lets go back so what im saying is in this option field you will tell the server that I want this type of tickets, lets say you are forwarding the ticket and there will ne forward able flag is up there ,so this ticket will be for renewable ticket so options and flags are pairs and flags comes with options, then ID_c will come they include $realm_c$, in version 4 we assume that we are working on only one realm, now they realize that they work on multiple Kerberos basically, so that's why they are putting it ,time is time stamp and nonce is a random number ,we will use timestamp along with random number , because to allow this to get the freshness of the incoming message actually, when you go and login , k we got this so what will happen AS is going to go and check the id of c and check the related password and from that password it will check the access and so on and so forth and in other words it will check your account and help you check your key K_c and then its going to generate ticket TGS and this

times is lifetime of the ticket but it is not 8bits it will be in the form of TGS, ID_c is there AD_c is there, $realm_c$ is there session key generated by AS is there and flags are based on these options everything is encrypted with the key shared by AS and TGS, so looking at the rest of the message $ID_{tgs,}$ $realm_{tgs}$ included in this ticket , ticket will be valid between this time and you may have multiple request and session key inside this ticket an everything is encrypted with the key derived from the password so compare this message with message two in version 4, remember technical deficiencies we said double encryption ,who has double encryption , you can see ticket is outside the encryption ,and we have the times and nonce now we are solving the problems and the options flag and we are solving the short comings and technical deficiencies of version 4 and when we messages ,client will tell us to enter the password and when we enter the password , you can decipher this and you know that this coming for this nonce and you got the session key that you are going to use for authentication and its going to be valid for this time and you are going to put the ticket there and you are going to put the ID_c and $realm_c$ and write the session key that I recovered from message 2, so while im forwarding the ticket to the authenticator ,im also defining the options these options are for , sgt, these options for tgt, and now im going to say this going to tbe ,1st one I have hardware authentication as my option and I want renewable ticket for this one, sgt, so you are going to send to tgs by checking the authenticator how it will check ?..how tgs will validate the ticket ?..decrypt the authenticator I have to know k_{ctgs} but it doesn't know still, first of all it has to decrypt the ticket and will be available for k_{ctgs} ,ID_c and address so it is going to check the incoming address , and it is going to check the ID of the address ,ID_c by doing that its going to validate the ticket, so get the ticket decipher it, get the authenticator, decipher it check id of c , and check incoming address AD_c and check incoming timestamp ,authenticator right here timestamp and the timing whether the authenticator is valid or not as authenticator life time is short. {question from student}. This time it didn't even in version I said the same thing ,the thing is address of the client is incoming message address, so they are going to check it form there ,still in version 5 your are struck with workstation, when you send the request ticket AD_c will be right there and this message should come from this ticket , and If it comes from different ticket its not valid, with this authenticator I can validate the ticket ,so the important thing is authenticator and timestamp this is short and if the authenticator timestamp is not in the valid time period then I m not going to accept this ticket im going to reject this, lets say I validate and everything is successful, now what, tgs is going to prepare what?..SGT that is ticket $_v$,which server, it is defined in packet ID_v so its going to have flags and kc_v will be shared between client and the server realm ,id $_c$,ad_c life time all of them encrypted with K_v what is K_v secret key that is shared between tgs and destination server and then when tgs prepare this ticket it s going to send this packet , ticket, id ,nonce, times encrypted with the k_{ctgs} that is session key distributed by AS, when destination know k_{ctgs} its going to get k_{cv} ,if it knows that then it can prepare the authenticator right now and can forward the packet with the authenticator ,so when the server receives this in the authenticator along with the options it is going to check the flags whether it is matching or not and its going to authenticate the ticket with authenticator ,look at the last part it comes from the v, it says that subkey and sequence number, but at the same time if you look at message 5 you can look at the authenticator and it Is little bit different that these guys it includes a subkey and sequence number client says remember after I got the key and encrypt the number I want to use this key and I want to use this sequence number and remember sequence number in tcp ip protocol and this number is unique and how many bits we are using for sequence numbers??..64 for ip version 4 , check it out, the thing is that sequence number is

decided by the client ,and it will say I want to start my first packet with this client and I m going to use this key and that key is subkey and if the server replied with the sequence number, server information is going to be filled with the client information in other words server has more power and if server send the sequence number then it will use client and this sequence number and if the server doesn't reply then the client selection going to be used so this Is going to be the version 5 ,look at all the technical deficiencies and environmental short comings we saw all of them in version 5.