Kerberos 4 message exchange

When the destination server receives ticket from the destination server it will do this it can recover the ticket and it can decipher $ticket_v$ and it can get the k_{cv} ,when it gets the k_{cv} it will get the authenticator and when it goes to authenticator it can decipher the authenticator and its going to recover ID_c, AD_c and timestamp and its going to go to Id_c , ad_c and if there is a match then this ticket is coming from this client. And this authenticator is going to be valid only for a short time. This way it does know that the know the ticket life is like this and this ticket is a new ticket and server is going to give the service to the client now you can see your inbox that is the idea, may be in some cases it may require mutual authentication ,we know that client sent this ticket and how the fclient knows that it is talking to real server and how do I know that the message that i m exchanging is also from destination v and not from some to there server and if this is the case this v is going to send this packet timestamp 5 which was in the authentication server will do something for this timestamp for the y1 and encrypted with the session key recovered form the ticket and only v who has kc_v information and k_v information can recover kc_v and no one else ,when I receive this one I m going to decipher this and im going to look at this time stamp 5 ,I understand that someone recovered the message that means someone received my information ,of session and who can do that whoever has the K_v information , so whoever gets the timestamp 5 must be the server that I want to access I want to get serviced for ,so this way the client knows it is talking to server , are we clear?..so this is Kerberos 4 , and we have one more version of this and that is Kerberos 5 but if you did understand then you can see the problems with this one ,do you see any problems??. What were the problems that we came up with this one?..first of all we were transferring this password in clear form and we will remove that one there is no password transmission second is the ticket could be replayed now the ticket cant be replayed as we are sending ticket with Authenticator and also ticket has a life time and no one can resend this request as ticket has timestamp for this one and also this ticket , we are not sending it alone we are sending with authenticator ,so we are avoiding the problems that we saw so far with the simple protocol , and simple solution and also it comes with key distribution solution what was key distribution over here, k_{ctgs} is sent the AS to the client in this ticket and to the client in this ticket , and TGSis distributed another ticket K_{cv} to another client and within this ticket to the server ,so im having solution to the session key distribution ,I didn't say which algorithm we are going to use and all I said we are going to use this key for encryption and this key for encryption what is this encryption ?..is it public key encryption it is obvious that it is symmetric key encryption as there is only one key distributed in other words ,Kerberos is based on symmetric encryption algorithm we are only using symmetric algorithms and we are using only symmetric encryption algorithms and I didn't give the name still ,whatis the encryption in version 4?.. it is des encryption it does use only DES algorithm and that is one of the drawback of the version 4, it is implemented to use , des algorithm, when I say des algorithm you guys should ask me question because des algorithm doesn't make sense which mode that they are using that is the important thing ,for des algorithm, unfortunately they are not using any mode ,they are using a totally different mode called PCBC mode and it is pretty simple mode what we are doing is this is the key and this is the plaintext and im getting the cipher text , lets put it right here and this is the cipher text and im going to xor cipehrtext and plaintext and that is going to be my cipher text, if you look at ,let show you guys the actual thing, and this is going to go this is xored and xored with the second plaintext block and this key is xored with the second cipher text block and

this cipher text block 1 ,block 2 and this is going to go like this until and p1 and this is goingto be encrypted like this and we are going to get c1 and that is called PCDC mode ,what they are doing is they are xoring the result with the cipher text and they are xoring that with the next block of plaintext and that is the result of the des algorithm to get the next block and this is PCDC kind of standard basically. Which is not good for he version 4 if you are user and if you are user of Kerberos then you will tell , I want to use AES algorithm or you will say why I cant use AES ,as it is a standard algorithm right now , in other words we have to change these things for those new algorithms , and may be we should put a option saying that I want to use AES or blowfish or IDEA for this , so in other words its going to come with some drawback and there is some major drawback in this one and what is that one can you see that ??..we solved our authentication problem and we are not going to go back and lifetime and if this lifetime is large then you don't have to come there and if it is short you have to go and visit AS and lifetime 2 will decide how many times you need to go to AS, {question from a student } .. 4th message right here and you can see the same thing in message 2 and ticket is already encrypted and hwy do you do it again , the time is pretty big and the time it is going to be encrypted will be bigger. In other words encryption is costly stuff and if you keep this ticket outside of the parenthesis It is going to be still same ticket is already encrypted , if you are doing it again you are increasing the message size first of all and you are doing another encryption on the bigger message size, so that means it is going to be unnecessary we don't have to encrypt the encrypted ticket , this is from the encrypted point of view ,if the client Is complaining that this system is slow and all, you can improve it, you can place this ticket outside the parenthesis and nothing is going to change, so that's the one drawback ,so far we found two drawbacks, one is double encryption and the other is PCDC mode which is not a standard mode. But the other things are also important ,what is the maximum life time defined by the Kerberos version 4, well they are using 8 bits tot define the life time and each bit defines 5 mins if 00001 is the defined in 5 mins then the lifetime is 5 minutes, if it is 10, then the lifetime is 10 mins if it is 11 then it is 15 minutes and like this we can have 2^8 times 5 minutes. Which is roughly around 21 hours in other words you cannot assign ticket life time in one day, it is going to be less than one day ,that is maximum, {question from a student }..it is the setting of the Kerberos to define the lifetime of the authenticator , it is going to be very short ,so as you see, ticket lifetimes are going to be limited ,maximum is 21 hours, is this a bad thing or good thing it depends on application ,give me example of some type of application ?...i would say this, if you are running simulation and simulation running time is more than 48 hours ,what are you going to do??.. and Kerberos is controlling access?..you are going to go there and you can run it only for 21 hours ,and after that server will say Im not running it again as your ticket is expired and you have to go and get another 21 hours and you have to start again and it is like never ending story, cant we have some more flexibility ??.instead of using 2^8 5 minutes kind of stuff, why don't we use the credit card approach to the lifetime of the tickets saying that this is valid form or valid to, this way you can be more flexible right?..so that is very good thing in terms of lifetime, otherwise we are going to have problems, in other words Kerberos will not be able to accommodate all the possible things and in Kerberos 5 we are going to change this thing and we are going to put valid from valid to expiration date without using 8 bit numbers and that's another drawback in version 4 so this life time is limited , so anything you can see?..any other short coming so f this version 4?...again think about the old network environment what you are doing in old environment, for example when you are having access to the print server and print server will go to file server to access the file right but these are two different servers , you have the

ticket for the print server and you don't want to deal with the file server you just want to say this is my file and I want to print this thing and with this I cannot so although you have access to the print server , file server does not have the access on behalf of you .so in other words in some cases we must give our rights to someone so that that someone can use the rights on behalf of me for example , to print the file ,so in other words I must produce a key and that key should be forward able or reusable, for the file servers, if it is forward able then print server is going to forward that file to the file server and its going to get that file and it will be able to print your file, in version 5 we are going to accommodate those kind of needs, we are going to speculate that stuffs and we are going to see what they have ,and the other thing is what happens when your credit card expires ,you will get your card before the expiration date ,they don't wait till the last minute , you may not get your credit card after expiration date and you are getting your new credit card while your old credit card is valid ,although I said lifetime are flexible ,sometimes you may not be able to guess the time required for your application and you said that ,48 hours , but it took 72 hours what you will say is , I want the ticket that is renewable so that before expiration date system can renew the ticket before the time is over , you can see the cases and this cases cannot address , with version 4 , my question is version 4 although it is pretty secure we saw all the security problem that we discussed and it has definite short comings and it definitely needs some extra thing to address the issue of modern world and we are going to change this guy and in version 5 while we are asking the ticket in packet 1 you will give your option 1 and you will say I want this ticket forward able, I want this ticket renewable and then when you get the ticket it will have flags and it will say this ticket it renewable and this is forward able kind of things so there will be options and flags will be responsible for options and now we are going change the life and we are going to use the symmetric encryption algorithms and we are going to just des anywhere and that's going to be version 5 and lets take a look at all the parameters once again as I may asks these things in exams ,you don't have to memorize the things you have to know the logic ,if you know it is easy to come up with this , you have a packet and what packet you will use for the request , you will say this is me I want to access this and I want to know the timestamp and in the second packet I know that I will receive the ticket ,I m receiving the lifetime of the ticket and the session key timestamp is going to be by default send this to destination and you see you don't have to memorize those things you just have to know it , memorizing and knowing are two different things ,they say memorizing is not a shame but it is a shame not to learn, we have to learn this stuff , but if you don't know it is not a shame but we are going to learn it , because lot of guys will ask questions as midterms are coming , so they will ask ,do I have to learn this packet, do I have to do this and that , you don't have to remember anything you have to know it ,because definitely questions will be coming from this one.