Elements of Kerberos 4 protocol

Lets have a look the meaning of all the parameters we saw in version 4 , in the first we saw ID_c, ID_{tgs} and we know that ID_c tell the AS identity of the user and ID_{tgs} tells the AS the user request access to TGS, verification of time and we use timestamp for that and in message 2 we have Ek_c remember ticket is encrypted with this one and it is the key that is derived from the user password K_{ctgs} was session key shared between client and TGS and lifetime 2 inform client the life of the ticket to TGS and TS2 , informs the client that time this ticket was issued ,and $Ticket_{tgs}$ and what is the name for this , in network security everything will be easy while im saying it , but when I stop then you go somewhere else, ,so what was the name for $ticket_{tgs}$ its is the TGT and im going to use this for long time for SGT, in message 3 there is authenticator generated by client to validate the ticket and this is encrypted with what?.. and It is distributed by AS and it is K_{ctgs} and that's the key shared between client and TGS and the session key K_{cv}this time this session key is going to be share between client and server and the rest one more thing actually AD_c prevents use of workstation one that initially request and this is going to give the limitation and there are two parameters ID_c and AD_c ,AD is network address, when your request the ticket from this workstation you can use only this workstation and whatever the work station is in the ticket and everytime you use AS , it is going to check the incoming packet or this ticket is going to check Address of c if the address is different address then you are struck to your workstation ,clear?...so that is the meaning of AD_c ,is there anyother new one, you guys know almost all now , so if you have multiple domains of Kerberos what will happen , lets have a look and we call it Kerberos realms

Kerberos Realms

When we look at the Kerberos environment consists of Kerberos server that is AS and TGS ,number of client all registered with server ,that means you have to open an account and application server sharing the key with the server, Kerberos server ,client and the application server, ,but this is going to be one Kerberos domain it is a single administrative domain and if you have multiple realms this time the Kerberos server must share keys and trust and In other words if I have one Kerberos key in one domain and the other in the other domain and my client wants to excess server on the other domain ,it means Kerberos servers must share some secret. And must share trust and if someone gets tgt from this domain and it will use the SGT to use this TGT and so that kind of stuff, so what im doing is this, this is one real ,real will be server and client , and there is one more realm this is the Kerberos server and my client wants to access my server in this domain controlled by both domains in Kerberos , im going to go to local AS which is message AS and im going to get the TGT and im going to use it to local TGS ,to issue me a ticket to go to remote domain tgs and I m going to get the sgt form tgs and 2 steps and I will be getting a TGT from a local TGS {question from a student}. As I said we don't have name for that one jus t it is ticket for tgs and it will be always local tgs and tgt will be always giving ticket by the AS {question from a student}. It doesn't work that way , we have to these three levels , and everytime I do this I have to go to AS to authenticate myself and what we are doing is we are not changing anything in original mode and logon session here and get there tgt and one more step here instead of tgt I m going to get the ticket that is for this digits. {question from a student}. Network transparency , you know that one most probably when you do this Kerberos reams we are going to see some table and then you can know I guess and this table is address ,this domain and this addresses ,system may had that network and all

you have to do is that you have to name the domain and if it is on the other domain and then you are going to get his step these are user transparency and Kerberos client require only you password and nothing else, {question from a student} if you are adding any other Kerberos right there you have to include all these guys and these guys should share and upgrade everything as client is quite dumb , as I said you are going to ,client doesn't know it is just doing its work and during the transmission may be client knows that this message is coming from TGS, it is random it is remote service, {question from a student}, when you go to TGS, we went AS, what we did is same ,we got authenticated and we got the ticket and client says that I want tot access to these id's is this id in my local domain? That is Kerberos implementation part ,all you have to do is name the system and it will check the necessary routes or whatever so its going to go and say this is this ,may be you know the sever , you say that its in santa Barbara then you can say this is a id of this server may be you will address something like server.address.santabarbara.edu kind of link and it is remote tgs ok. Alright we got the ticket lifetime of the ticket is ts 2 and ID_c and key is K_{ctgs} ,so nothing is different because I was in my domain and this ticket is tgt and I got this one encrypted with k_C and client will ask the password , recovered the session key and I recovered the ticket what im doing now is im using the ticket along with authenticator, as I have K_{ctgs} and I cant prepare authenticator now and im going to my local tgs and my local tgs is generating a ticket and ti is for local tgs and it is the session key that is going to be used between client and remote tgs, lets have look at structure this ticket is going to be encrypted with what?, this is going to be secret key that is shared between local tgs and remote tgs ,remember I said, when you re using a domain then you are going to share the tgs ,so you must install that kind of secret jus tbe case of these two so that the other guys will not understand the ticket and that is the reason why ticket should be shared between domains , and when we go to the remote tgs im presenting y ticket along with the k_{ctgs} and the remote tgs and then tgs is going to get this one and authenticator will recover the ticket and form the ticket its going to authenticate the use and over here its going to prepare sgt, timestamp of the ticket and a session key and it is shared between client and remote domain server and then when I receive the ticket I will recover the session key and I will get the ticket and go to the actual destination sever and this way actual sever will recover the ticket and it is called kerberi , multiple realms Kerberos and it si called kereberi , this is distributed environment and you will use Kerberos as multiple domain and most of the time the Kerberos that you are going to use will be kerberi so I want you guys to focus on packets and we know only version 4 packet standard and we talked only about the access upto that point and last message was mutual authentication TS5 , encrypted and that is shared between server and the client , but we didnt talk after that, they know that we have authenticated and we are talking to real guy ,you got the ticket and you have the excess what will happen ,you have to download that file and you got the file server and lets say that those files require security what you will do ?. in Kerberos it doesn't say you are going to use encrypted AS after you exit, we didn't say that ,that's another drawback ,as an engineer you can think that I can use a key that is between client and server so I can use that key to encrypt all the incoming and outgoing message between source and destination, but we are either establish the key for encryption in version 4 or its going to be like, client may I m going to use this key or sever might say that im going to use that key , but we may think as an engineer as an option ,but we may use that key used between server and client and that will be remote destination ,so you may use this key and you may have the better structure for the version 5. Now we can go and look at version 4, any questions?..{question from a student}. All you have to do is install Kerberos software , client version

of the Kerberos and on the server side ,server version of the Kerberos but this is a good point Kerberos is in client server mode and the architecture is like that

Kerberos version 5

It was established in 1990's and it provides improvements than version 4, and addresses environmental shortcomings , and technical deficiencies so two things environmental shortcomings what is that , encryption algorithm , what was the encryption algorithm in version 4, only DES , we need AES or someother protocol, it was ip based and nothing else byte order is implementation thing and I think it was most significant byte like little india n and big Indian , lets say that this is message and one is most significant bit and the other is least significant bit you can write in two ways , most in one side and least is other side, first most significant bit and it comes from microprocessor architecture Pentium is this one and this one is spark ,some station use that ones too, so when you encrypt this one using implementation algorithm or something it is very important to know how you are using the data you are doing how you are solving this problem in CN , if there are different data types in the computers how the hell you are going to communicate ,I don't care if you are using Pentium or motorola or spark ,do I care , something handling that problem and what is that ?..which network and layer is responsible for that ?. the presentation layer, is responsible for the common data representation and the protocol and it is very important is ASN 1 and when you receive the message from the presentation layer you are going to use this ASN1and this is common for everybody whether it is little or big Indian ,because everytime the format that you are going to use is ASN1 it is very important and the one other is BER , but the major protocol is ASN and this way we are solving this incompatibility problem ,so this way they are using little Indian thing and they are also for big Indian problem , and ticket life times are limited what was the maximum life time is and the actual is 2^8 5 minutes , and we don't have any authentication forwarding and for example if the print server wants to access to print my file I must present ticket to print server so that it can use that ticket to go to client server on behalf of me to go to the fileserver to get the file and that kind of forwarding is not available in version 5 and what about technical deficiencies ,double encryption packet 2 and the other is packet 4 and in other words we are encrypting the tickets twice that is not good and they are using non standard mode of use TBC and the session key is distributed both AS and the tgs and password attacks are possible and Kerberos is based on passwords, imagine passwords are discovered what will happen?.. in packet 2 I receive the ticket encrypted with , sent by the AS that tgt and is encrypted with the key shared between AS and tgs and im going to get the ticket and im not able to see the content of the ticket , ticket is safe ,but remember in packet comes session key also , if I know the password I know K_{ctgs} and the ticket is sent through authenticator and I know the session key right so that I can fool the other person In other words I can impersonate the other person if the password is known clear?. What about on the tgs side and im receiving the tgt , along wit the session key ,encrypted with what the session key distributed by AS, that is called K_{ctgs} I already recovered it as I know the password that means that I can recover the other key I can recover the server and if I can recover the other keys then I recover the server and did you follow ?..so I can impersonate any one in the client level and in the server level , that Is 2nd and 3rd level and that's why the security in Kerberos are based on passwords. Thats the idea.