Introduction to Cryptology

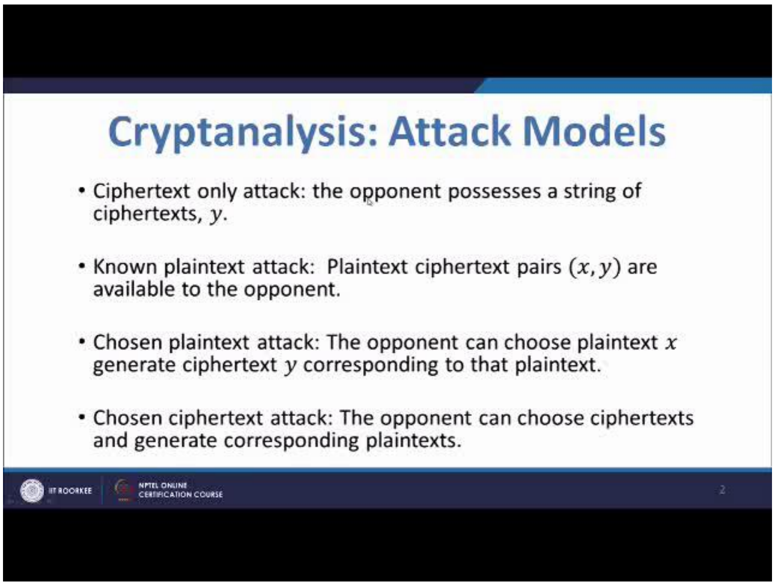
Dr. Sugata Gangopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee

Lecture – 09
Cryptanalysis and its variants, linear attack

Welcome to week 2, Lecture - 4. Now in this lecture, we will be talking about Cryptanalysis and give a very simplified version of Cryptanalysis of a block cipher, in fact rather to say a cryptanalysis of a single s box, but before that we have to understand what is meant by cryptanalysis.

So, roughly speaking Cryptanalysis is the subject which is the interest of an attacker because he would like to intercept messages sent by Alice to Bob and without knowing the key he would like to know the key or at least the content of the message. Now here there are several attack modules, depending on the power of the opponent or the adversary.

(Refer Slide Time: 01:29)

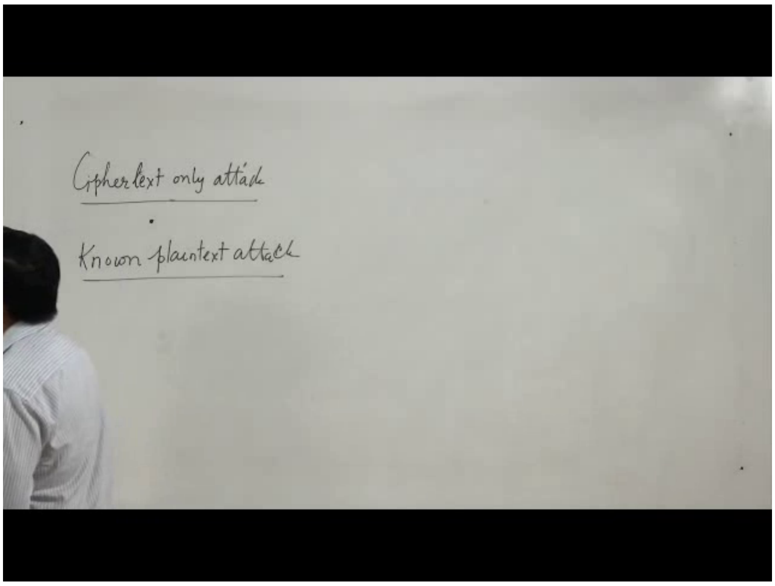


Cryptanalysis: Attack Models

- Ciphertext only attack: the opponent possesses a string of ciphertexts, y.
- Known plaintext attack: Plaintext ciphertext pairs (x, y) are available to the opponent.
- Chosen plaintext attack: The opponent can choose plaintext x generate ciphertext y corresponding to that plaintext.
- Chosen ciphertext attack: The opponent can choose ciphertexts and generate corresponding plaintexts.

So, the first module is ciphertext only attack.

(Refer Slide Time: 01:40)



So, in this module the attacker has only access to ciphertext and nothing else; ciphertext only attack. So, of course, we assume that the attacker knows the cryptosystem being used, he knows the circuits, he knows the algorithms and everything, but he does not know the particular key that is used by Alice to encrypt the messages that he is encrypting.

So, the first module as I said the ciphertext only attack module, here we assume that the attacker knows only ciphertext and nothing else, so is a ciphertext only attack. Now of course, one will say that what is do you think the attacker should be knowing, now the answer is that attacker in practise can know something much more and that gives raise to known plaintext attack, where we assume that the attacker has several pairs of plaintext and ciphertext enciphered by using a particular key. But the attacker does not know the key, so we come to known plaintext attack.

So, here as I said the plaintext ciphertext pairs x y's are available to the opponent. Well then one can ask that how is it possible? The point is that in practise it is indeed possible because it is quite possible that; parts of the messages are something very standardise. So, everybody knows that if; let us say Alice communicates to Bob, everybody may know that some parts of the messages that are being sent; it might be a form where many things that are written are common to everybody's form. So one can know that, one can easily presume that I mean suppose Alice is talking to Bob, then one can assume that okay we know that she must be sending this form, so this part of the message is going to be this part of the form which we already know.

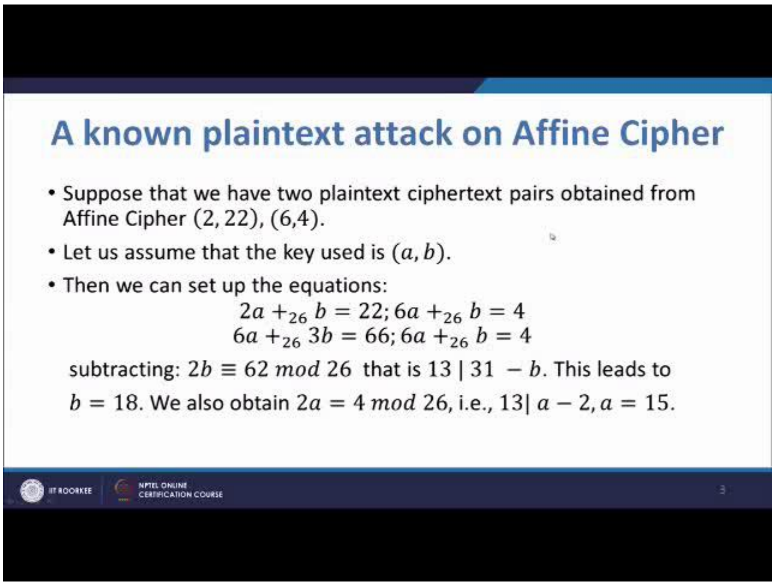
So, I know the plaintext and ciphertext pair, but of course I do not know the key so this is this model plaintext ciphertext known plaintext attack and we would like definitely that our cryptosystem should be resistant to known plaintext attack. Now, there are certain other attacks which are also feasible and we would like to have our cryptosystem resistant to those attacks, so one is chosen plaintext attack where we have the attacker with the ability to choose the plaintext and that is also something very strange.

One would say that how is it possible, but it is not at all impossible because for example, Alice might be using a device to send message, but the device for some reason may be available to the attacker for certain amount of time so that the attacker or (Refer Time: 05:52) can put chosen plaintext inside the device and check the ciphertext coming out of it and generate several pairs; this is possible.

Now, the last one is chosen ciphertext attack where the opponent can choose ciphertexts and generate corresponding plaintexts. Now this is a scenario where the opponent has a access to description function. In our discussion we will be particularly considering known plaintext attack and chosen ciphertext attack; sorry ciphertext only attack is difficult, so we usually do not consider that; although it is possible in several instances to launch ciphertext only attack, so we concentrate on known plaintext attack.

Now, we go back to affine ciphers, which is a classical cipher and we show that affine ciphers are particularly vulnerable to known plaintext attack. Now affine ciphers are also vulnerable to ciphertext only attack because when we encrypt plaintext to ciphertext by using an affine cipher then the letters; each individual letter is mapped to a unique letter and therefore, the statistical distribution is not changed therefore, we can see the statistical distribution and very quickly guess which letter is changed to which letter and then we can kind of since we know that affine cipher is used so we might be able to know the key with less a time than exhaustive such, but the known plaintext attack on affine cipher is even most serious that let us look at this example.

(Refer Slide Time: 08:06)



A known plaintext attack on Affine Cipher

- Suppose that we have two plaintext ciphertext pairs obtained from Affine Cipher (2, 22), (6,4).
- Let us assume that the key used is (a, b).
- Then we can set up the equations:

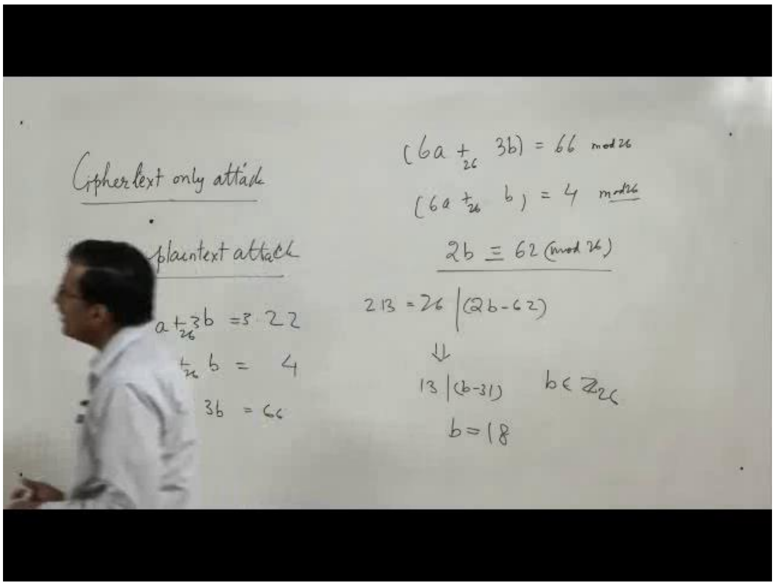
$$2a +_{26} b = 22; 6a +_{26} b = 4$$
$$6a +_{26} 3b = 66; 6a +_{26} b = 4$$

subtracting: $2b \equiv 62 \; mod \; 26$ that is $13 \mid 31 \; - b$. This leads to $b = 18$. We also obtain $2a = 4 \; mod \; 26$, i.e., $13 | \; a - 2, a = 15$.

IIT ROORKEE NPTEL ONLINE CERTIFICATION COURSE 3

Suppose that we have two plaintext ciphertext pairs obtained from affine cipher. So, suppose this pair is 2, 22 and 6, 4 and now we ask a question; can we find out the keys and suppose the key is assumed to be; a b we do not know the values of a b, but since we know that the plaintext 2 goes to 22, then we can write 2 a plus mod 26; b is equal to 22 and 6 a; modulo 26, p is equal to 4, so we set up an equation.

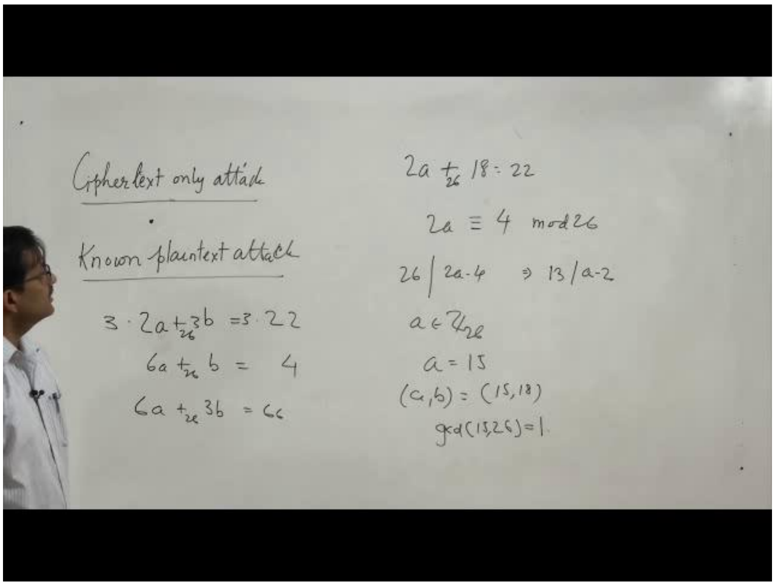
(Refer Slide Time: 08:52)



Now, let us look at this equation 2 a; modulo 26; b equal to 22 and 6 a addition modulo 26; b equal to 4. Now what I can do; is to multiply the first equation by 3 because after all 3 is not 0 mod 26, I can do that and still this relationship will hold and we obtain an equation 6 a plus 26; 3 b and equal to 66. So, we write these equations as fresh, I have got 6 a addition mod 26; 3 b is equal to 66 and of course, I have to put mod 26 here; you do not do that now and here 6 a plus mod 26; times b equal to 4, I could put mod 26 here as well and I do not have to do that because it is 4 already and I am allowed to cancel 6 a because if I subtract this quantity from the top quantity, then also this relation; equivalence relation going to hold.

So, therefore, I will get two times b equal to 62; mod 26 and 62; mod 26 is going to be; let see, I do not do anything right now. So, what I do is that; okay I see it is 62 mod 26, the question is what does it mean, is essentially is congruent to 62 mod 26; therefore, I know is 26 divides 2 b minus 62, but 26 is 2 into 13. So, from this we know that 13 divides b minus 31 and by our definition b is an element of z sub 26 and the only such b possible is b equal to 18.

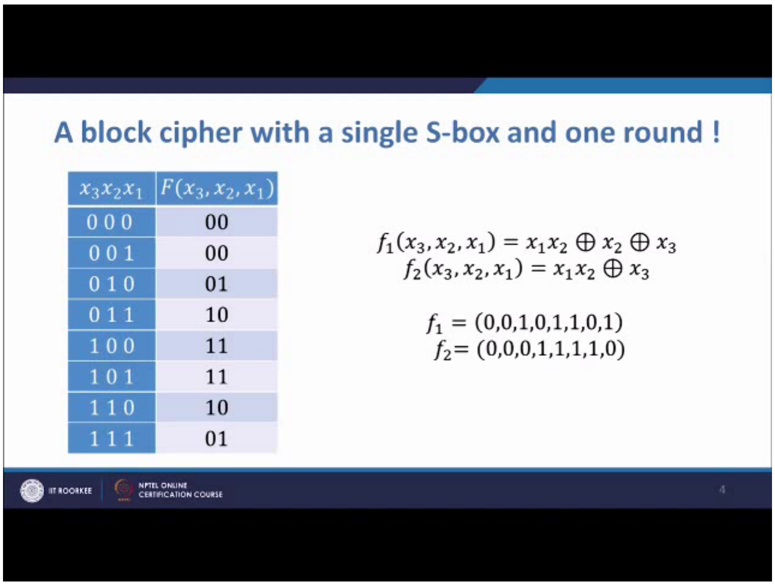
(Refer Slide Time: 12:31)



So, I have retrieved b that I see over here somewhere b is 18 and when I put the value of b in the first equation, so that I get 2 a plus modulo 26 times 18; 2 a plus mod 26 times 18 equal to 22 and which gives me 2 a plus mod 26 no sorry, so it basically gives me this that 2 a is congruent to 4 mod 26. So, 26 divides 2 a minus 4 which implies 13 divides a minus 2 and I know that a is an element of z sub 26 and only the possible element is 15 and we see that we have retrieved a and more so that a, b is 15, 18 and well gcd of 15, 26 is 1; that means, 15, 18 is indeed a valid key of affine cipher. So, we have obtained a key by looking at just two plaintext ciphertext pair.

So, I am giving you a very, very simplified example of a known plaintext attack on affine ciphers. Of course affine cipher is weak, of course affine cipher is a classical cipher, but it gives a very good example of this known plaintext attack. We see that it is weaker if we assume that the opponent has access to sum in fact just two plaintext ciphertext pairs.

(Refer Slide Time: 14:09)



A block cipher with a single S-box and one round !

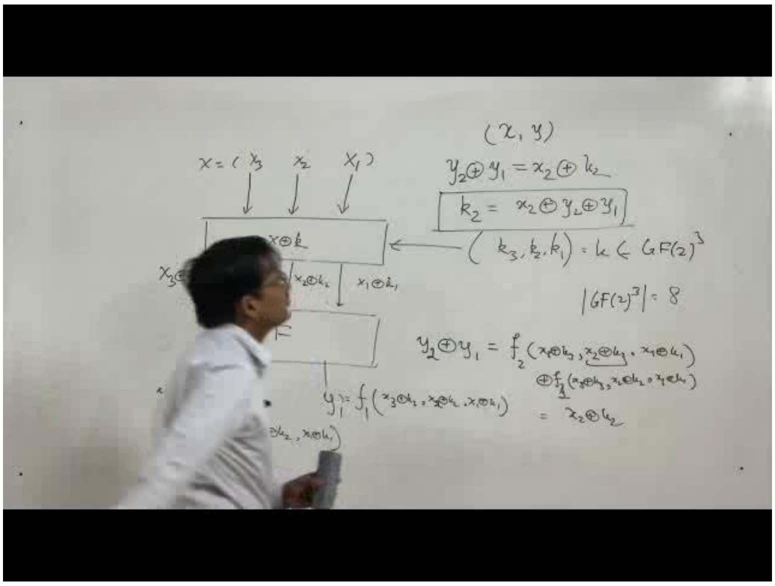
$x_3x_2x_1$	$F(x_3,x_2,x_1)$
0 0 0	00
0 0 1	00
0 1 0	01
0 1 1	10
1 0 0	11
1 0 1	11
1 1 0	10
1 1 1	01

$$f_1(x_3,x_2,x_1) = x_1x_2 \oplus x_2 \oplus x_3$$
$$f_2(x_3,x_2,x_1) = x_1x_2 \oplus x_3$$

$$f_1 = (0,0,1,0,1,1,0,1)$$
$$f_2 = (0,0,0,1,1,1,1,0)$$

Now, we come to something which is a very, very simplified block cipher it is; well I think the most; the simplest possible block cipher and of course, it is not strong, but this cipher gives an understanding of what we mean by linear approximation attack. So let us look at the description of this block cipher, now I will be eventually using an s box which is given by this. So, if you check that f 1; by f 1 I mean the left most column this is f 1 and by f two I mean the second one, so I have got the coordinate functions f 1; f 2 and the function itself; the s box is capital F and I can write f 1, f 2 as arrays like these. Now this is my block cipher which is easy; I will draw a figure on the board as well

(Refer Slide Time: 15:31)


(x, y)
$x = (x_3 \;\; x_2 \;\; x_1)$
$x \oplus k$
$x_2 \oplus k_2$
$x_1 \oplus k_1$
$y_2 \oplus y_1 = x_2 \oplus k_2$
$\boxed{k_2 = x_2 \oplus y_2 \oplus y_1}$
$(k_3, k_2, k_1) = k \in GF(2)^3$
$|GF(2)^3| = 8$
$y_2 \oplus y_1 = f_2(x_3 \oplus k_3, x_2 \oplus k_2, x_1 \oplus k_1) \oplus f_1(x_3 \oplus k_3, x_2 \oplus k_2, x_1 \oplus k_1) = x_2 \oplus k_2$
$y_1 = f_1(x_3 \oplus k_3, x_2 \oplus k_2, x_1 \oplus k_1)$

I have got three bit input and I have got the key mixing layer, I have got three bit keys, so this is k 1, k 2 and k 3 and inputs are x 1, x 2 and x 3; that is what is happening here and if you look here I am doing the key mixing by bit by XOR. So, after the key is mixed; this is this x, XOR k because my input is x and my k is key, it is a single round, single s box, no permutation layer s box. So, they are going like this.

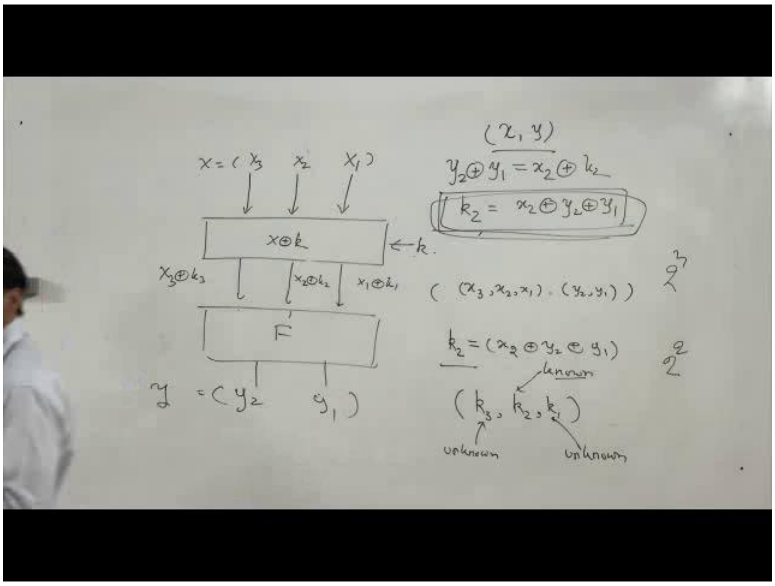
So, whatever is coming out is something like this x 3, XOR, k 3, then x 2; x or k 2 and x 1; XOR; k 1 it is coming like this and then I am passing this through the s box denoted by f and whose function is given over there and therefore, I am getting this; no I am getting only two output, so just two outputs over here and this output will be denoted by y 1 and this output will be denoted by y 2, I am not written over here, but please look at the blackboard and y 1 is essentially a 1 and then I will put those values. So, this is x 3; XOR; k 3, x 3; XOR; k 2; x 2; XOR; k 2 and x 1; XOR; k 1, this is f 1 and y 2 is f 2 which is x f 2 of x 3; XOR; k 3; x 2; XOR; k 2; x 1; XOR; k 1 so i have got this.

Now, suppose I have got a plaintext ciphertext pair so; that means, that I have got this pair x, y then what I can do is that; I can vary this k over all possible values and there are 8 such values. So, because k is an element of gf 2 to the power 3 and there are 8 bit patterns in gf 2 to the power 3, I can vary this value and run this and see where the input is matching with output and that is my candidate of key, this is called exhaustive search and this is a very important technique, but our question is that based on the functional properties, can I reduce to something less than exhaustive search in this case.

Now here; we see the weakness of this function is that, if I add f 1 and f 2 then it becomes x 2. So, f 1 and f 2 are essentially quadratic functions which are quite nice and all that, but if I add them that is; it is a component function, that component function is just x 2, so if I add y 1 and y 2. So, let us see here if I add y 1, let me write over here y 2 and y 1 then; that means, I am adding f 2; x 3 plus k 3; x 2 plus k 2 and x 1 plus k 1; XOR; f 1; x 3 plus k 3; x 2 plus k 2; x 1 plus k 1 and I know that it is only x 2, but in this case x 2 is x 2; x 2; XOR k 2, so this is what I am getting.

Now that means that; I have got an equation involving k 2 let me write it over here. I have got an equation like this; k 2 equal to x 2 plus wait a moment. So, I have got this one; that means, y 2 plus y 1 equal to x 2 plus k 2, of course this plus is XOR and therefore, k 2 is equal to x 2 plus y 2 plus y 1 and this is interesting. Now why this is interesting? Now let me remove the calculations and just look at this particular expression. So, I remove this portion of the blackboard, I remove this; I do not want all these things; just the rudiments, here k is coming in and here this is y 2 and y 1 and this is the input, this is the output and I have got the access of a pair of input output variables

(Refer Slide Time: 22:17)

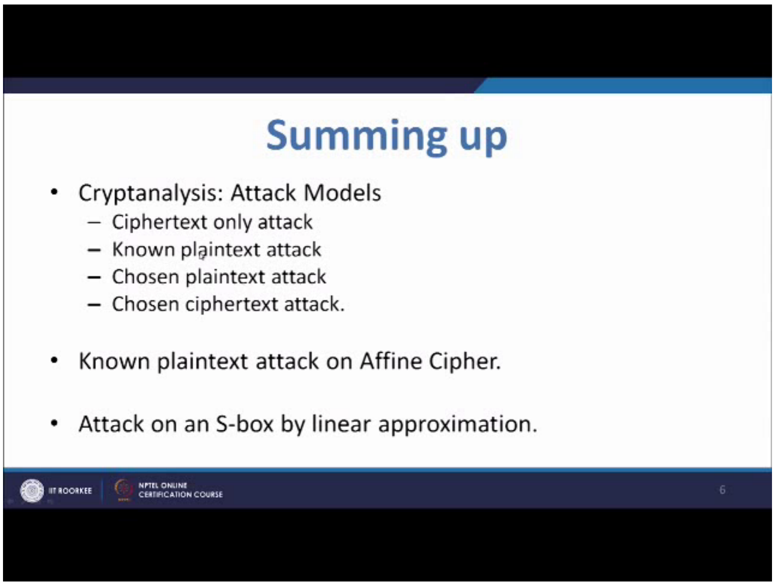

$x = (x_3 \quad x_2 \quad x_1)$
$x \oplus k$
$\leftarrow k$
$x_3 \oplus k_3$
$x_2 \oplus k_2$
$x_1 \oplus k_1$
F
$y = (y_2 \quad y_1)$
(x, y)
$y_2 \oplus y_1 = x_2 \oplus k_2$
$k_2 = x_2 \oplus y_2 \oplus y_1$
$((x_3, x_2, x_1), (y_2, y_1))$
2^3
$k_2 = (x_2 \oplus y_2 \oplus y_1)$
2^2
known
(k_3, k_2, k_1)
unknown
unknown

So, I know these; knowing that means, that I know x 3, x 2, x 1 and y 2, y 1; I know particular values of this, once I know this; I know that because of the properties of the function k 2 is equal to x 2 plus y 2 plus y 1, I can always evaluate k 2. Once I evaluate k 2; that means that when I am searching over all the keys, one coordinate of the key is known to me and that is k 2. So, k 3; k 2; k 1 this is the key string this is known, therefore, I have to search for only these two unknown keys, unknown key bits and therefore, my search reduces from 2 to the power 3 to 2 square because I have got just four possibilities.

So, if I do that; I will be definitely able to get the value of the keys. So, this is what happens if a function; if a component function of an s box becomes linear. Now one may say that OK fine it may not become linear then what happens then, now in that case; even if it is close to linear that is its hamming distance is close to linear, then for; if we take many pairs, probability that an equation like this; a linear equation will be satisfied between some of the key bits which are unknown to the bits of the plaintext and ciphertext becomes high and therefore, we can mount an attack which will not be successful always, but a some kind of statistical attack and be very be quite positive, quite optimistic of getting a solution.

So, this is a very, very simplified example of linear attack on a block cipher, rather I should say it is an example of a; simplified example of a linear attack on an s box.

(Refer Slide Time: 25:26)



Summing up

- Cryptanalysis: Attack Models
  - Ciphertext only attack
  - Known plaintext attack
  - Chosen plaintext attack
  - Chosen ciphertext attack.
- Known plaintext attack on Affine Cipher.
- Attack on an S-box by linear approximation.

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
6

Summing up, in this lecture we started off with the definition of Cryptanalysis and the attack models. We saw that there are several attack models and we have to know which attack model we are using and then we checked affine cipher and saw that it is not resistant to known plaintext attack and finally, we got a somewhat simplified idea of a linear Cryptanalysis, that is all for today.

Thank you.