CF mode

and we said that this ciphertext block will be input to the next plaintext block , here what is going to happen is IV is going to shift to the left, so that there will be some space for S bits. And that S bits will be filled with the previous ciphertext block. So what Is the value of IV here, shift it to and get the previous ciphertext block value . Are you following me, so that is going to be encrypted. And I am going to have the same DES output with the same key that I am using and the result is 64 bit cipher text and I am going to select S bits from that cipher text, most significant bit for example and second, bit from the plaintext block and I am going to create the second cipher text block and so on. So the explanation will be something like this, previous cipher text block will be in the IV input and it will go into the result of the encryption and from the result of the encryption, I am going to select S bits. To be used as key to encrypt the S bits plaintext block. To get the S bit cipher text block. If you look at here, I said that in the vernam cipher I said that I am generating the key, so if you look at it, the XOR ing part the S bits, that part this and this, I am actually generating the key, Im using the key using DES encryption. {question from a student}. As we said we have one bit and 8 bit, for this thing, we are going to select most significant one bit or most significant one byte so ,it is the most significant, you can select and forget the others . so this way what I am doing is I am using DES, algorithm as stream cipher, remember we use DES algorithm as Block cipher. But iam using DES algorithm as stream cipher here .by generatin key for the stream cipher, but actual encryption, but actual message encryption is not this one, but the actual message encryption is the XORing ,XOR ing is the actual message encryption. This encryption creates key for vernam cipher, that's what basically. But remember at the beginning of the chapter I said that, we may be using block cipher as stream cipher. We are going to replace with DES with AES. With blowfish it is going to work , this encryption, although this stuff is defined for encryption algorithm, it is all the block ciphers that you may use. so this encryption right here is AES encryption. So when we use the AES algorithm we are going to use same modes with the AES algorithm .although I said these are designed for DES. so what about the decryption part, decryption is not easy as there's shifting over here and XOR ing over here. In other words I am receiving the whole cipher text message and how I am going to decipher it. I am going to use the same IV and I am going to use the same decryption part, I am going to distribute the IV as well or this is not going to work. By using the same IV, im encrypting IV and generating the key again. So what is that output right here??..what is that output ,it is the same as the key of the vernam cipher,.i have to generate the same key ,right, I have to generate the same key. So I am selecting the same key by selecting S bits or J bits or whatever. And XORing the C1 with that key. In the vernam cipher how you are going to get the plaintext back ,you have to XOR the cipher text, with the same key that you use to get the plaintext. Remember the property of exclusive OR, when you XOR the two samething, the result is 0, if you XOR,cipher text with the same key that you use, and cipher text is the plaintext Exclusive OR key,and this key will be 0 and P XOR 0 , will be P, so recover the plaintext block back. I have to generate the same key that I used in encryption .and I am doing that using the same plain key and the plaintext and I m XOR ing C1, with same key and the result is first plaintext block. Ok for the first plaintext block it was easy may be, lets have a look at the plaintext block two ,I am shifting IV to the left, like I did in the previous block. {question from a student}. Whatever the IV that you use in encryption you will use the same in decryption. You are distributing it by two keys, and you have to know these two entities right. And IV also should be known by two entities, so when you are

distributing the keys, you are going to distribute IV as well don't forget that one,without the IV you cannot generate the key, you must have the same key and the same DES key cipher ,to generate the same key for decryption. Ok I am shifting IV to the left bit , and the shift with C1 ,we have the new IV, and the new IV, is equal to this IV and I am using this key, and the result is encrypted and I am using the same key , that is used for the encryption. {question from a student}. Algorithm is there, so the IV provides the security ,these are known only by two entities ,remember in symmetric algorithms security was based on secrecy of the key, same thing over here. Secrecy of the key and we may discuss that actually, do you think we must keep that IV secret. Lets jus think that I am not keeping it secret , they know my everything, so are they will be able to generate the key over here?...because the don't have DES key information, secrecy again depends on the key. In reality in real world we are keeping the IV secret. So while we are distributing the secret key we are distributing the IV along with it. So when we say key distribution we say key distribution and necessary IV values. If you look at the details of SSL protocol ,if you had taken NW security course from me, in SSL protocol that is used for web applications, while they are generating the keys, secretly or securely, they are also generating IV securely, so they keep the IV secret and not making it public and that it is the part of the key generation process. {question from a student} . we are going to solve that problem when we go into public key cryptography. We do solve, key distribution problems, if we don't solve, we don't have to explain all these things basically. Because everything depends on the secrecy of the key and rest depends on everything. Just lookat the last block and the last block says the same thing and it says decryption. It is decrypting the last block and why it is decrypting it instead of encrypting??. This is wrong and it should be encryption, what you are going to do is you are going to change every block into encryption why iam pointing out this ,we are talking about decryption process. But we are using encryption process , in DES if we use encryption you don't need the decryption part of this. Decryption doesn't require decryption, instead it requires encryption. And that is why these things are good for applications and you don't need the decryption part of the algorithm and all you need is the encryption part. In encryption you can encrypt or decrypt the messages, in DES, they didn't ask too many things from you, if you see the implementation part the y didn't ask too many things from you. What was the only change in decryption in DES. Only moving the Subkey, but when you look the AES encryption ,AES encryption is completely different ,when you implement AES encryption then you have to implement different decryption. But if you are implementing AES in CFB mode you don't need the second decrypting part. Its going to save lot of time and effort basically. So that's why these guys are pretty good guys for fast implementation ,because these guys the actual implementation will be only in this part. Lets say we have a customer and we have generate keys and he requires stream cipher. We generated this and gave it to him ,he saw it he liked and he really like the encryption part of it also , but he starts complaining about the performance. He said do something on this one and make it fast what can you do??. This speed thing is very important in cryptography, in this cryptography the most demanding thing is the crypto algorithm. They are going to be used in memory process, processors, chips, CPU all over. If you can make it faster then it will be much much more better. Lets move the feedback from this point to the other point , right here, so instead of S bit of the key , I will get C1 ,to this place. Does it going to change the things??.. why do you think that is going to make things faster?..XOR ing the most fastest thing in the computer implementation ,if you using it in all your services, its going to make your services more and more faster believe me.{question from a student}. We have to select some bits and that is the meaning of stream

cipher anyways. Either its going to be one bits or 8 bits. For this to generate the second key we have to wait, for the completion of thecipher text block ,because if I don't get the cipher text block. I am not be able to generate the second key, because second key is the function of cipher text block .but if I get the feedback from here, I don't have to wait for ciphertext . I can precalculate the things before anything, before even plaintext is existed. So keys are going to be ready and I am going to be saving them some where and Im going to XORthe plaintext block with the key ,so that the total encryption ,will be depending on the course of XORing that is much faster, believe me, if you precalculate keys, at this point ,all you need is get the plaintext block and XOR it .and the time will not be effected and the overall process, the actual time required will only be the XOR process, nothing else and that thing is called OFB Output Feedback Mode.

Output Feedback (OFB)

The change in the CFB, will give us another mode which is called OFB mode. You see the feedback is moved to the output of the encryption. That is DES encryption by the way and i don't have to wait for the C1.and rest of the algorithm is same as the CFB, the only difference is instead of getting the feedback from this point ,I got it from this point , so that I can pre calculate the keys, that is going to be used for encryption and I am going to have enormous amount of time. The actual encryption will be only XOR and nothing else will be there. and at the last block will be decrypted from here. Reason ?? its going to use only encryption .even thought OFB is going to use only the encryption, we don't need the decryption side of the algorithm even though the OFB, encrypts. You can see suddenly these guys encrypted and so you can use these OFB mode, for fast encryption ,if the application requires fast encryption , lets say for the physical encryption we are using 2 Gb encryption and 2 GB per sec speed required for you fiber optic cable. And you are going to use encryption algorithm to provide physical security. And performance of your encryption is 10 kb per second for example your physical bit is going to be 10KB per second. And they are going to fire you if you do that, for that fast environment. The simplest thing is XOR ing ,its not going to effect .it will reduce for the capacity of you communication link basically. So these kind of stuff is more suitable for those kind of speed required applications. And they are going tomake things little bit better, instead of shifting and those kind of stuffs, they are going to simplify the things little bit more better. That thing is called Counter mode

Counter (CTR)

This is the actually thing that is used in todays technology by the way instead of OFB. We have here Oi ,it is the encryption of the counter algorithm, this could be DES, this could AES with key K. instead of IV iam going to use counter over here, something like this, IV,I am going encrypt the counter value with key and result is the vernam cipher, this is still a stream cipher and I am going to select one bit or 8 bit or whatever, and XOR It and I am going to get C1. There is no feedbacking over here and what I am going to do is I am going to define the relation ship of the counter ,so in the next block the counter value will be incremented by what??? Lets say that you started with 5 and right here its going to be 6. In OFB, changing depends on the output of the algorithm, in other words it is more random basically. But this one is more random ,as I am going to encrypt it anyways and I am creating, so I don't care input is that random or this random, as long as I am encrypting, I am getting keys out of that encryption , im okk. So

this way things are much easier, you don't have to deal with feedbacking ,or shifting and so on.and you can precalculate the keys if you decide about the counter and the keys. DES key I should say, when you do this you have keys ready all you have to do is select the bits from the plaintext and get the plaintext block basically, so this is the basic,this is the mode for high speed application and decryption it's the same thing. You are going to use the encryption, counter and the same key, by using this you will be able to Decipher all the cipher text blocks basically, so it is the easy thing so these are called mode of using the algorithms and when we have this symmetric encryption algorithms .we are going to use them in one of the modes so whenever we see this symmetric encryption algorithms ,you are going to see the name of the mode that is used.for example CBC - DES , OFB –AES and CTR- DES something like that, you are not see these for example, hey we are using the DES algorithm what is the meaning of that anyway. I am using DES algorithm which mode that is my definition in my manual , that is actually ECB by default,

Remember ECB is the straightforward use of the algorithm ,other than that they can say ECB-, CBC- ,CFB-,OFB – and so on so every time they are going to use this. So most important mode that you have to, that we didn't discuss it from CFB is lets say that there is one bit error in the cipher text block and what is going to be the effect. Well it will propagate, will it propagate until the end or it wil stop somewhere ,you may have to think something like that ,because I may ask lot of question like this, these are multiple choice things, I can give this thing and say give this thing and stuff like that, errors and what will happen if there is error in CFB mode and how many bits are going to be affected. So there might be lot of good questions like this for each mode actually. some of them are not that easy by the way, so think about it and focus, on u will get, like I said one bit error on C2and show me the effect. So when you decipher it, P2 will not be not correct one right, so lets have a look at it again, C2 received with one bit errors, so C2 has one bit error, so that C2will definitely going to affect P2 but it will not affect P2, so P1 can be recovered with errors, P2with errors, if P2 with errors. Feedback comes from the C2 to the C3 and P3 will be effected , because its not a correct IV over here. And S bit shifts, how many bits are going to be shifted until , this C2 has to got rid of IV, if it is one bit that you are using then it is 64 bits. But this is going to go and affect the 64 bits. And you have to think that way basically, so you C2 goes to shift register and its going to be the last register if you are using CFB1, of the shift register and that is going to stay in the shift register, until 64 shifts and that means ,after this one 4 bits are going to stay with errors. You are going to recover them with errors basically . so you have to think this way basically. To answer these kind of questions and if you think that way, you can actually understand the algorithm much better. For all of the modes , you may think this way ,in the plaintext block what is going to be the effect of that one, so on, and any error in shifting over here and what are the effects. {question from the student}. This is feedback and in feedback, C2 will go to IV of the next, and 64 times if you use the CFB then C2 will be staying in the IV. Until you get rid of it until 64th shifting. Something like this,you guys have decide like this. And for CTR we discussed that one and what are the advantages and disadvantages of these guys is as you see CTR is pre calculated the keys, and these guys are much faster than CFB. CFB is a good stream cipher and you have to wait for the ciphertext for the plaintext blocks. But you cant precalculate the keys ,but three of them actually, vernam cipher, what we are doing is we are creating the key for vernam cipher, with CFB wit DES, With ECB with DES, with OFB, with DES, and the difference is OFB and CFB uses different feedbacks, one is the output of the encryption algorithm and the other is the output of XOR. CTR doesn't have feedback, all you have to do is you have to define

the relationships of the counter. And the value of the counter incrementing by one basically. It is pretty simple stuff as you see. Ok lets go back right here. As you see ECB is a straightforward, mode and it is used for secure transmission of single values. That is one block or two block of information . you are not going to use more than that and CBC, general block encryption iam going to call block encryption ,you can use CBC mode for huge message encryption basically, and CFB general purpose stream oriented transmission and authentication ,OFB, stream oriented transmission over noisy channels. Like satellite communication and so on , and CTR mode ,is block oriented transmission used for high speed requirements ,these and OFB both are used for high speed transmission, these are all the defined for DES algorithms. It doesn't mean that we cannot use it for AES, we are going to look into details later, in other words, the modes of operation is genera for symmetric encryption algorithm ,alright!!..that is I believe the end of the chapter.