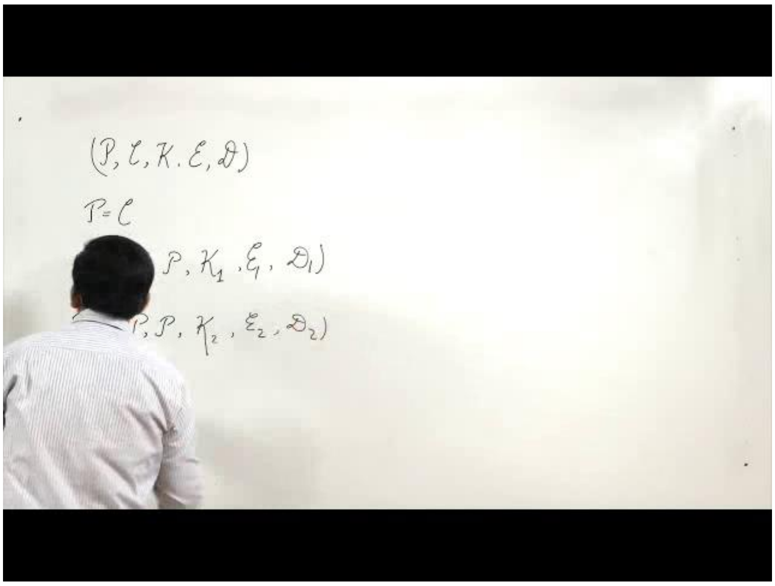
**Introduction to Cryptology**
**Dr. Sugata Gangopadhyay**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Roorkee**

# Lecture - 06
# Product Cipher, Block Cipher, Modes of Operation for Block Cipher

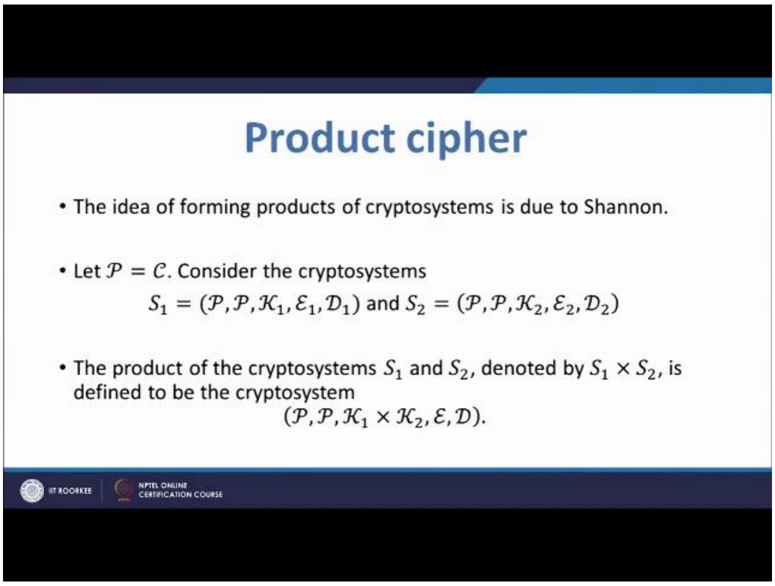
Welcome to week 2, Lecture - 1. We start from product ciphers. Idea of forming products of crypto systems is due to Shannon.

(Refer Slide Time: 00:44)


$(\mathcal{P}, \mathcal{C}, \mathcal{K}. \mathcal{E}, \mathcal{D})$
$\mathcal{P} = \mathcal{C}$
$\mathcal{P}, \mathcal{K}_1, \mathcal{E}_1, \mathcal{D}_1)$
$\mathcal{P}, \mathcal{P}, \mathcal{K}_2, \mathcal{E}_2, \mathcal{D}_2)$

Now, we know that a crypto system can be specified by P, C, K, E, and D, where P is the set of plain texts; C is a set of cipher texts; K is the set of keys; E is the set of encryption functions and D is the set of decryption functions. Now, we consider, for simplicity, the situations where P is equal to C, and we consider a cipher crypto system S 1 which is denoted by P, P, then K 1, the set of keys for S 1, E 1 and D 1. We consider another crypto system S 2, which has a same plain text and cipher text sets, and these sets are equal. Then we have K 2, then E 2, and D 2.

(Refer Slide Time: 02:07)



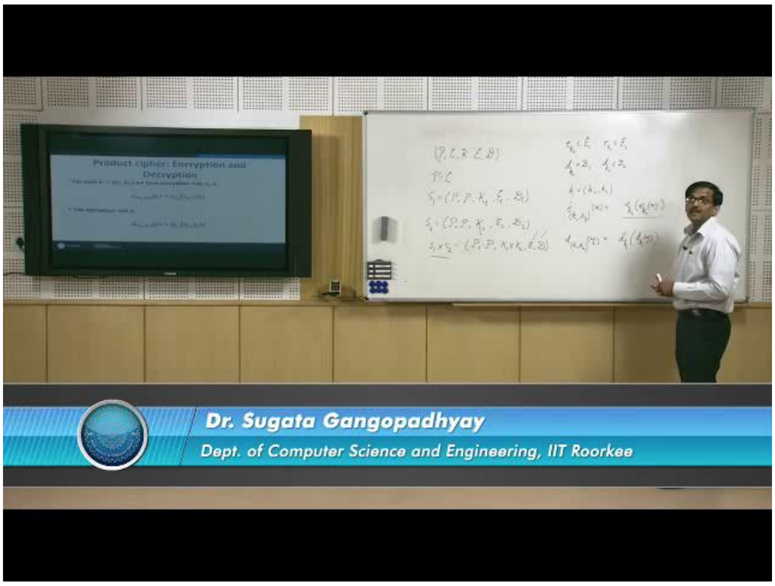
## Product cipher

- The idea of forming products of cryptosystems is due to Shannon.
- Let $\mathcal{P} = \mathcal{C}$. Consider the cryptosystems $$S_1 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_1, \mathcal{E}_1, \mathcal{D}_1) \text{ and } S_2 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_2, \mathcal{E}_2, \mathcal{D}_2)$$
- The product of the cryptosystems $S_1$ and $S_2$, denoted by $S_1 \times S_2$, is defined to be the cryptosystem $$(\mathcal{P}, \mathcal{P}, \mathcal{K}_1 \times \mathcal{K}_2, \mathcal{E}, \mathcal{D}).$$

According to Shannon, the product of these 2 crypto systems S 1 and S 2 is written as S 1 cross S 2 and we have a crypto system, P, P, K 1 cross K 2. by this K 1 cross K 2 we mean the Cartesian product of the set K 1 and K 2; that means, that it is a set of all ordered pairs of keys.

First one coming from the keys set of the first crypto system, and the second one coming from the key set of the second crypto system, and then, we have E and D which we will have to define. So, we see that S 1 cross S 2 which is a product of S 1 and S 2 that is P, P, and then K 1 cross K 2, then E and D. Now the question is that how we define E and D, which is given in the next slide - this one. So, encryption and decryption of product cipher. So, we have to know how to define E and D.

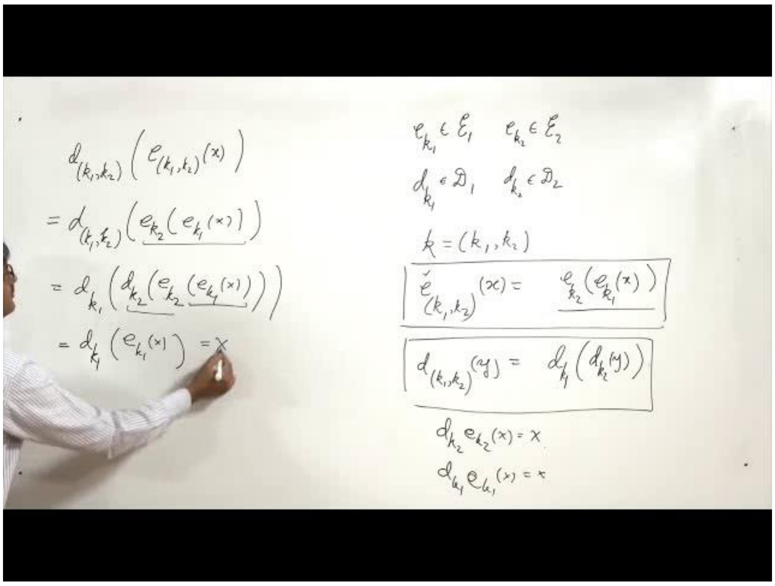
(Refer Slide Time: 03:41)


Product cipher: Encryption and Decryption
Dr. Sugata Gangopadhyay
Dept. of Computer Science and Engineering, IIT Roorkee

In the case of individual ciphers S 1 and S 2, the encryption functions were like this e K 1 belonging to E 1 and e k 2 belonging to E 2. Whereas, the corresponding decryption function d K 1 belongs to D 1 and d k 2 belongs to D 2. Now we have a combined encryption function for the product cipher we call the ordered pair of the keys as k. So, that is K 1 comma k 2 and the encryption function is labeled by k which is ordered pair K 1 k 2, and it operates over x, the plain text in p. So, it is x, and then we have a sequence of functions, the first one coming from E one that is e K 1 applied over x, and then e k 2 applied over e K 1 x. So, this is the result of the operation of e K 1 k 2 over x in the product cipher S 1 cross S 2.

Now, the question is that what is a decryption function? The decryption function D; again, K 1 k 2 applied over y will be equal to d K 1 applied over d k 2 y, where y is an element of C - that is a cipher text, and in this case C and P are same. Now, we know that if we apply encryption function over a plain text, and if we apply a decryption function with a same key over the cipher text generated from that plain text, then we should get the plain text back. Now we have to check whether that property holds for this new definition.

(Refer Slide Time: 06:24)



$$d_{(k_1,k_2)}\left(e_{(k_1,k_2)}(x)\right)$$

$$= d_{(k_1,k_2)}\left(\underline{e_{k_2}(e_{k_1}(x))}\right)$$

$$= d_{k_1}\left(\underline{d_{k_2}(e_{k_2}}\underline{(e_{k_1}(x))}\right)\Big)$$

$$= d_{k_1}\left(e_{k_1}(x)\right) = x$$

$e_{k_1} \in \mathcal{E}_1 \quad e_{k_2} \in \mathcal{E}_2$

$d_{k_1} \in \mathcal{D}_1 \quad d_{k_2} \in \mathcal{D}_2$

$k = (k_1, k_2)$

$$\boxed{\check{e}_{(k_1,k_2)}(x) = \underline{e_{k_2}(e_{k_1}(x))}}$$

$$\boxed{d_{(k_1,k_2)}(y) = d_{k_1}(d_{k_2}(y))}$$

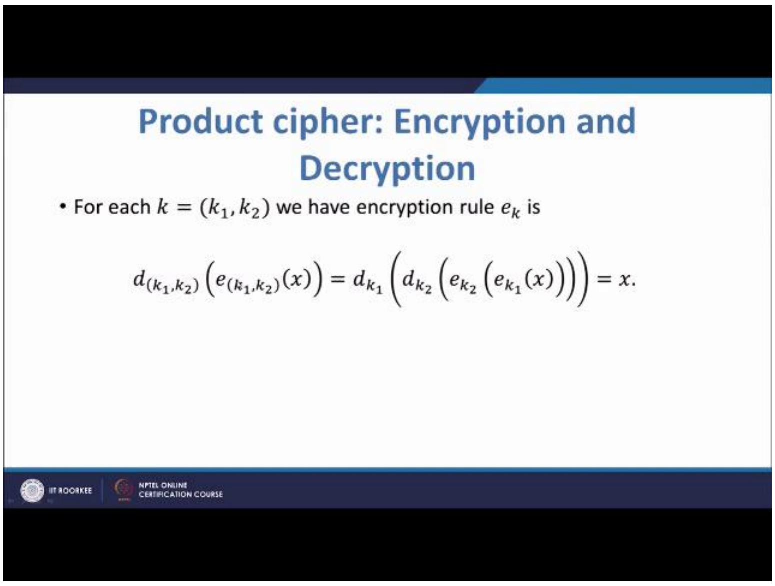
$d_{k_2}e_{k_2}(x) = x$

$d_{k_1}e_{k_1}(x) = x$

So, let us see. Suppose, I take d K 1 k 2 applied over e K 1 k 2 x, then by using this rule I can write d K 1 k 2 applied over e k 2 of e K 1 x. Now, we use the second rule. Now, the argument of the decryption function is the sequence of encryption functions; therefore, we will write d of K 1 applied over d of k 2 applied over e k 2 e K 1 x; I close the bracket, I close the other bracket, I close the last bracket.

Now look at this - this one. Since S 2 is a cipher system or a crypto system we know that d k 2 e k 2 x is equal to x for all x in p. Therefore, when d k 2 is applied over e e k 2, I know that I will get this argument back, and therefore, I will get d K 1 applied over e K 1 x, and then, since S 1 is a cipher system I know that d K 1 e K 1 x equal to x, and therefore, I will get x. Thus we have this result that d K 1 k 2 e K 1 k 2 x gives me x for all K 1 k 2 and all x. And therefore, we see that we indeed have a crypto system.

(Refer Slide Time: 08:32)



## Product cipher: Encryption and Decryption

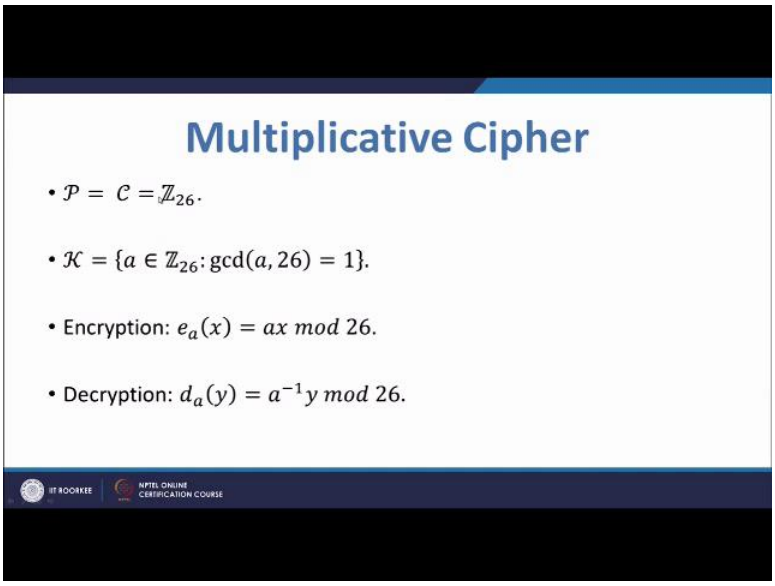
- For each $k = (k_1, k_2)$ we have encryption rule $e_k$ is

$$d_{(k_1,k_2)}\left(e_{(k_1,k_2)}(x)\right) = d_{k_1}\left(d_{k_2}\left(e_{k_2}\left(e_{k_1}(x)\right)\right)\right) = x.$$

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE

Now, an example of a product cipher is a combination of multiplicative cipher and a shift cipher, and we will see that if we take a multiplicative cipher and product it with a shift cipher, then we get our affine cipher. Now, let us look at this. A multiplicative cipher is something easier than affine cipher.

(Refer Slide Time: 09:21)



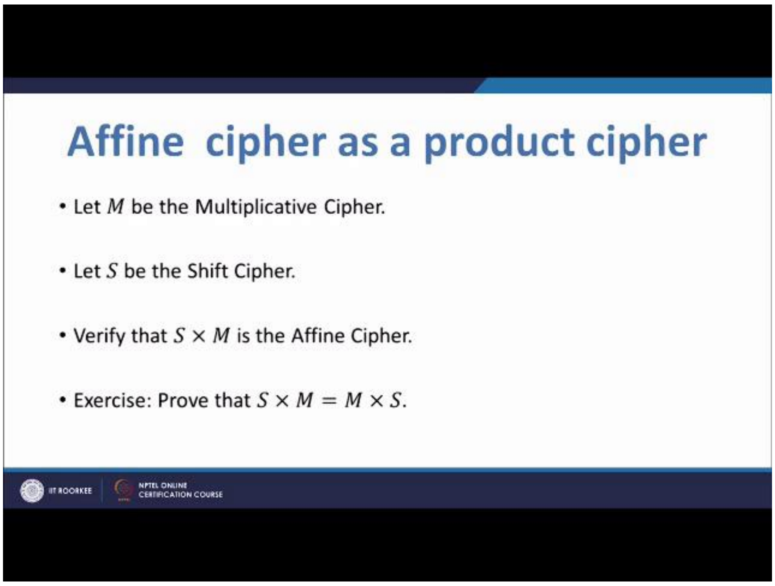
## Multiplicative Cipher

- $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$.
- $\mathcal{K} = \{a \in \mathbb{Z}_{26} : \gcd(a, 26) = 1\}$.
- Encryption: $e_a(x) = ax \; mod \; 26$.
- Decryption: $d_a(y) = a^{-1}y \; mod \; 26$.

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE

We have got P and C both equal to Z  sub 26 and the set of keys is all numbers co-prime to 26, as I have written over here, and the encryption function is just multiplication modulo 26 with that chosen key - in this case it is a - and decryption is done by taking a inverse and multiplying to the cipher text modulo 26. So, this is the multiplicative cipher.

(Refer Slide Time: 09:56)



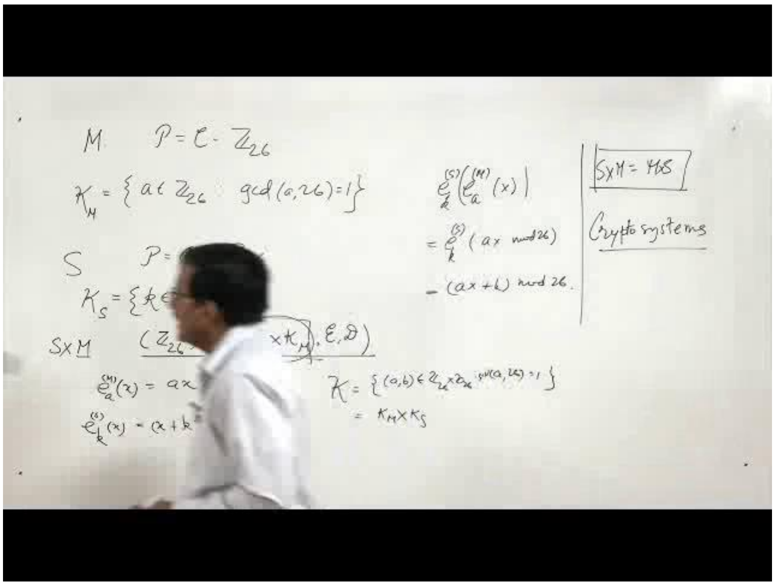
## Affine cipher as a product cipher

- Let $M$ be the Multiplicative Cipher.
- Let $S$ be the Shift Cipher.
- Verify that $S \times M$ is the Affine Cipher.
- Exercise: Prove that $S \times M = M \times S$.

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE

And now, we know that we have the usual shift cipher, and let M be the multiplicative cipher that we have seen just now, and usual shift shift cipher is just to add the key modulo 26 to the plain text between 0 to 25. If we take the product S cross M, then we will get affine cipher. I will quickly show this result and I will ask to you check this in more details after the class.

(Refer Slide Time: 10:50)



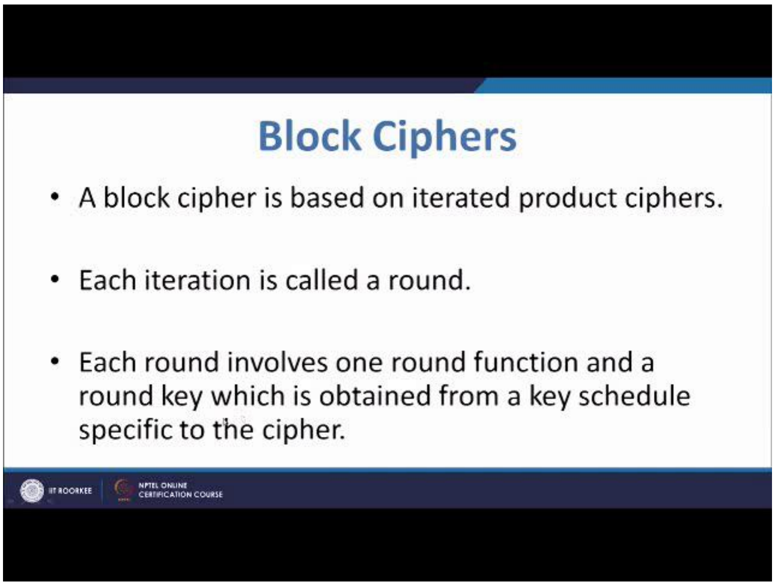
$M \quad \mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$
$K_M = \{a \in \mathbb{Z}_{26} \quad \gcd(a,26)=1\}$
$S \quad \mathcal{P} =$
$K_S = \{k \in$
$S \times M \quad (\mathbb{Z}_{26}$ $\times K_M), \mathcal{E}, \mathcal{D})$
$e_a^{(M)}(x) = ax$
$e_k^{(S)}(x) = (x + k$
$e_k^{(S)}(e_a^{(M)}(x))$
$= e_k^{(S)}(ax \bmod 26)$
$= (ax + k) \bmod 26.$
$K = \{(a,b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} \ \gcd(a,26)=1\}$
$= K_M \times K_S$
$\boxed{S \times M = M \times S}$
Cryptosystems

So, we take the multiplicative cipher M. P and C both are Z sub 26, and the key - let us call it k m - which is equal to all a belonging to Z sub 26, such that g c d of a and 26 is equal to one. Now, for the shift cipher P is equal to C is equal to Z 26 and k s which is a shift cipher the key set that is equal to k an element of Z 26 - any element of Z 26. So, if I am taking S cross M, then according to the definition P will be Z 26, C will be Z 26, that is alright, and the keys will be K s cross K m, and then, I will have the encryption function and decryption function.

So, now, let us see the encryption function. So, the encryption function of M, that is e a of x is equal to a x mod 26. Now the encryption function of; let us superscript it with m to indicate that this is the encryption function of the multiplicative cipher, then for the shift cipher the encryption function e s on x is going to give me x plus k with a key k is going to give me x plus k mod 26, and now, if I apply e a m on e k s like this over x, I will get e k s and here I will get a x mod 26, and this is going to be a x plus k mod 26, and this is exactly the encryption function of the shift cipher, sorry, the encryption function of the affine cipher. So, it is an encryption function of that.

Now, only thing that we will see is that here the key set is turned in a way, turned the other way around. In the case of affine cipher, we write the key space as this a comma b z 26 by z 26 where a 26 g c d is equal to 1 which is equal to K m cross K s, but in this case, I have got this case cross K m, but that is not going to be a big problem, because they are essentially the same set. Now there is question - are these 2 sets S cross M, M cross S, I am sorry, are these 2 crypto systems same? Now, this question I leave as an exercise. Please try to prove it whether these 2 crypto systems are same and the answer is yes, they are same, but it needs a proof.

(Refer Slide Time: 15:34)



## Block Ciphers

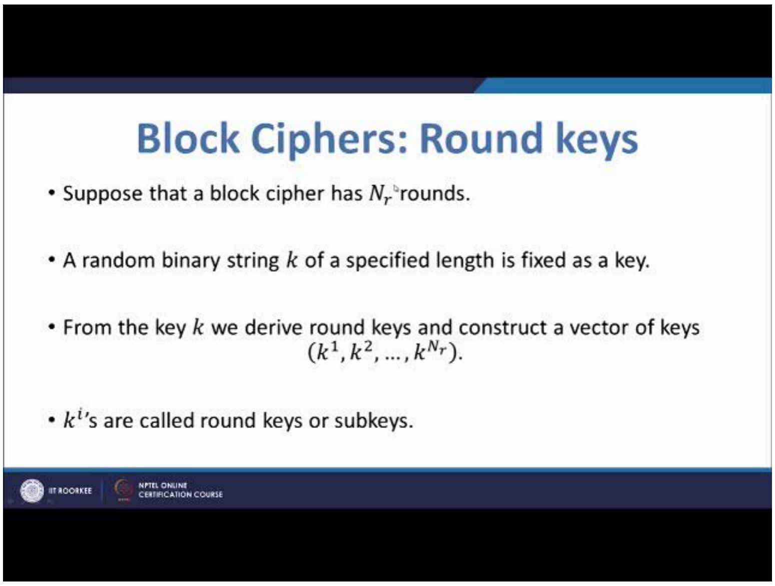
- A block cipher is based on iterated product ciphers.
- Each iteration is called a round.
- Each round involves one round function and a round key which is obtained from a key schedule specific to the cipher.

Now, we move onto block ciphers. The idea of block cipher is based on product ciphers and on something more specifically an iterated product cipher. So, we have a particular cipher, which we keep on applying over a plain text iteratively, several times, and in the process, get something which is called a block cipher.

It is called a block cipher because it is applied over a block of plain text or a block itself is called the plain text. More particularly, we will be taking a 64 bit segment and call it a block, and we will apply our transformation of the crypto system on that 64 bit block or it may be a 32 bit block or something like that, but it will be a block, and we will essentially apply a simple transformation over and over again, and then, that will result in security.

Now, there are some terms which are used in the context of block ciphers very frequently. So, each iteration is called a round. Each round involves 1 round function and a round key which is obtained from a key schedule specified to the cipher. Now, it is like this that we will have a key - a kind of master key - which is in the key space.

(Refer Slide Time: 17:30)



## Block Ciphers: Round keys

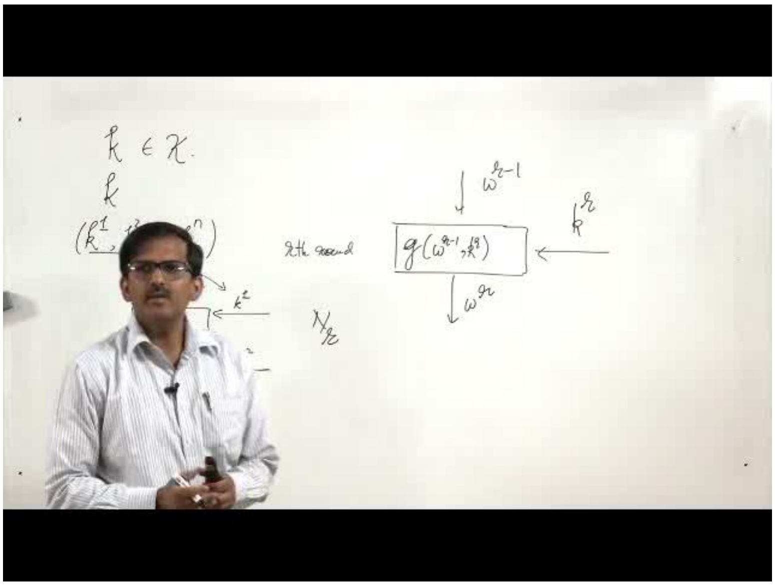
- Suppose that a block cipher has $N_r$ rounds.
- A random binary string $k$ of a specified length is fixed as a key.
- From the key $k$ we derive round keys and construct a vector of keys $(k^1, k^2, \ldots, k^{N_r})$.
- $k^i$'s are called round keys or subkeys.

And from this key, we will generate a key schedule of this type. May be some k and I have to see what we have written in the slides, but. So, it is called some kind of sub key or round key or something like that, and we will have a round, right now I am writing it as a as a box. So, we will have an input, let us say x, and it is a first round. So, first sub key or first round key will go here.

We will somehow combine these 2 things. So, I will have a function g x, g x k 1, a function g which combines x and K 1 and gives me something let us call it w 1. Then this w 1 will be passed to the next round; it is an iterative cipher. So, usually it is a same function. So, applied over w one with the round 2 key, so it is w 2; it will again pass on to the next round and so on. Ultimately, it will come to the last round and it will give me the final n ciphered, final n ciphered segment.

Now, this whole thing can be written by using some standard symbols. Now, the rounds we will be denoting by n r. So, n r is a number of rounds, and as I have said, the key is k, and these are the round keys. Then, as I have explained, g is the round function, and here I put a particular symbol, instead of x, I will write w to the power 0, which is an input to the first round.

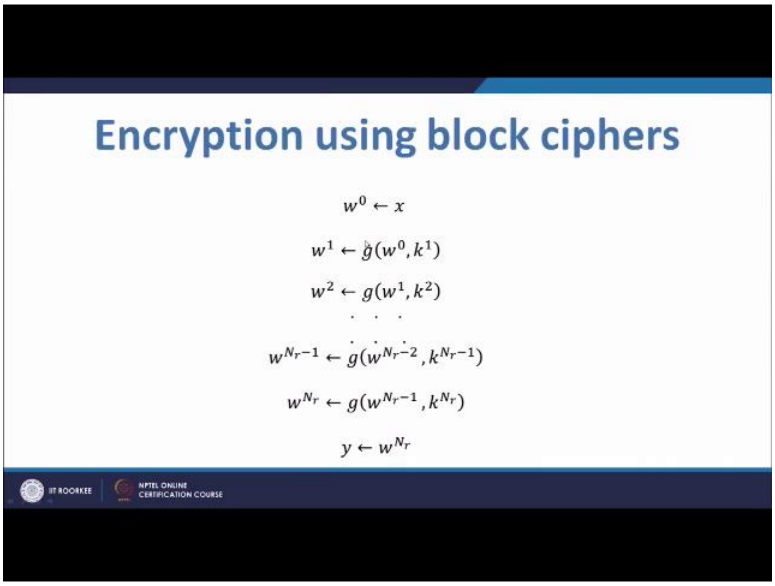
(Refer Slide Time: 20:07)



Then w to the power one that is the input to the second round and so on. So, this is what I have written over here which I have explained already on the blackboard. Now, what about the rth round? What is happening in the r th round? In the r th round; in the first round what is going in? w 0 is going in. In the second-round w one is going in. So, if I am considering the r th round, if I am considering the r th round, then w r minus 1 is going in.

In the first round what is the key? It is k 1. In the second round, it is k 2. So, in the r th round the key is k r. So, the round function g is going to process w r minus 1 comma k r, and it will produce w r, and that is what I have written here. So, this is the input to the next round. We will keep on doing this for inner rounds, and eventually, we will arrive at the final cipher text. These steps are formally represented in the slide over here.

(Refer Slide Time: 22:02)



## Encryption using block ciphers

$$w^0 \leftarrow x$$

$$w^1 \leftarrow g(w^0, k^1)$$

$$w^2 \leftarrow g(w^1, k^2)$$

$$\vdots$$

$$w^{N_r-1} \leftarrow g(w^{N_r-2}, k^{N_r-1})$$

$$w^{N_r} \leftarrow g(w^{N_r-1}, k^{N_r})$$

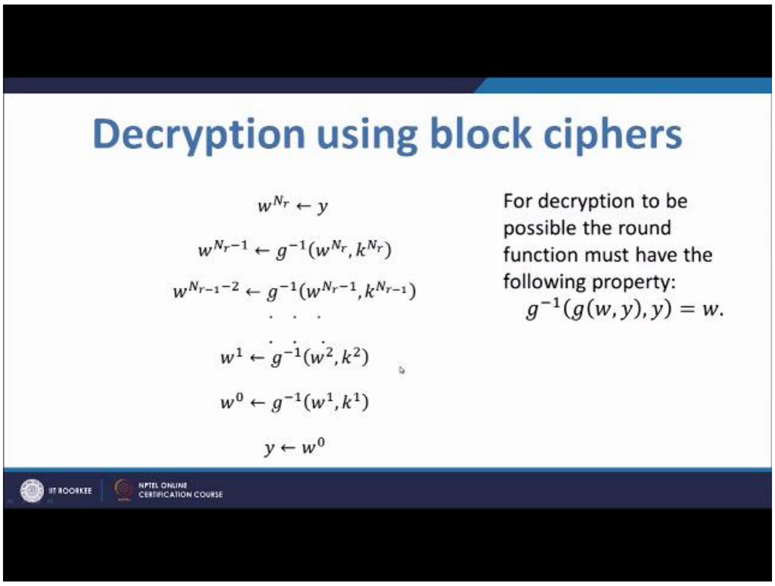
$$y \leftarrow w^{N_r}$$

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE

That is x first w 0, then how w 1 is constructed, how w 2 is constructed so on, and ultimately how w n r which is the cipher text is constructed. Now, this n r has to be decided by the designer, that how many rounds they are going to apply the function g successively over a plain text and with a sequence of round keys; that has to be done by the designer.

For decryption, we have to take care of one point when we are defining the round function, which is this 1 - which the round function g should be such that g inverse g of w y comma y has to give me w. Suppose y is the key. So, if y is the key, suppose I have computed the image of w with y, and suppose I want to compute back w, I can do that by applying g inverse over this thing and I will get w. So, this is a very important condition that a round function has to obey in order that a block cipher is practically useful or meaningful.

(Refer Slide Time: 23:43)



## Decryption using block ciphers

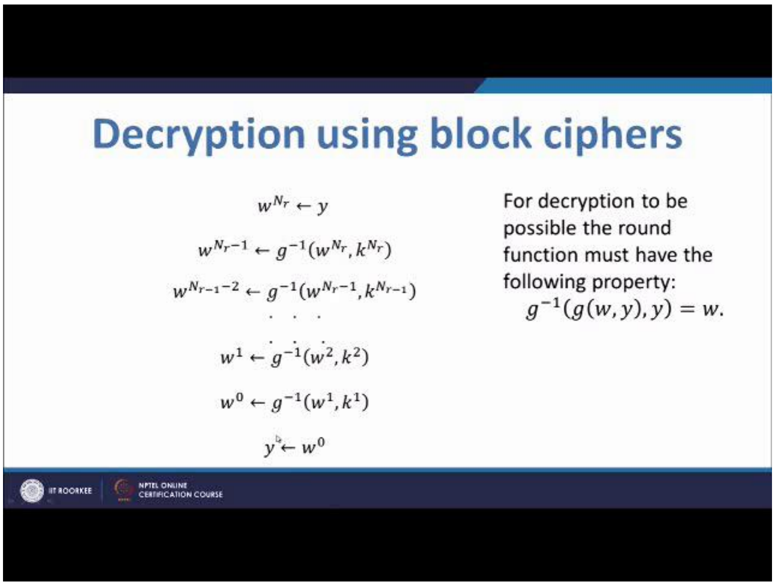
$$w^{N_r} \leftarrow y$$
$$w^{N_r-1} \leftarrow g^{-1}(w^{N_r}, k^{N_r})$$
$$w^{N_{r-1}-2} \leftarrow g^{-1}(w^{N_r-1}, k^{N_{r-1}})$$
$$\cdot \quad \cdot \quad \cdot$$
$$w^1 \leftarrow g^{-1}(w^2, k^2)$$
$$w^0 \leftarrow g^{-1}(w^1, k^1)$$
$$y \leftarrow w^0$$

For decryption to be possible the round function must have the following property: $g^{-1}(g(w, y), y) = w.$

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE

So, if you look at this sequence of steps, you will see that this is how we decrypt. We had w n r, and we applied this property over and over again, and from that we will have w n r minus one, and in the next step we will have w n r minus 2, and eventually we will have w 0, which is equal to not y, but it is x. So, here is a correction in the slide, in the last line w 0 goes to x and not w 0 goes to y - please note that. I will update the slide when we upload all the slides.

(Refer Slide Time: 24:33)



## Decryption using block ciphers

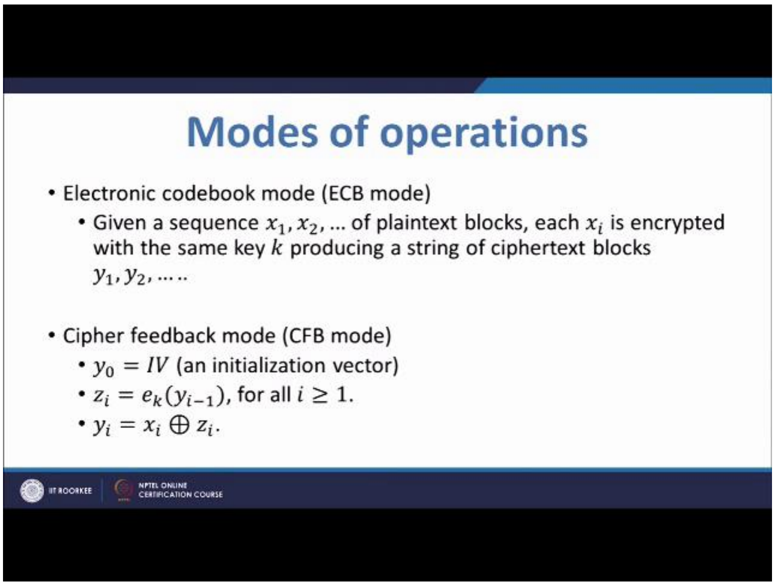
$$w^{N_r} \leftarrow y$$
$$w^{N_r-1} \leftarrow g^{-1}(w^{N_r}, k^{N_r})$$
$$w^{N_{r-1}-2} \leftarrow g^{-1}(w^{N_r-1}, k^{N_{r-1}})$$
$$\cdot \quad \cdot \quad \cdot$$
$$w^1 \leftarrow g^{-1}(w^2, k^2)$$
$$w^0 \leftarrow g^{-1}(w^1, k^1)$$
$$y \leftarrow w^0$$

For decryption to be possible the round function must have the following property: $g^{-1}(g(w, y), y) = w.$

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE

Coming to block ciphers in general, when block ciphers are used for cryptography, then they are used according to certain modes of operations. We will discuss four important modes of operations right now. The first one is called electronic code book mode or in short ECB mode; this is a first one; this is the easiest possible mode, where the plain text are segmented into blocks, and these blocks are x 1, x 2, and so on.

(Refer Slide Time: 25:14)

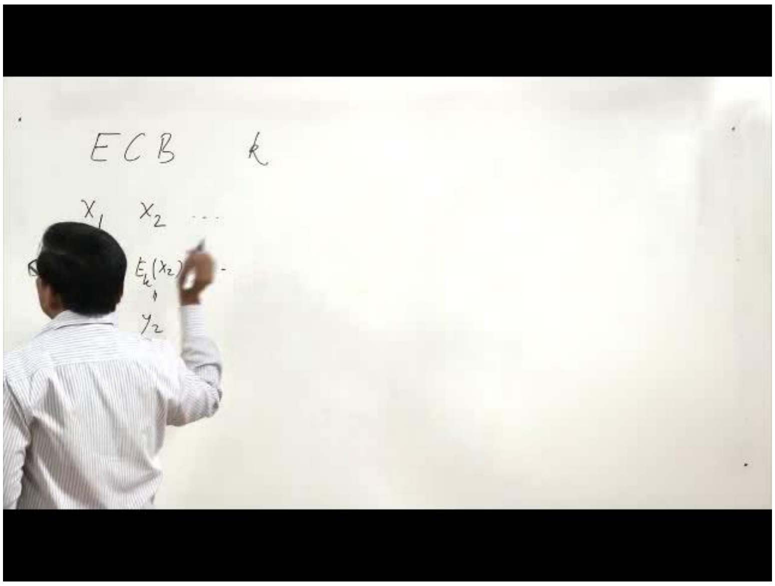


## Modes of operations

- Electronic codebook mode (ECB mode)
  - Given a sequence $x_1, x_2, \ldots$ of plaintext blocks, each $x_i$ is encrypted with the same key $k$ producing a string of ciphertext blocks $y_1, y_2, \ldots\ldots$
- Cipher feedback mode (CFB mode)
  - $y_0 = IV$ (an initialization vector)
  - $z_i = e_k(y_{i-1})$, for all $i \geq 1$.
  - $y_i = x_i \oplus z_i$.

So, we have x 1, x 2, and so on, and we choose a key which is fixed, and each time I am using the encryption function with the same key.

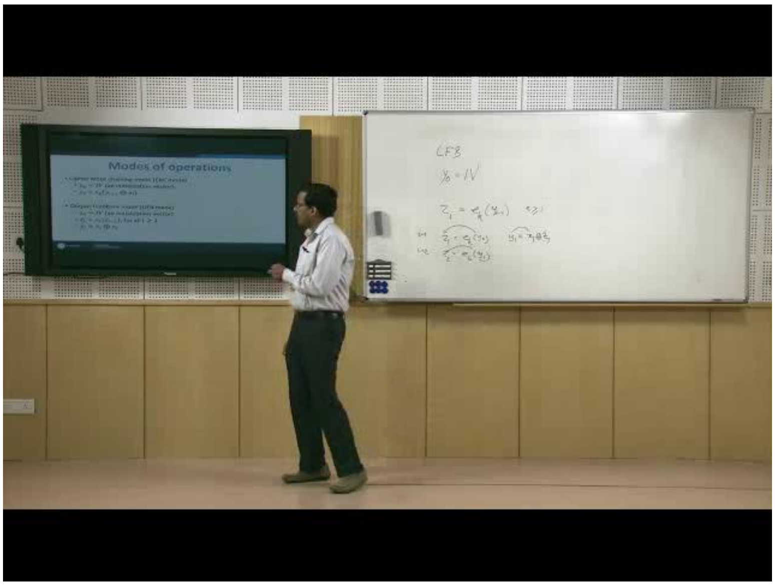
(Refer Slide Time: 25:49)



Let us say, this time on x 1, so I am getting y 1; this time on x 2, so I am getting y 2, and so on. This is called the ECB mode.

Next, we have the cipher feedback mode; in short CFB mode. In CFB mode, we introduce something called an initialization vector, in short i 0 and say y 0 is equal to that initialization vector called i v and we say that y 0 is equal to i v and we compute z, z i which is e k y i minus 1, where i starts from 1 and increases.

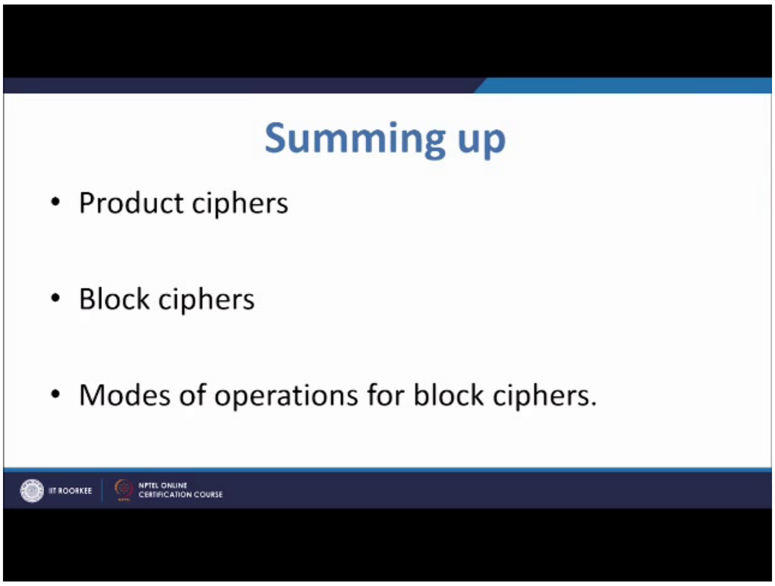
(Refer Slide Time: 26:23)


Modes of operations
CFB
$y_0 = IV$
$z_i = e_k(y_{i-1})$ $i \geq 1$
$i=1$ $z_1 = e_k(y_0)$ $y_1 = x_1 \oplus z_1$
$i=2$ $z_2 = e_k(y_1)$

So, if i equal to 0, we get let us say, not i equal to 0, but if equal to 1, then we get z 1 is equal to a k y 0; y 0 is a public vector, which is the i v - initialization vector. So, I get this, and then, z 2 for i equal to 2 z 2 is e k of y 1, and I have to calculate y one, once I know z 1. So, y 1 is x 1 plus z 1. So, here from e 0 I will know z 1, once I know z 1 I sum, that is I take the x or bit wise x or with x 1 and z 1. and I get y 1; once I know y 1, I will get z 2 and so on, I will keep on encrypting. So, this is the cipher text feedback mode.

And 2 last modes are like this - that is cipher block chaining mode, where I start with an initialization vector, and then, add that initialization vector bitwise modular to the message, and then apply the encryption and construct y i. Again, this y i will be pushed into this and so on. And output feedback mode is where we start from initialization vector, and then keep on generating these z (s) of i (s), by applying successively over the results, and then, add the z i (s) to x i (s) and generate y i. So, these are the four important modes of operations of block ciphers.

(Refer Slide Time: 29:06)



## Summing up

- Product ciphers
- Block ciphers
- Modes of operations for block ciphers.

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE

So, summing up - we started with product ciphers in this lecture; after that we studied the general principles and designs of block ciphers, and then at the end we have talked about the modes of operations of block ciphers. This is all for this lecture. We will study block ciphers in more details in the subsequent lectures.

Thank you.