Substitution S boxes

You are going to have 8 S boxes which is going to map 6 bits into 4 bits. Each S box is actually 4 little 4 bit boxes. So the first and the last bit is going to show us the row. And 2 to 5 is going to show the columns, and result will be 8 , 4 bit total 32 bits. And the row selection depends on both data and the key, why Is that, lets go back, and look at the row selection ,this row selection depends on both data and the key. The key number over here is depended on both data and the key. Not only the row selection, even the column selection depends on the key information and the data, in other words, when data changes, substitution will remain the same ,but the rule, substitution table remains the same, because these things are going to define, substitutions that we are using, right now we have fixed substitution ciphers, fixed key S boxes, fixed blocks. In that 54 bit block size is fixed,56 bit key size is fixed , the substitution table and other table that I am going to use ,those things doesn't change depending on the data, whatever the key you are using, whatever the table that you are encrypting, you are using the same table. In other algorithms these things aren't going to be fixed. It will be dynamic. And it is going to give greater security for those algorithms. And there's a example over here, and we understood the example, in the example, selection of the row and the selection of the column.

DES key schedule calculation

Remember key generation we need to have left circular shifts for half of the bits. Key shifts, one half is shifted circularly left and the other half is also shifted circularly left, but we do not say ,how many bits we are going to shift, because that Is going to come, in standards again, it says, first bit one round, shift, in the second round one and in the third round, 4,5th and so on one, and then right one again its 2, 2 ,2 and last one is 1, as an implementer you are not going to come with your own sketch. In other words you cannot say that you are going to use these things and these shifts, in first shot I am going to shift this thing 4 bits and in the third round I am going to make it 6 shifts and so on. If the destination knows about this, they are going to shift it back anyways, and that is going to be your own key schedule basically, you may think that ,this is going to improve your key schedule, you might be right , but you might be wrong also, and that wrong side is dangerous. You don't know , whats going to be the overall changes of that effect. Actually what we are doing in the security world is that ,we are using the actual standards, so whatever the standard will be telling us to do, we are going to do it. We are not going to change a bit, I am not saying that you are doing wrong by changing the stuff, but what I am saying is you may not be knowing the implication stuffs. If you don't know it, then don't touch it. Because this is security, and national security may be danger for that matter, or may be your personal security will be in dangerous the decryption .the subkeys in reverse order and you don't have to change anything. We are going to do that basically, first round will have last round, in decryption we are going to use ,last round key, first round, that is the whole idea. You may ask this one, that decryption is nothing but inverse of encryption ,so why don't we have that inverse of expansion table. Why don't we have inverse of S box table, why don't we have inverse of permutation table 2. Answer is you don't need it, this is a feister structure , in this structure we don't use inverse of any function, all you need is main function and reverse the order in subkeys. In DES while implementing algorithms we can use the same implementation for decryption ,it doesn't require totally different decryption, but we will see in some other algorithms that's not going to be the case, encryption and decryption are going to be

implemented separately, because the decryption part going to be total different than. It requires inverse of everything that you use in the encryption ,but DES it is perfect , while implementing the encryption you can easily, you can easily decrypt the ciphertext ,so let me ask you if you use these keys in reverse order, lets go back and look at the keys, I generated the keys while I am encrypting the data, first round key, second round key, third round key ,like that, based on shift, 1 shift, 2 shift, 2 shift, 1 shift and so on, how are going to decrypt on the decryption side. You are pretty close, but right here I didn't give idea for decryption, you are not going to create the same subkeys by using these schedules, remember, In the first round ,in the encryption you are going to generate the last key that you generated. If you use the same schedule, the last key that you generated will be the last key. It will not be the first key that you used for the decryption. Then generate all the 16 keys which is not going to take too much time and start generating the last key first , second key second and so on. Or come up with the new schedule yourself , but you are going to change the schedule of the shifting yourself, in a way that the first key that is being generated will be the last key that is generated in the encryption, that will be a nice question, so think about it. That is the DES decryption ,where we are going to use the subkeys in reverse order.

Triple DES

Instead we are going to use triple DES, it is a standard encryption algorithm for US government, what we are doing is, plaintext will be encrypted with K1 and will be decrypted with ,another key K2 and the result is going to be encrypted with K1 again and that's going to be the cipher text. So lets analyse this position, I am encrypting the key with K1, im getting A and K is getting decrypted with K2 so am I going to get the plaintext back??.. no if you are using different key for decryption then that is encryption. What I am doing is I am encrypting again and again, and I am getting the cipher text back and that I one version ,but actual version of triple des is, K1 = K3 , in other words we are going to use 3 different keys. They are going to work with 2 keys or with three keys , in other words, triple DES with two keys or triple DES with three keys. {question from a student}. No they are using it just for fun so that they can name it EDE. So they say we are using triple DES in EDE mode. So , it doesn't give any advantage ,but you may say that if there is any problem in the encryption side, may be it is not that weak in the decryption side , and it is going to go make it secure ,and in other cases it is secure and from the security point view what you are doing is you are increasing you security, because you never know this is crypto ,you may discover some error In the encryption and decryption is still secure. {question from a student}..you can use it for some other algorithms as well ,you may call it with whatever name, so they are calling it EDE mode, encryption-decryption-encryption mode .so what is the decryption side this K1 is going to be used first, and K2 for encryption , and again K1 for decryption so this is DED mode. So that is tripe DES, so instead of so instead of using Double DES we are going to use triple DES. So there is not known attack for triple DES { question from a student}..second one is the decryption side of that one, we are decrypting, so we have the encrypted data and with triple DES we don't have , don't get confuse with the man in the middle attack..meet in the middle attack is what we use here. So don't get confused. Try to do it yourself, if you can do any meet in the middle attack yourself. I will give A, you don't have to take any exam, it is more difficult and not practical. If you are trying to get all the key possibilities with this one then ,again with the other one, to get B ,and then you are going to do the same thing with k1 to

get it from A. so generally what you are doing in implementation is we are not going to see the blocks differently.and that's why you are not going to get these outputs ,so this is one close box and one close DES implementation. During implementation you may not discover those numbers, but in meet in the middle you don't have any other option basically. In other words we don't have any known attacks for triple DES. It is still safe. But tripe DES, I am using 3 times 64 or 3 times 56. But the other one that is used , I will say it is not a big deal. But anyway as far as I know no one has managed to break it, that shows that this guy is still scare. We are going to AES for most of the application and it will be lost in few day , okk so triple des, has to compare with double using this, why not double des, one of the thing is meet in the middle attack, which produces x for encrypting the plaintext ,so it is called meet in the middle attack and we are using triple DES, we are using either two keys or three keys, if k1=k2 then we can vote as single des, and the whole triple DES is standardized in ANSI X9.17 and ISO8732, as this is standard so these are the standard documents for triple des. And we don't have any practical attacks .

Triple DES with three keys

And actually, those stuffs are the triple des with three keys and plaintext will be encrypted with key 1, and the result is going to be encrypted with key 2 and again with key 3. And the result is going to be my cipher text and we are using this triple DES in PGP. What is PGP?..Pretty good privacy, it provides, email security, it is important application which is used in triple DES and what is S/MIME?... secured email, multipurpose some extension ,and those extensions are securing the mail , protocol that we are using in mail is called SMTP and these makes SMTP secure and it works on top and one of the algorithm that you are going to select is triple DES and one of the major application that you still use, triple DES in there implementations basically.