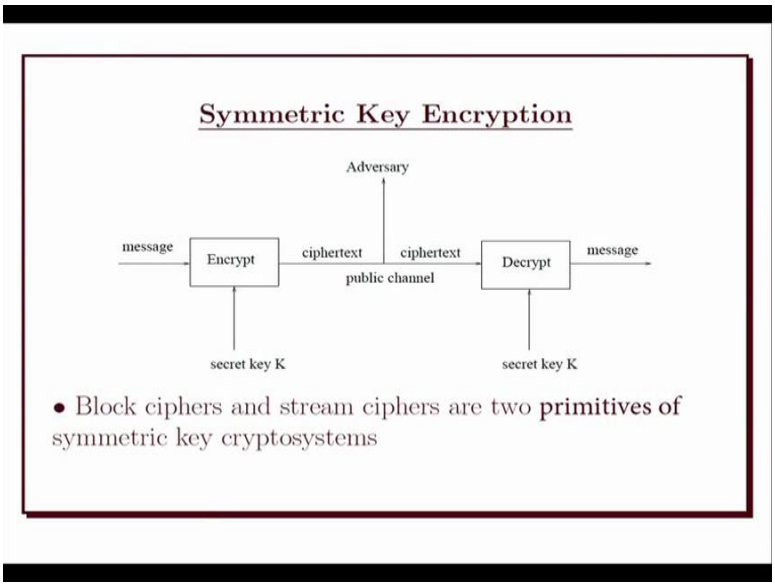
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 05
Block Cypher

So, we talk about block cipher, this is a symmetric key encryption primitives block cipher and stream cipher; please go to the slide please.

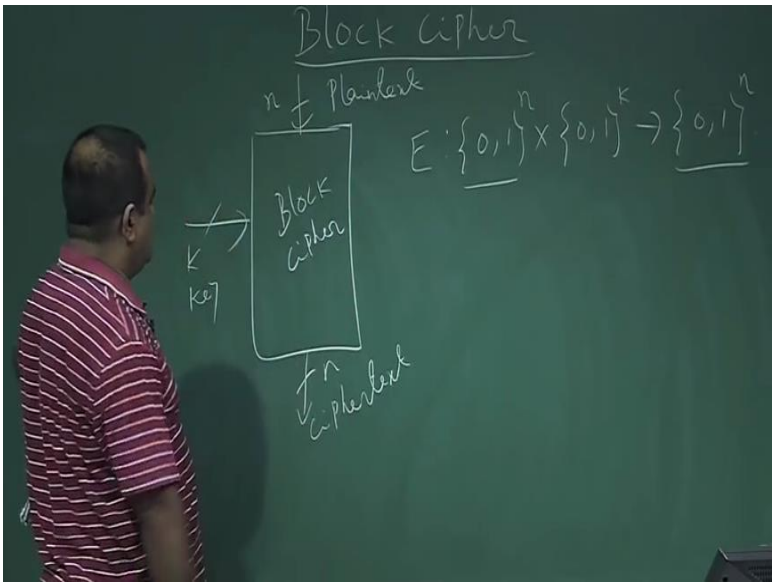
(Refer Slide Time: 00:29)



So, block cipher and stream cipher are 2 symmetric key primitives. So, in symmetric key there are 2 parties Alice and Bob, they have a shared with a common key K the secret key K and they want to communicate with each other. So, Alice is having the message. So, Alice is encrypting the message using this K and sending over the public channel and that this channel is very much captured by the adversary or the third party. So, this is the ciphertext after the encryption and this ciphertext is sending to the Bob the receiver and receiver is having the same key. So, receiver will decrypt the ciphertext using the same key and get the message.

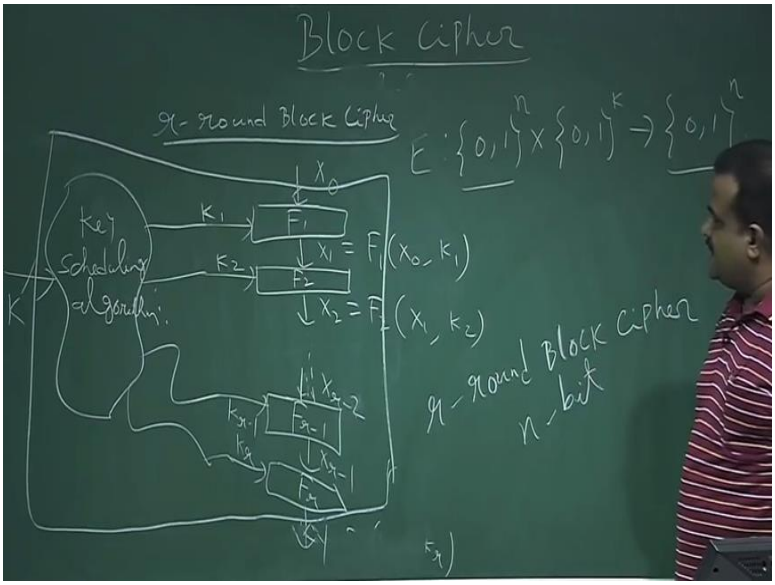
So, this is the typical setup of this symmetric key encryption and block cipher is a, block cipher is a primitive of symmetric key encryption.

(Refer Slide Time: 01:32)



In block cipher we have basically a black box. So, it takes the inputs say a plaintext and plaintext say n bit we have n bit plaintext and it will generate a n bit ciphertext, and it is taking a key which is a K bit, it secret key which is being shared between the Alice and Bob and inside it, if it is r-round block cipher. So, this is block cipher, this function. So, basically block cipher is a function it is encryption function, it has 2 input: one is plaintext, another one is key and it is giving us the ciphertext. So, plaintext is n bit ciphertext will be also n bit and the key is the K bit K binary bit.

(Refer Slide Time: 02:48)



So, if we consider the r bit r-round block cipher; so it is basically having r-round inside. So, this is the first round F 1. So, which is taking input as a plaintext which we denote by X or X 0 and which is having another input, which is called round key this is typically a first round key and then in the second round which is the input after this first round this is we can denote by X 1, which is basically F 1 of X 0 comma K 1, and which is a input of this second round function and for second round function we have another input which is the second round key K 2, and it will give us the output X 2, which is basically F 2, X 1, K 2, like this we continue, dot, dot, dot. So, this is the r F r minus 1 and this is F r ok.

So, for F r minus 1 we need another. So, 1 input is coming from the output of the previous round and another input is K r minus 1 and so this is basically X r minus 1, now this is basically X r minus 2, this is basically X r minus 1 and so this is K r, r th round key and this is basically called Y which is the ciphertext, which is basically F r of F r minus 1 comma K r.

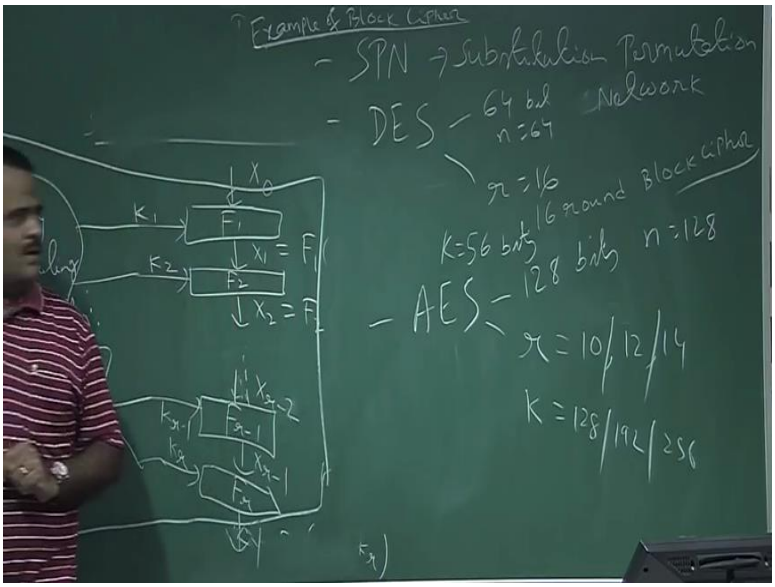
So, this is the plaintext and this round key is, are basically the inputs of each rounds respectively. So, where from we will get this round keys, this round keys is coming from a public algorithm which is called key scheduling algorithm; there is a public function public algorithm which is called key scheduling algorithm, which is taking input as a secret key K and it should able to generate the round keys K 1, K 2 how many rounds we want dot, dot, dot, dot, dot.

So, this key scheduling algorithm is taking the secret key which is shared between Alice and Bob, and giving us the round keys as many round we need. So, this is typically r-round block cipher; r-round n bit. So, plaintext is n bit ciphertext this is Y this is the last round output this is Y. So, this is also n bit. So, this function is called. So, this is taking input plaintext and another input is the key and inside we have a key scheduling algorithm who is generating the round key as much we need.

Then after this round; so first round input is plaintext, then it is taking this round key it is outputting the intermediate value X 1, then it is the input for second round function, then third round function so on and each time for each round function we need a another input which is called round key and which is coming from the key scheduling algorithm of this which is public function, which is a algorithm which give us the round keys from the

secret key. So, this is a typical r-round block cipher n bit block cipher; n bit block cipher means plaintext and ciphertext sizes in. So, there are some examples of block cipher like SPN.

(Refer Slide Time: 07:46)

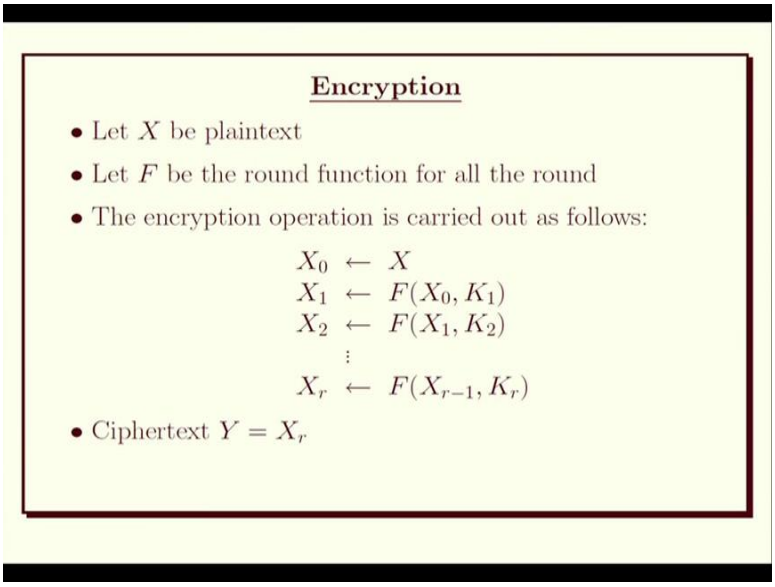


We will talk about SPN now, it is substitution permutation cipher, which is called substitution permutation cipher, substitution permission network SPN and we have this as some example of block ciphers; another one is DES; data encryption standard, it is a 64 bit block cipher. So, n is 64. So, n is 64 and r is 16. So, it is 16 round block cipher. So, we have 16 round function inside DES, we will talk about those details and the key size is 56 bit, 50 size K is 56 bits for DES.

And another example is AES advanced encryption standard, which is basically Rendell. So, this is also a block cipher, it is 128 bit block cipher that means, m is 128; so plaintext and ciphertext size is 128 bit, and r if it has different version r the round functions r either it is 10 round, 12 round, 14 round depending on the size of the key. So, if the key is 128 then it is10 round, 192 then it is 12 round, 256 then it is 14 round; and then we will talk about details of this block cipher, basically if we take any block cipher its structure is, general structure is basically same as this. So, if it is r-round we have r many round functions and we have the input size is the size of the plaintext if it is n bits. So, n is the input of the plaintext and we will get the n bit ciphertext and for each round we

need to get the round keys and that will be coming from a what is called key scheduling algorithm.

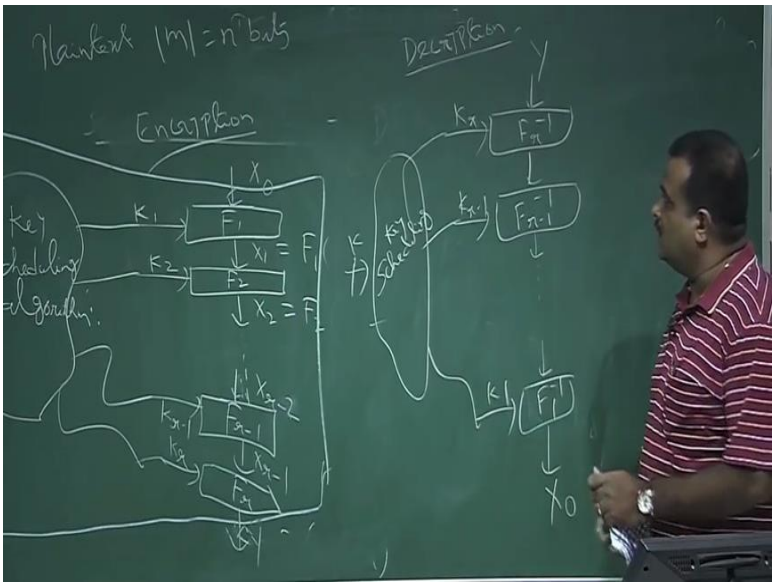
(Refer Slide Time: 10:30)



Now, go to the slide please. So, this is a typical example of a r-round blocks cipher. So, here we are assuming the all the round functions were same which is f. So, F 1 F 2 all are same functions. So, we have a plaintext x, which is we denote by X 0 then we apply F with the round keys K 1 and we got X 1, then we apply that value we apply on another F which is again F we are assumed all round function are similar. So, X 2 in this way we continue. So, finally, we got the ciphertext which is basically X r, which is the e output after r th round function and the for that r th round function we need a round key, K r rth round key and these round keys are coming from the from the secret key K, which is shared between Alice and Bob and with that we are applying the key scheduling algorithm on this.

Now, the question is how to decrypt such a block cipher? So, for decryption we have to assume F should be injective, it should be invertible; otherwise we cannot dictate these block cipher. So, for decryption we have Y from Y we need to get back the message X. So, we just reverse the process. So, we will start we start from the Y here, so Y is the ciphertext now; this is encryption of a block cipher and for decryption what we do? So, for decryption our input is this Y. So, Y is the input.

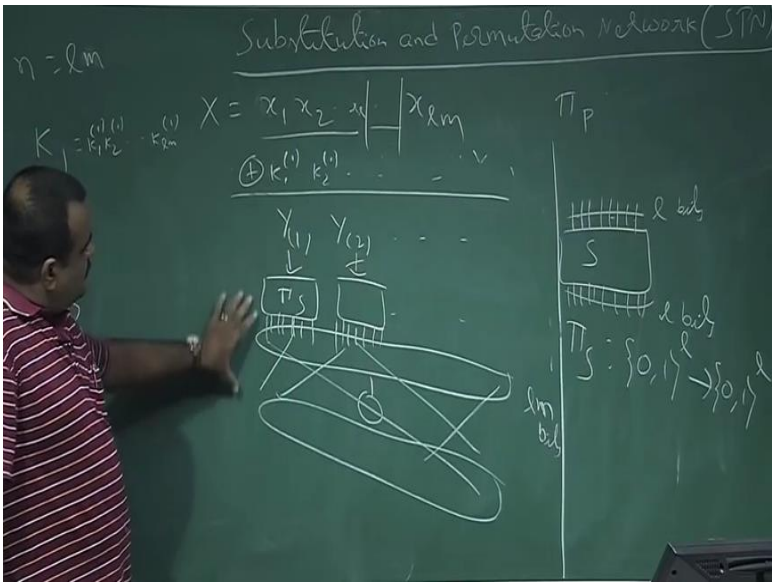
(Refer Slide Time: 12:21)



So, we apply that we have to go back here. So, for that we have to apply F r inverse if the round key, K r and then F r minus 1 inverse with ground key and. So, that is why this round function should be invertible, you should able to invert this round function it is injective function; so dot, dot, dot like this. So, we have this is F 1 inverse and this is K 1, so we should get X 0 or the X basically the size plaintext. So, it is the ciphertext. So, again this is doing by the receiver Bob.

Alice and Bob sharing the common key K; so Alice is encrypting this, so Alice has a plaintext m, which is basically n bits. So, what Alice will do? Alice will apply this block cipher on this n bit and get the ciphertext Y and Alice will send this Y to Bob. So, upon receiving the ciphertext, Bob will also Bob have that secret key k. So, Bob will generate Bob will this function is public everybody knows this key scheduling algorithm only thing the secret is the secret key K, K is not known to other party other than Alice and Bob. So, this is decryption this is Bob is doing decryption c r y. So, Bob is using the same key scheduling algorithm and getting the round keys and applying this on the respective inverse round function and get the plaintext. So, this is the typical decryption of a block cipher.

(Refer Slide Time: 14:53)



So, now we will discuss the block cipher SPN which is substitution permutation network. So, let us talk about an example of a block cipher. So, this is a substitution and permutation a neutral, this is the first example you will study for example as a block cipher; this is the in short it is called SPN. So, it has some substitution function and the permutation function. So, for this we have this plaintext. So, here plaintexts are x 1, x 2 like x l n bit. So, size of the plaintext is l n bits. So, n is equal to l into l m this is the size of the plaintext.

So, what we are doing? We are just dividing the plaintext into l bit each. So, this is dot, dot, dot x l. So, this is basically x 1, x 2, so x n; if we break this plaintext into l bit each. So, this is first l bit. then second l bit like this, you are blocking this then we denote this first l bit as x 1, second l bit as x subscript 2 like this. Now suppose we have S-box, which is a function which is taking l bit input dot, dot, dot this is l bits and which is taking a dot, dot, dot l bit output. So, basically S is the function, which is 0, 1 to 0, l to l. So, we denote this by say pi of this and suppose we have a permutation, which is basically permutation on these bits of l n to 1, 2, l m. So, we have these two.

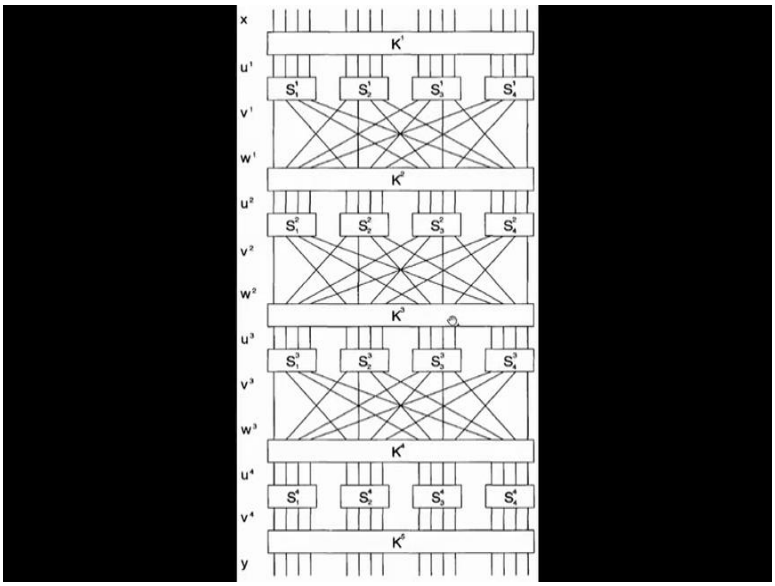
So, this is substitution permutation network. So, this is the substitution function we have a substitution function S which is taking the l bit input and giving us the l bit output. So, this is basically a function l bit to l bit function. We will come to an example and we have a permutation which is basically supplying the bits, so permutation from 1 to x n.

So, now, so for encryption what we will do we will just first apply S on this, S-box on this like this and this is the substitution operation and before that we need to apply the round key, add round key.

So, suppose we have round key say K 1, K 2 say we have say 5 round keys. So, what we do, before this we just add the round key. So, beta is adding. So, this is also l bit. So small K 1 small K 2. So, this is K 1 like this K lm like this. So, we just beta is X or with this. So, this is we can just use with this K 1, K 2 1 like this. So, beta is x 1 this is the round key operation we are doing; then after that we are making in to the blocks. So, maybe this is some Y 1, Y 2 like this. So, then we apply the S-box on it again it is l bit; we apply S-box on it, again this is l bit, all are l bit.

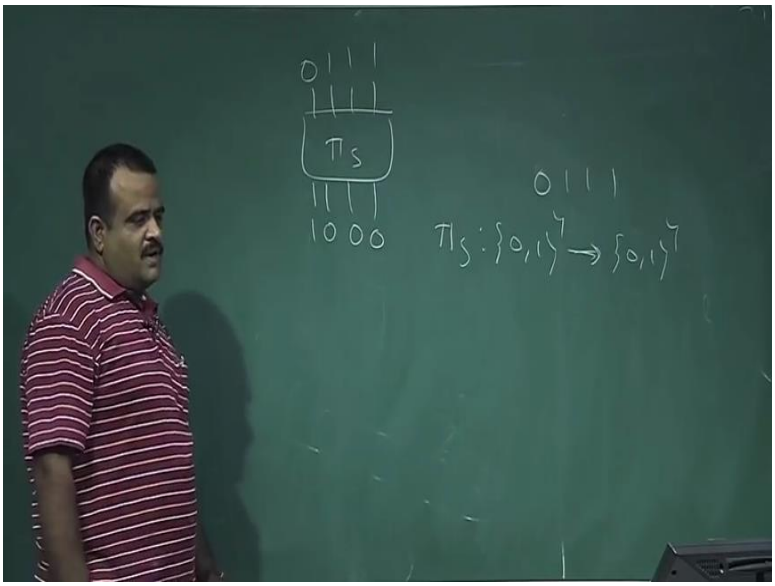
So, after this, we have again l m bit and this l n bit will apply the permutation we have pi of p which is just basically shuffling the bits, maybe this is which is going there, this bit is going here like this. So, this is the permutation we will do. So, this is the typical 1 round of SPN.

(Refer Slide Time: 20:03)



So, now let us this is can you go to the slide please. So, this is a diagram which you have taken from the Stinson book. So, this is an example where our l m r 4.

(Refer Slide Time: 20:20)



So, suppose we have 16 bit, n is equal to 16 bit number and this 16 bit we are dividing into. So, say x 1, x 2, x 3, x 4 like this x 16. So, x 5, x 6 like this.

So, what we are doing? We are breaking into 4 bit block. So, how many blocks we have? 1, 2, 3, 4, so x 15, x 16, x 14, x 13 is the last block. So, each is a single bit. Can you go to the slide please? So, here this is our plaintext x this is the 16 bit number. So, this we are blocking into 4 bit each and then we have a round key K 1 which is again a 16 bit number, so what we are doing? We are just doing the XOR with this x and this K 1 bits. So, both are 16 bit, we will do the bit wise XORing.

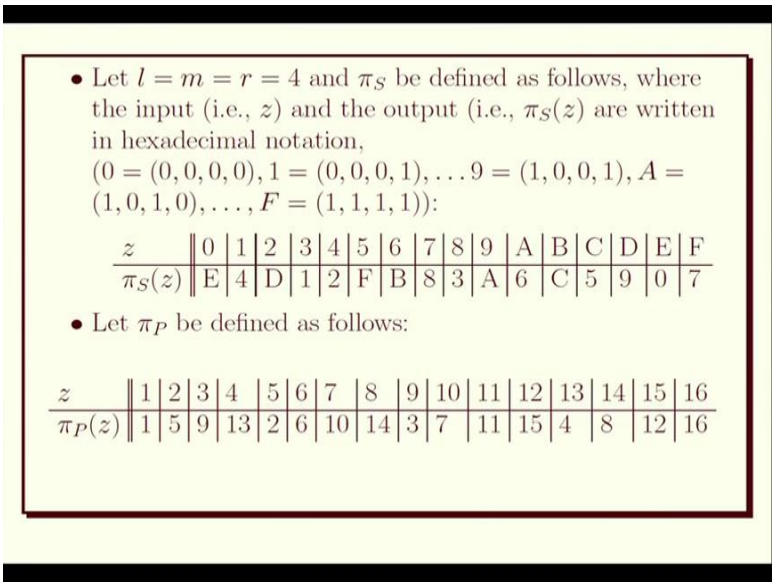
Then after that we are getting again a 16 bit number, this u 1 and this 16 bit number again we are breaking into 4 bit blocks. So, for this 4 bit, this 4 bit like this and then again we are we apply the S-box; we have S-box pi of S, which we will define which is taking a 4 bit input and 4 bit output. So, again this is 16 bit and this 16 bit we are applying that permutation. So, it is a permutation from 1 to 16, so it is just shuffling the position of the bits and this is a typical 1 round S P N. So, this is the first round.

So, similarly in the second round we will do the x 2 and then we again we block it into 4bit blocks and then again we will apply the S-box, and then again we will do the random permutation on it, that permutation we will define. So, this way we continue. So, this is a 4 round block cipher, but in the last round we will have extra we will have a, we

will have extra round add round key here, this is the extra ring, this is the extra operation otherwise this is a 4 round block cipher.

So, now we will talk about the S-box; so for this we need to have S-box, which is basically taking a 4 bit input, 4 bit output. So, 4 bit we can represent binary bit.

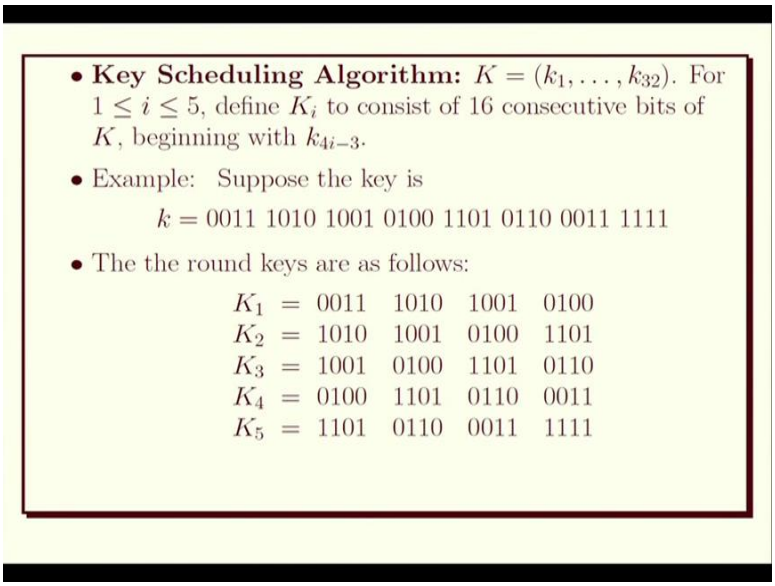
(Refer Slide Time: 23:18)



So, 4 bit number can be represent as hexadecimal number, so, this is the S-box. So, this is given an input is 4 bits, so any 4 bit number is can be represent in hex number. So, this is say 7, 7 is basically what? 7 is 1 0 sorry 0 1 1 1 this is basically 4, 2, 7 so that means, 0 1 1 0. So, if the input is 7, can you go to the slide please, input is 7 slides please. So if the input is 7, here table is given showing that output will be 8. So, the 8 means, in hex it is 8, so 8 means we have 1 0 0 0.

So, if the input is this output is this. So, this is the function for 0 1. So, 4 bit to 4 bit, function which can be very well given in this table. So, it is a hex table. So, in this table we just represent in the hexadecimal form, because 4 bit can be in hex form and this permutation is also given. So, this is 16 bit, n is equal to 16. So 1 to 16. So, this is the permutation; that means x 1 is going to 1 and this x 2 position x 5 is coming here, x 9 is coming here, x 13 is coming here, x 2 is going here. So, this is the shuffling of the bits.

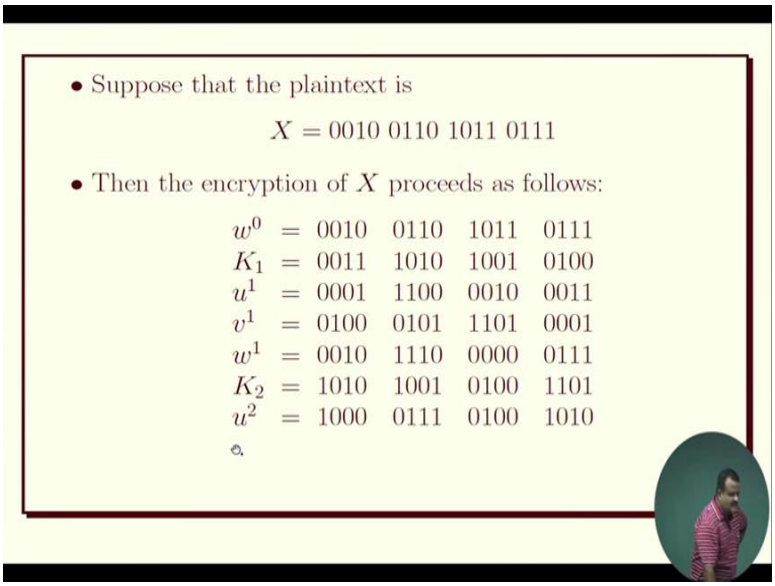
(Refer Slide Time: 25:18)



So, this is the SPN network, now we will take some example how it is working and before that we just discuss the key scheduling algorithm for this. So, in this case we need to have how many keys? We need to have 5 keys K 1, K 2, K 5 and suppose this is the secret key shared between Alice and Bob. Now from here this key scheduling algorithm is very straightforward, very simple for this cipher; what it is doing in the key scheduling algorithm we are just taking K 1 as a this fist 4 x bits, so this 1, this 1, this 1. So, this is the K 1 and the K 2 will be we do not consider the first one from the second one 4 hex bits. So, this is basically this, this, this, this. And K 3 will be similarly this, this, this, this K 4 like this.

So, this is the simple key scheduling algorithm from this round key, from this secret key this is the secret key sheared between Alice and Bob and this is the way how we generate the round keys; we need to have 5 round keys.

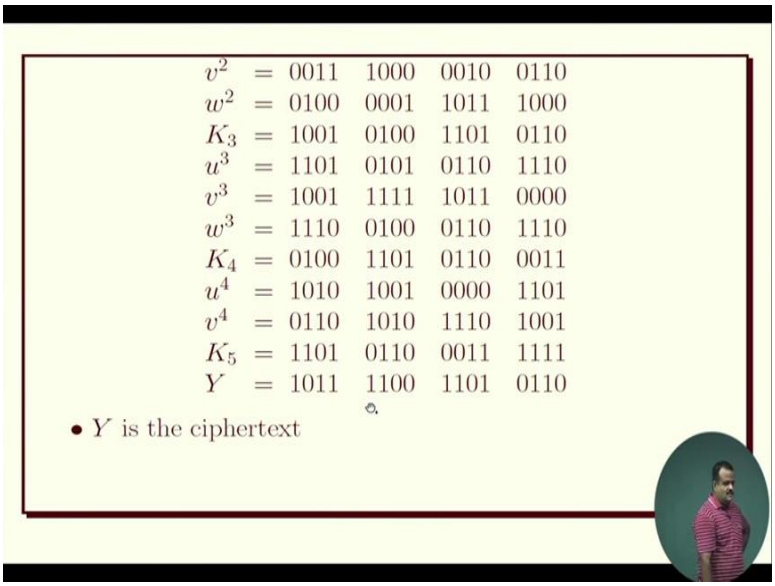
(Refer Slide Time: 26:36)



And here is an example if we execute the SPN. So, this is suppose our plaintext is this, which is again a 16 bit. So, what we are doing? We are just taking this plaintext and then we are just having the first round key K 1 and we are doing bit wise XORing with this. So, 0 0 0 0 0 0 and 0 1; I sorry 1111 is 0 there is no carry. So, this is just a bit wise XORing and then we are doing this bit wise XORing with this round key K 1 and we are getting this output and this output we will apply the S-box; we have S-box which is taking the a 8; 4 bit to 4 bit, so this is giving us the S-box output.

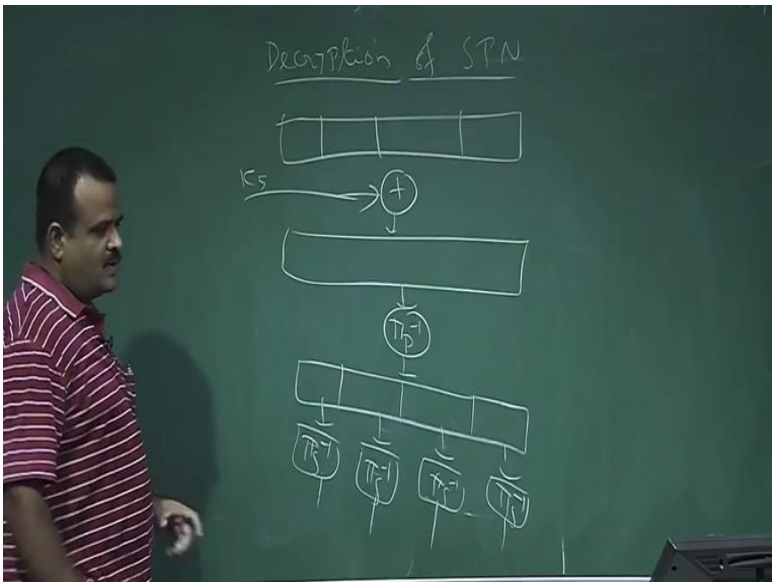
So, we got this v 1 and after getting v 1 we are shuffling the bit position. So, we are applying the permutation p which is basically changing the position of this bit position and we are getting this. So, this is the typical first round of a SPN and then we will go for the further round there are 4 rounds, then we will go for the next round which is again K 2.

(Refer Slide Time: 28:05)



Then a x 1 with K 2 then like this. So, if you continue like this we will just end with this the ciphertext Y and this is the Y.

(Refer Slide Time: 28:24)

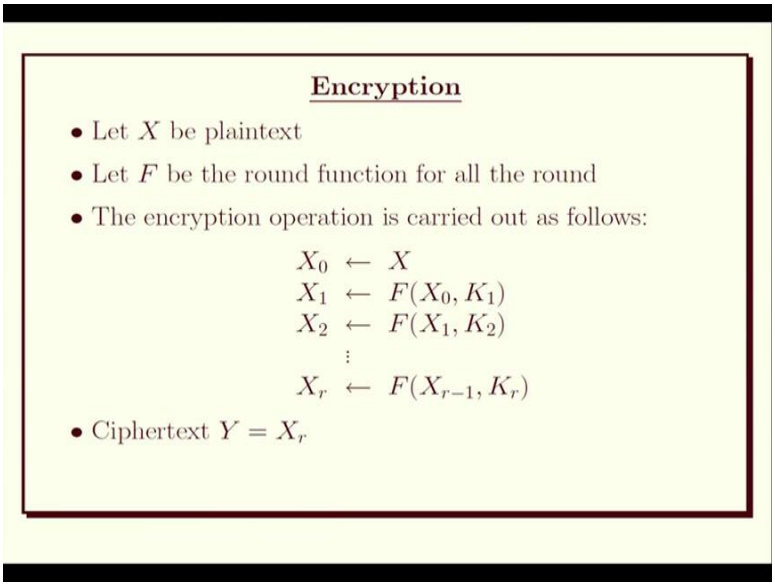


Now, question is how we decrypt it? So, this is the encryption, this is Alice is doing, now how to decrypt it decryption of SPN? So, for decryption Y is given, which is again a 16 bit what we do? We break it into 4 bits, now, we will XOR with the K 5 and then we have decryption is reveres. So, we have to apply the inverse permutation, we are using permutation. So, you have to apply the inverse permutation and then again we have to

apply the inverse S box. So, this is inverse S-box like this. So, this is a one round decryption of SPN.

So, we have the input, we just do the bit wise XOR with the round keys then we apply the inverse permutation and then we apply the inverse S-box, to go back to the plaintext. So, this is the way. So, is this inverse should exist and permutation we have the inverse and the round keys we all have also have to generate by the by Bob which is the receiver. So, this is the typical SPN network. So, this is the inversion.

(Refer Slide Time: 30:09)



So, this is a general structure of a stream block cipher where we have this round functions and we have the round keys, we have the mechanism to generate the round keys, that is the key scheduling algorithm and later on we will discuss the DES, which is which is another block cipher and then when we talk about AES slowly.

Thank you.