Simple authentication dialogue

Lets go into details of Kerberos 4 and lets also consider these multiple logon sessions

A more secure authentication dialogue

First of all im going to have multiple steps 1st step is called once per user logon session what I am going to do over here is , im going to get authenticate and im going get the ticket and that ticket will be something that will tell other to authenticate, and the other draw back was transmission off password in clear form and what I am doing is im not transmitting any password now , im only, transmitting ID_c and ID_{tgs} ,TGS is going to be the second server, which is going to give me service granting ticket and I will be telling I want to authenticate someone and im going to collect TGT, and give the ticket, im only sending IV of myself, look at the response of the server, I m not sending any clear password, ,in response it generates TGT of the successful authentication encrypted with K_c, it is something interesting , if you look at the definition of K_c , let me show al the guys here, if you look at the ticket $_{tgs}$,it Is encrypted with ID_c and ID_v and ticket $_{tgs}$ and to avoid the replay and the life time and this is going to be valid for sometime and timestamp will be valid for sometime and this timestamp will be in such a way that it cannot replay it , in other words, this ticket although encrypted like this it ,it is also encrypted like this , K_c is the key which is derived from your password, into there words password related key ,so what we are done is this, this guy sent the ID_c and with this remember when you install the Kerberos we are installing the passwords of the user and ID_c is coming and from this I can see the password and from this password I m going to generate a key, and that key is going to be used to encrypt this ticket and this is encrypted using the key sharing with AS and tgs and this key installed during the Kerberos installation and you are the Kerberos client and you receive this ticket what are you going to do?.. how you are going to decrypt it??...client will ask a password from you when you enter your password they are going to use your and they are going to use your password to create the same key and if you don't enter the password the system will not be able to generate the same key, which is K_c, but if you are entering the right password, K_c will be generated easily and you can get the ticket when you transfer the ticket ,look at the package tree how TGS knows that you have the key that is authenticated , ?..there is only use r who can decrypt the ticket who is client who have the password, and the result of authentication , is right here not here. So lets have look at the figure and know how we are going to generate the password, this is the password, how many bits I don't know its going to be character ,character and I m going to remove the 8 bit of each character and im going to make 7N bits and im going to fault on 56 bits, on top of 56 bits and im going to xor it and if the last block is not 56 bits ,that doesn't matter, and append something and I m going to accept the block and bit wise xor this is the result and im going to put the parity bit away and im going to make it 64 bit input key and im going to make it input bit k_{pw} and this k_{pw} is going to e the key for DEWS algorithm and input tot the des algorithm is going to be your password and first 8 character is going to go to DES and output is going to go to another DES and second block is going to be here and again DES encrypted with K_{pw} and finally you are going to get 64 bit output key , K_c and this is the key that is going to be input to the key and that is the key generated a form your password, when I logon to Kerberos and when I get this ticket and im not going to get the AS just to logon to this session that's why im having type of service and session I need another server called TGS and for every server im going to go there an with this two packets here I eliminated the password problem in the previous

protocol and I also eliminated the replay attack of the ticket ,you cannot replay the ticket anymore, if you don't get the ticket replaying is not going to play an important role, because TGS is going to play a main role not the encrypted role {question from a student}.. in protocol we are going to put some nonce and time stamps so that we can say that this is the protocol, so this is the first part ,I hope this is clear that why these guys are like this and that's why im trying to explain like this and that is why it is an original Kerberos protocol and in original you are going to have more things,