Modes of operation

When we describe bock ciphers we are not going to use the block ciphers, that we described earlier. As encryption for DES. We are going to use different modes of using the algorithm. And everytime you see the uses of the algorithm ,in any security algorithm they are not going to call it DES algorithm, instead they are going to call it mode and DES. That application is used in DES in that mode. And we have different modes discussed for DES algorithm, these are the standard for DES and these are the documentation standard,four were defined for DES in ANSI standard ANSI X3.106-1983 and is called modes of use and they called 5 for DES and these modes are applied to different standards and are called AES, advanced encryption standards. And we are going to have 5 different usage of DES algorithm, generally distinct for all the block ciphers. So lets have a look at the modes of operation , the first mode of operation is called ECB (Electronic Codebook). In ECB, each block of 64 bit is encoded independently using the same key. Each block will be encrypted independently using the same key. So guys these are directly used in the algorithm ,so it is called Electronic Codebook(ECB). So what I am saying is, first of all , message will be broken into independent blocks, as this is block cipher and we are going to do it anyways. It will be 64 bits. And then each block is a value which is substituted like codebook, because block cipher is big cipher remember ,each block is encoded independently of the other blocks. That means current plaintext will be encrypted with the DES by using Key K1 and we are going to get current cipher text block. It's the same thing again you are going to get the plaintext block , then encrypt it with DES with key K1 and get the Cipher text block. We are going to use this for security transmission of the single values, why we will use only single values, lets have a look at the encryption so that we will get to understand why we will use single values ,so here we have plaintext , the first block will be encrypted with the plaintext and the result will be ciphertext. So I am going to get the second plaintext and encrypt it and get the second ciphertext. {question from a student}. Different key for the plaintext and not for the blocks. Plaintext consists of blocks and we are going to use the same key for the blocks, and we are not going to change the key for the blocks. P1,P2 P3..PN is the plaintext. I split the plaintext into blocks. And I spit the plaintext block and encrypt it with a key and get the ciphertext. And again the plaintext block will be encrypted will be the same key and the key is only for my whole plaintext {question from a student}. P1 and p2 are 64 bits blocksize, in the DES algorithm there are 64 bits block size,so I am going to do this until the last block, and the last block size might be 64 bits. So what iam going to do , padding ,so I am going to do padding and I am going to do it 64 bit and I am going to get the 64 bit cipher text block. {question from a student}. You can put 0 or something like that imagine this as binary files. So right now C1, C2, Cn will be my cipher text. And block plaintext and block ciphertext , I created and its going to be , and its going to be transmitted to the otherside and the other side when it receives C1 decipher it with the same key ,to get the first plaintext block .first of all remember this we are not transmitting this cipher text block independently. We are constructing the whole cipher text and we are transmitting the whole ciphertext to the destination. And when destination receives the ciphertext its going to receive it like the cipher text block, because we know that, cipher text block size is 64 bits, and we are going to take the first 64 bits, and then it is going to use the key that is used in encryption and it is going to use the decryption side of the encryption algorithm to get the first block of the plaintext back. And its going to go and get the second cipher text block. And then decipher it and get the second plaintext block until we get all the plaintext blocks. Okk?.. this is the

straight forward use of the algorithm ,if we use the algorithm like this, it is ECB. But what is the problem with ECB, we said we use this for single values {question from a student}. If you look at the block cipher we are assuming that especially in the DES. You are not going to find, any relation between plaintext and cipher text, any encryption algorithm that we are going to use in this course will not disclose any relation between cipher text and plaintext and cipher text and key. So that's not the problem, lets say that we are using DES, what could be the problem ,imagine that plaintext blocks in your plaintext and two of them are all same, what will happen., you see by checking the plaintext ,by changing the cipher text block itself, if something is repeated then it is repeated in the plaintext itself. The plaintext will be directly reflected to the cipher text value. That means, if P2 = P3, then C2 = C3. By checking C2 and C3 someone can say that or intruder can say that something is repeated in the text. So anything repeated in the text can be seen in the cipher text , although I am not saying, for example lets say that this repetition is number 8 ,I don't see the number 8 ,what I see is the two blocks, are the same blocks because to see the 8, to do that I have to decipher it, to do that, but the key is secret, in that sense we see the repeated blocks are same , but this reveals something could be valuable for some other applications, and permutations. I don't want to reveal those kind of information, if you want to reveal those blocks . one blocks or two blocks and those blocks are not the same, I can use these guys , that's why I said ,we can use these guys for the single values , means 64 bit values or 128 bit values. One block or two block, if your message is short by anyways, then this is pretty good method to use, but if your message is large then this is not good. Because in the large message they can be repetition and those repetition can be easily seen in the cipher text. Imagine if something went wrong when I encrypt the P2 and C2 will have some errors in it ,so what is the effect of errors in C2. ??..ok lets say C2 has some errors and it went to destination it has C1, C2 , P1, and if we Decipher C2 I am not going to, get P2, so error in C2 will effect P2. Is it going to effect P3, P4 , P5, no as these are all independent. That means independently encrypted in other words. So there is not error (…….) ,it means it doesn't effect next block or the block next to it and all. It does effect only in the current cipher text block. Imagine that there is error in P2, this time in plaintext. While I am typing it I made an error, what will happen ,what will be the effect of error in P2??..do you think its going to effect C2?. Anyways you don't know encryption outputs so you can speak random. You don't where C2 has some errors ,because C2 has some errors. C2 will be C2 ,when you decipher it, you are going to recover it with the errors. Its not going to effect anything basically, P2 with errors will replace with errors only. That's it, it does not do anything else.

Advantages and limitations in ECB

As I said, repetition in message as shown in ciphertext. If aligned with message blocks particularly with data such graphics, or some other things its going to be called code book analysis problem. This is a repetitive analysis basically and why we are having such problems ,because of the weakness due to encrypted message block being independent. Main use is sending feedback of data, I am going to go back now, lets go back ,it says weakness in the encryption because we are encrypting it independently, it considers as weakness, so how are you going to improve the system to avoid reputation ,what you are going to do??..we are going to take this output and we are going to feedback it to the second block, what I am going to do is I am going to XOR P2, with C1 and that's going to be the input to the C1,c2 will be XOR ed with C3 and C4, P4 and so on and here we are avoiding the repetitions, the problem with

here was repetition and the repetition in the plaintext can be seen in the ciphertext and I want to avoid that one and how am I going to do that??.. I am going to make the output of second function as the first output ,this way we are not going to , see any repetition and imagine, P2 is equal to P3, what is going to be the result in C2 and C3, do you think C2 and C3 are going to be equal, P2 will be XOR ed with C1 and input to the encryption algorithm is entirely different. It is not just P2 , for C3 input to the encryption algorithm is the function of C2 and P3 ,even though C2 and C3 are taken they are not going to equal to each other, and that's called CBC.

CBC(CIPHER BLOCK CHAINING)

Messages are going to be broken in blocks, but these are linked together in the encryption operation ,each previous cipher blocks is chained with current plaintext block what I am going to do is , previous plaintext block will be XOR ed with current Plain text block. And that is going to be the input to the DES algorithm and the key that I am going to use is the K1, to get the current cipher text block. What about it is not I-1 and it is first plaintext block algorithm ?? , if Pi is P1 and we don't have any previous cipher text block. Because 1-1 is 0 and we do not have C0. The first cipher text block that we have is C1 ,so I am having trouble with the first encryption, first block encryption ,what I am going to do is I am going to do the initial vector, there is another concept, in the use of crypto algorithm. And I am going to have 64 bit some random number. And I am going to assume that ,this random number is previous cipher text . but on the decryption side the person who is going to decrypt must have, same initial vector ,so we have two parameters now, known by the two entities, secret key , and the IV(initial vector) and when we are distributing the secret key, and we are distributing IV as well ,so secret and IV are the two parameters shared between, sender or the recipient , source or the destination . so another concept ,initial vector which will be the part of encryption process, and without the initial vector, you will not be able to encrypt it or decrypt it, because during encryption ,you need that IV. And the uses is Bulk data encryption , means it is used in large data encryption and that is why it is called bulk data encryption .so what we did is this, message will be split into small plaintext block P1. That will be XOR ed with IV, and that will be the input to the encryption algorithm with the secret key K and we will get the cipher text C1 and this C1 will be the previous cipher text for the current plaintext P2. Second block will be XORed with C1 and that will be the input of the encryption key and iam going to use the same key and the result is going to be C2. C2 will be XORed wit P3 and C3 will be XORed with P4 , eventually Ci-1 will be XORed with Pn to get Cn. So this is the actual usage for large encryption or bulk encryption. All the time you are going to use this one, so everytime you are going to see the usage in security algorithms, they are going to say they are using CBC DES ,not only CBC. All the time when we use that algorithm we are going tosay that we are using that algorithm. CBC is the best for the encryption and what about the decryption ?..what is the rule ?..lets see, C1 is coming , and I am going to decrypt the C1 with K and the result is what?.. result is P1 XORed with IV, right here the result is P1 XORed with IV. So how am I going to recover P1, again what I am going to do is , I am going to XOR, IV based on the same rules, because during XORing, two same things will get us 0 , as in XOR 1,1 is 0 , 0, 0 is 0 and 0 XOR P1 is similar to 1 XOR 0 which is 1 or 0 XOR 1 which is 1, so 0 XOR P1 is P1, so to get the P1 back I have to XOR the Decryption result with the IV that is used and that is why , both entities should share the IV information along with the keys other wise,this guy will not recover P1 and C2 now will be decrypted, with key K ,

and the result is ,what is the result ,you guy tell me,??.. Ya its P2, XORed with C1, to get P2 , I have to XOR that result with C1 ,so C1 XOR C1 is 0 ,then 0 XOR p2 is P2 ,so , its going to like this basically, you see decryption is not causing problems to us basically, so we can easily decrypt the CBC and ECB mode, so we can easily encrypt and decrypt, there is no problem and this is very useful as I said in the bulk encryption ,we are going to have some other modes as well.

CFB (Cipher Feedback)

In CFB message is going to be treated as stream of bits, as this is stream cipher. It added to the output of the block cipher and result is the feedback for the next stage and thus it is called cipher feedback and standard allows any number of bit like J bit, which can be 1 bit, 8 bit or 64 bit, generally we are going to use one bit. Because if you use 64 bit then it is called block cipher and its not stream cipher. One bit actual stream cipher and 8 bit is ok. If you are using 1 bit then it is called CFB 1 and the if you are using 8 bits it is called CFB 8 and CFB 64 will be 64 bit, but they are not going to use too much as too much doesn't make any sense anyways. If you look at the description of the algorithm , then it will be this, if you look at the encrypting DES algorithm with Key K1 and the result is going to be XOR ed with the current plaintext block. Because actual encryption is not the DES encryption but actual encryption is the XOR. Because, we are not encrypting the plaintext with DES. But we are encrypting the previous cipher text with DES. Remember the definition, in DES encryption ,we are encrypting pseudo random number. And that random number is used to encrypt the plaintext. Actual encryption is XORing, remember the vernam cipher ,there is plaintext, there is XOR and there is key. And the result is ciphertext and we said this is the most secure algorithm, and it is most simple, to make this algorithm unbreakable what we do is , we are going to break plaintext only once and what we are doing is ,we know the encryption output of DES is quite random and why don't I generate this key, with the DES algorithm, so what I am doing is this algorithm uses DES algorithm and DES will use another key K1, and input to the DES algorithm will be Ci-1, input to the previous cipher text block,this is actually CFB, this is actually vernam cipher and that's why it is called Stream cipher. So this way im generating the key using the DES encryption, but DES encryption ,plaintext is encrypted with XOR. This encryption creates the key that is used for this encryption only. For XOR encryption, so this is used for stream data encryption ,authentication basically. So what I am saying is this, if this Ci-1, or lets say if this , plaintext block if it is the first block ,Pi in other words. What will be Pi-1??... That Is IV, but this is not IV over here. This is the previous plaintext, but for the first one ,we are going to call it definitely IV. So we are going to use IV, in other words, a 64 bit random number. And that random number will be encrypted with the key K and the result, is 64 bit ciphertext, vernam cipher encrypts this 64 bit cipher text by using 1 it or 8 bit, what I am going to do is I am going have 64 bit text right now, and I am going to use 1 bit key. Or J bits, lets put it that way. I am going to select S bit from the plaintext and the result Is going to be S bits, ciphertext and that Is my first block. Alright , are you clear?..