Key Distribution Issues

What are the issues in key distribution? We may have hierarchies of KDC's and with this KDC's we are going to have some public key infrastructure , we are going to have some servers are going to distributed keys and so on, certificate I have to say, one and it is going to be responsible for one domain, and the other is going to be for other domain and other servers responsible for the keys for these servers so and so forth we can have those kind of K ,this is a big deal, when we see the details of public key cryptography we will have better understanding of public key infrastructure ,or the hierarchies of KDC for example ,but this time we are not going to call them KDC we are going to call them CA's , and session key lifetimes should be limited for greater security ,thing is you have to change a session and in every session and you may have, automatic key distribution on behalf of the user , but you must trust the system, or we may have decentralized key distribution in other words, we may remove that KDC, and without the KDCwe can distribute the keys lets have a look how we do that ,1 st key distribution in automatic key distribution we are going to have front end processors ,there functionalities to deal with the key exchange, so what we are going to do is ,host is going to send a message to FEP , requesting a connection ,when these guys got the request it will go directly to the KDC and KDC has the front end process, in other words dedicated process for one function and this guy is going to communicate ,lets say this guy use (……..) algorithm ,so they are going to use that algorithm without telling the user and when they establish the key they are going to encrypt and send it here and this guy will decrypt the message and will send it to host and this way I m providing what?..user transparency, all you have to do is send this message to the encryption and this guy will take care of the rest and you know that they are goingto establish the session key and they are going to encrypt the message and they are going to send the message to destination, even the host doesn't know the session key. And they may hide session key from the host and in some scenarios they may say that the host to recover the messages or something like that you may require those keys, so that if the host knows the keys it will be good, but if the host knows those keys ,there is another security problem, how are you going to protect those keys basically, so we have to be careful for dedicated transparencies for dedicated key distribution and implement that (……) except that KDC is sending the key or the ticket to this guy ,it Is not coming via this guy ,that could be slight variation that you can come up with the same protocol if you want and the other one is the decentralized key distribution ,again imagine this is a software and I am going to install the software using initiator A and B ,that is shared between A and B, I am going to call it MKM. Initiator is going to send a key to nonce B and B is going to select the session key and its going to change the request N_1 and N_1 is changed a little bit and lets say it is N_1+1,and its going to select another noun plus two and its going to encrypt, all of them with the secret shared between A and B installed ,in A and B ,and when A receives this message 2 what are the steps that I have to take ,its going to decrypt the message using the master key and its going to recover the session key and it will understand that this session key is for this request and for this noun, and it is for ID_a and it is going to get N_2 and its going to increment N_2 by one and its going to send, this N_2 encrypting the session key recovered from packet 2 ,why we need this third step, any idea?..whether A is using the correct key to recover session key because B doesn't know whether A recovers session key or not ,B only knows that it is talking to A as it receives the message from A and it is trying to send K_s information to A, only in packet B,it knows that B had successfully A recovered session key, if A hasn't received the session key it cannot send the packets, it cannot, encrypt $F(N_2)$,as you see in this scenario, I don't have, KD_c so I eliminated this third party, and I focused on the elimination of the third party and without any third party I am able to exchange the session key, alright?..another thing sometimes I am going to use some keys for encryption and sometimes I am going to use for MAC calculations or hash calculation , but I want to generate a key in such a way that I am going to put a tach to the key saying that this key is for encryption and this key for MAC or hash if I am going to do those kind of stuffs ,then I am going to use these kind of techniques,

in this control vector it is going to define the functionality of the key, and I am going to have master key and it will be xored with session key and we are going to encrypt session key with key generated with the control vector and result will be encrypted vector and I know the functionality of the session key, this is an encryption key or signature or MAC key I know the functionality, to recover the session key what I am going to do is ,I ma going to use the same control vector which goes to the hash function and it produce the same result in the output of the hash function in the this side and in this side and I am using the master key again and using the same key for xoring again then I have the session key back with the knowledge of usage of the key {question from a student}. It is not it may be something like this if my control vector is 0001 and this key is for encryption and if it is 0001 this key is for signature generation and if it is 0010 ,it is for MAC calculation if you need a common function ,its not a big deal its just defines which type of key you are generating and where you are using and you will see those keys are used for MAC calculations. And also for the symmetric signature generation, although we don't use too much as actual signatures are public key based, but in case any application that you have to use in signatures we can define that key for that generation and that is one way to define the functionality of the key.