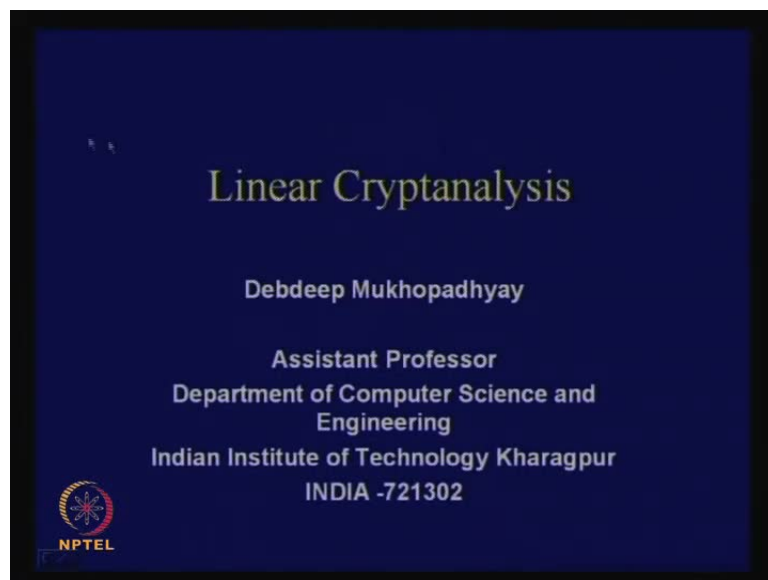


**Transcriber's Name: Arun Raj V**  
**Cryptography and Network Security**  
**Prof. D. Mukhopadhyay**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

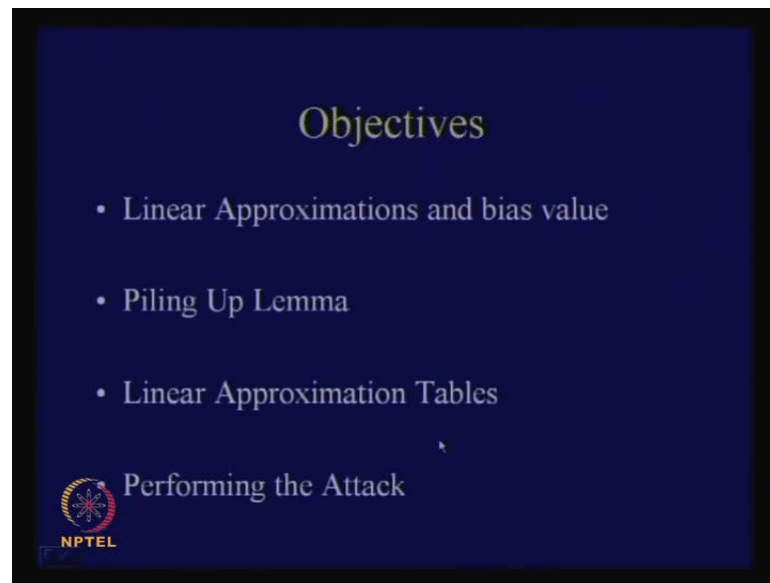
**Module No. # 01**  
**Lecture No. # 14**  
**Linear Cryptanalysis**

(Refer Slide Time: 00:23)



Ok. So, in today's class, we shall discuss about a particular cryptanalytic technique which is known as linear cryptanalysis. So, till now, we have seen the construction of block ciphers, but this is the first time when we will see how standard block ciphers, that is, modern block ciphers like *DES* or *AES* or any ciphers of these families or cryptanalyst.

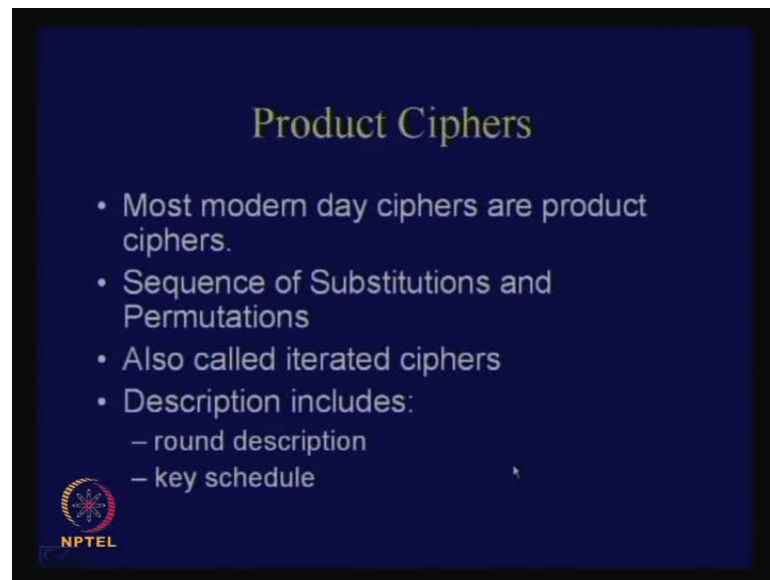
(Refer Slide Time: 00:43)



So, linear cryptanalyst is a very powerful tool, and all the modern constructions of block ciphers are based upon this analysis, that is, first this analysis is done; the robustness is measured, and based upon that, we will decide at whether the given block cipher is able to protect against linear cryptanalysis or not, ok?

So, the objectives of today's class are as follows: first we will try to understand what is meant by linear approximations of non-linear Boolean functions, and then discuss about something which is called bias. So, what is the definition of bias that we will see, and then we will study a lemma which is known as piling up lemma to **[und/understand]** and that this lemma and these theory essentially was given by a paper by Matsui. So, all those who are interested can read that paper as well, ok?

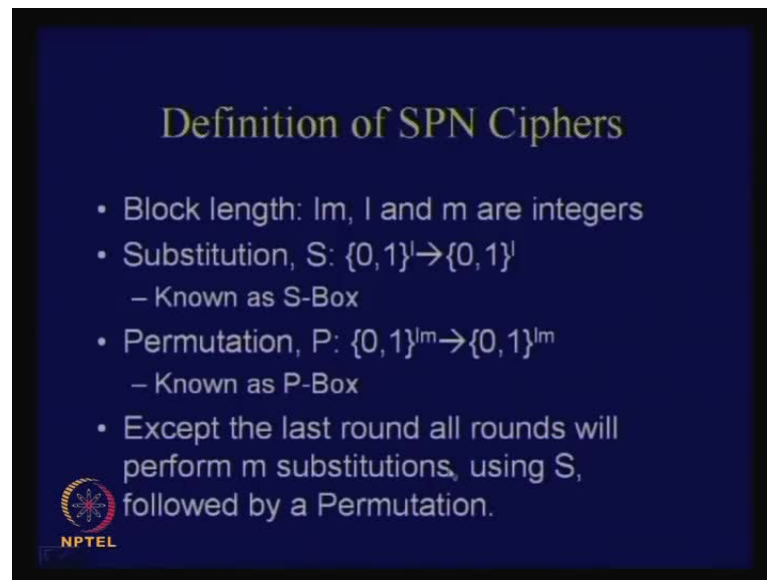
(Refer Slide Time: 01:48)



So, the, after that, we will discuss about something which is called linear approximation tables, which is constructed for the s boxes, and follow that up with the actual performance of the attack, that is, how do we carry out the attack given all these previous things, like given linear approximation table, how do we really perform a linear attack, a linear cryptanalytic attack. So, this is a brief revision. So, we were discussing about product ciphers like d e s a e s as we discussed. Essentially all of them are product ciphers. So, most modern day ciphers are product ciphers, and essentially they [co/could] encompass a sequence of substitution and permutation, ok?


So, these are also called iterated ciphers, and the description generally includes a round description, that is, that is the description of round. So, the rounds are applied one after the other, and that is also a description of the key scheduling algorithm, that is, how the round keys are derived from the given input key, ok?

(Refer Slide Time: 02:25)



**Definition of SPN Ciphers**

- Block length:  $l$ ,  $l$  and  $m$  are integers
- Substitution,  $S: \{0,1\}^l \rightarrow \{0,1\}^l$ 
  - Known as S-Box
- Permutation,  $P: \{0,1\}^m \rightarrow \{0,1\}^m$ 
  - Known as P-Box
- Except the last round all rounds will perform  $m$  substitutions, using  $S$ , followed by a Permutation.

 NPTEL

So, consider that, so, essentially we can sort of define this as follows: that suppose the round function is denoted by the variable  $g$ . So,  $g$  takes two inputs. So, what are the two inputs? It takes the round key corresponding round key  $k_r$  and it also takes the corresponding current state which is represented by  $w_{r-1}$ .

So, therefore, this current state is the output of the previous round, and then  $g$  essentially acts upon these two variables, that is,  $w_{r-1}$  - that is the previous state, that is, the output of the previous round, and the corresponding round key and produces the output of these particular round, that is, the  $r$ th round, ok?

So,  $w_r$  is equal to  $g$  which is a function, which is acting upon  $w_{r-1}$  and  $k_r$ . So, and, and, so, your input essentially that is the plain text can be denoted by  $w_0$ , right? Because  $w_0$  is the input to your 0th round. Where cipher text is  $w_{nr}$ , where  $nr$  denotes the number of rounds of the cipher. Therefore, after  $nr$  number of operations of the rounds, the output that you get is the corresponding cipher text. The decryption therefore, you can easily understand that since  $g$  has to, I mean since we have to decrypt the corresponding cipher, therefore,  $g$  has to be an invertible mapping, ok?

So, therefore, in order to do the decryption, we require the transformation  $g^{-1}$ ; that is the inverse of  $g$ . So, these are very simple sort of you can say a mathematical way of stating or what we have studied as product ciphers, ok?

So, the definition of substitution permutation networks, I mean in order to understand how linear cryptanalysis works, you will consider a very toy example; you consider a simple toy example; a simple construction of an s p n cipher. See imagine that the s p n cipher that we have you, this is [mo/more] generalized view of seeing an s p n cipher. So, when s p n cipher will generally will act upon a block length, right? So, therefore, it is a block cipher. Since it is a block cipher, so, imagine that suppose that the length of the block is equal to  $l$  into  $m$ , where both  $l$  and  $m$  are integers. So, and your substitution block or a substitution [bo/box] box acts upon  $l$  bit elements.

So, therefore, it acts upon say zero one and I denote that by zero one  $l$ ; which means the bit string of length  $l$ , and it produces also another bit string of length  $l$ . So, that means what is the total size; I mean this s box acts upon  $l$  bit value and produces an  $l$  bit value. Therefore, this is a, imagine this is a bijective mapping, ok?

So therefore, this is, [comm/commonly] we have seen that this is commonly known as the s box. The p box or the permutation box is a permutation of  $l m$  bits therefore, it is a permutation of  $l m$  number of bits. So, therefore, it acts upon the entire block of the data and it permutes the bit; transposes the bit. So, therefore, this description should be quite clear, and, I, I think we can figure out how many s boxes do we have. How many s boxes do we have?

(( ))

$m$  number of s boxes. Therefore, this is a repetition of the same s box and there are  $m$  s boxes. So, except the last round, all the rounds will perform  $m$  substitutions using s followed by a permutation. Why I have read, why I have return except the last round because the last round does not have the permutation step. So...

(( ))

(Refer Slide Time: 05:57)

### Algorithm

- Input,  $x: \{0,1\}^{lm}$ ,  $K_0: \{0,1\}^{lm}$
- Output,  $y: \{0,1\}^{lm}$
- Key-schedule: generates  $(K_0, K_1, \dots, K_{Nr})$

$w^0 = x$

for  $r=1$  to  $Nr-1$

$u^r = w^{r-1} \wedge K^{r-1}$

for  $i = 1$  to  $m$

do  $v_i^r = S(u_i^r)$

$w^r = v_{P(1)}^r, v_{P(2)}^r, \dots, v_{P(lm)}^r$

$u^{Nr} = v^{Nr-1} \wedge K^{Nr-1}$


for  $i = 1$  to  $m$

do  $v_i^{Nr} = S(u_i^{Nr})$

$u^{Nr} \wedge K^{Nr}$

} Nr-1 rounds


} last round



(Refer Slide Time: 06:16)

### Definition of SPN Ciphers

- Block length:  $lm$ ,  $l$  and  $m$  are integers
- Substitution,  $S: \{0,1\}^l \rightarrow \{0,1\}^l$ 
  - Known as S-Box
- Permutation,  $P: \{0,1\}^{lm} \rightarrow \{0,1\}^{lm}$ 
  - Known as P-Box
- Except the last round all rounds will perform  $m$  substitutions, using  $S$ , followed by a Permutation.



This is, this is not, this is a feistel cipher and not an substitution. This is not a feistel cipher; this is an s p n cipher. So, in case, when I am, when I am writing as an s p n cipher, then the substitution box that I am considering has to be bijective.

(Refer Slide Time: 06:30)

**Algorithm**

- Input,  $x: \{0,1\}^{lm}$ ,  $K_0: \{0,1\}^{lm}$
- Output,  $y: \{0,1\}^{lm}$
- Key-schedule: generates  $(K_0, K_1, \dots, K_{Nr})$

$W^0 = x$

**for**  $r=1$  to  $Nr-1$

$u^r = W^{r-1} \wedge K^{r-1}$

**for**  $i = 1$  to  $m$

**do**  $v_i^r = S(u_i^r)$

$W^r = V_{P(1)}^r, V_{P(2)}^r, \dots, V_{P(lm)}^r$

$u^{Nr} = V^{Nr-1} \wedge K^{Nr-1}$

**for**  $i = 1$  to  $m$

**do**  $v_i^{Nr} = S(u_i^{Nr})$

$y = V^{Nr} \wedge K^{Nr}$

Key Whitening

Nr-1 rounds

last round

So, if you remember definition of s p n ciphers, all the transformations were invertible transformations, right? Therefore, the s boxes also invertible here. So, we can write this as an in an algorithm formal so. Therefore, the input essentially is at 0 1 stream of how many bits? l m bits. Your k 0 or rather the corresponding input k is also an l m bit key. So, the output that you produce is also another l m bit output; I mean l m bit sub strings. So, the key, and, and imagine that the key schedule generates all the round keys, that is, it generates k 0 k 1 to k n r, ok?

So, how many round keys are generated? n r plus one. So, you can compare these algorithm or the way the algorithm has been stated with their corresponding description of the a e s algorithm. Even the, even the d e s, you can also, you can also represent in this format, ok?

So, just imagine how you can write the d e s, for example, in this format. So, I leave it to you an exercise; you can do it. Even although the s box of your d e s algorithm was actually an non-invertible s box, but you can actually describe this in this format. You maybe you can just think over it.

So, therefore, in this case, you see the w 0 is equal to x. Therefore, you takes the, you take the plaintext text or the input x and assign it to the variable w 0, and after that, what you do is that you apply the corresponding rounds, ok?

So, if we apply round 1 to round  $n - r - 1$ , so, what you do in the rounds? You take the state  $w_{r-1}$ , you ex-or it with the corresponding key, right? You mix the key. So, you take  $k_{r-1}$  and you ex-or that. So, this symbol represents an ex-or operation, and then what you do is that for each of the  $s$  boxes and the  $m$   $s$  boxes, you take the corresponding component and I denote that component or rather the, I denote the input to the  $s$  box by the variable  $u$ , and  $u_r$  means the  $r$ th round and  $I$  means the,  $i$ th,  $i$ th block, the  $i$ th or rather the  $i$ th word. So, there are how many  $s$  boxes there? There are  $m$   $s$  boxes, so there are  $m$  words.

So, I take  $u_i$  odd and I apply that on the, I mean pass it through the  $s$  box, am I obtain  $v_i$   $r$ . So, that is the output of the  $s$  box. I can do it for all the  $s$  boxes, that is, for, for all the  $m$   $s$  boxes. Then the output that you get, which, which I represent as, for example, represent by or denote by the variable  $v$ , you take that and perform a permutation and a transposition. So, remember the transposition can be also denoted in this way. It is just a permutation of the bits, and there are how many bits? There are  $l$   $m$  bits, right?

So, you take this  $l$   $m$  bits, and, and assume a permutation denoted by  $p$  and perform a permutation over the bits. So, therefore, this is a corresponding output. So, you keep on repeating this, and finally, when you you do an ex-oring with the  $n - r - 1$ th key, and then, you remember the that in the last round, that is only the  $s$  box application, right? So, there is no permutation. So, you just do the  $s$  box; you just obtain the output of the  $s$  box in the last round, ok?

So, in this case, the key whitening steps are two - this is the key whitening step and this is the key whitening step. (Refer Slide Time: 09:30) These two steps are commonly known as the key whitening step, ok?



(Refer Slide Time: 09:46)

### Example: GPig Cipher

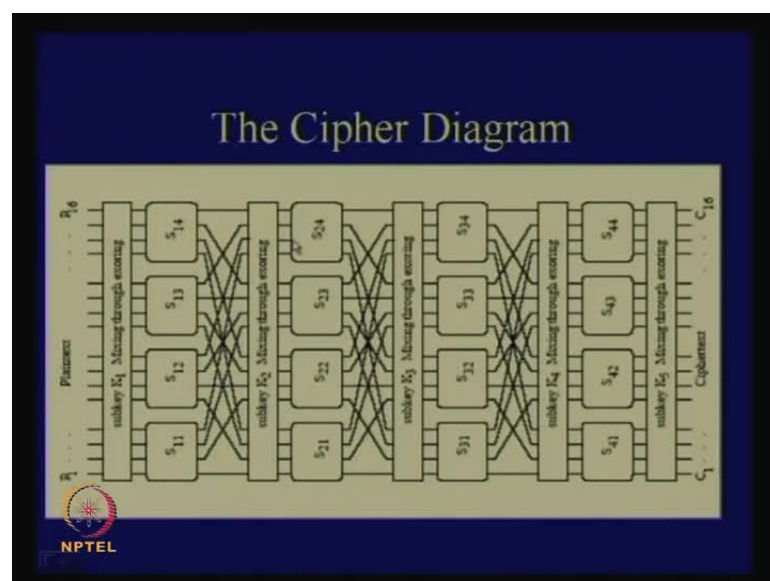
- $l=m=Nr=4$
- Thus plain text size is 16 bits
- It is divided into 4 groups of 4 bits each.
- S-Box works on each of the 4 bits
- Consider a S-Box (substitution table)

Table 1: S-box Representation (in hexadecimal)

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

NPTEL

(Refer Slide Time: 09:51)



This perform to do the next mixing perform to do mixing with the key. So, therefore, [pict/pictographic] I mean, I mean, I will see, so, therefore, if I represent this in a pictographic format, this would, this is how it will look like. So, you take a plaintext and this is the corresponding  $k_1$ ; I mean you can imagine that is the, key, key layer.

This is the another key layer; this is the another key layer and so on. (Refer Slide Time: 10:01) There are some key layers and there are four s boxes. So, you can just imagine a very simple version of that cipher. So, there are four s boxes and each of the s boxes

work on four bits. Therefore, it also produces another four bit output. Now, these outputs are being transposed by a permutation box, ok?

So, therefore, this wiring represents a permutation transposition. So, you obtain next output, that is the next, and therefore, you again perform a key mixing, again apply an s box, again do a transpose and you keep on doing that, ok?

(Refer Slide Time: 10:45)

**Example: GPig Cipher**

- $l=m=Nr=4$
- Thus plain text size is 16 bits
- It is divided into 4 groups of 4 bits each.
- S-Box works on each of the 4 bits
- Consider a S-Box (substitution table)

Table 1: S-box Representation (in hexadecimal)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

NPTEL

So, this is how you, I mean cryptographically would have represented the, [ci/cipher] the, the cipher, and I call as a g pig cipher, because it is a gini-pig cipher in our case. So, this, this cipher, essentially, essentially, so, if I write, I mean in terms of the variables  $l$   $m$  which we have denoted, what is the value of  $l$  and  $m$ ? All of them are equal to four and the number of rounds also equal to four. So, I am considering a four round cipher, ok?

So, the plaintext is therefore 16 bits because that is  $l$   $m$  bits. So, therefore, it acts up on a 16 bit block and it is divide into 4 groups of 4 bits each. The s box works on each of the 4 bits, and therefore, you can just take an example s box. So, this, have an, have an example s box could be like, ok?

So, therefore, this is just a substitution table. You take all the 0, 1, 2 and so on. The elements will run till f, right? Because there are [six/sixteen] sixteen elements possible from zero to fifteen and imagine these are the corresponding outputs.

(Refer Slide Time: 11:50)

### Example: GPig Cipher

- $l=m=Nr=4$
- Thus plain text size is 16 bits
- It is divided into 4 groups of 4 bits each.
- S-Box works on each of the 4 bits
- Consider a S-Box (substitution table)

Table 1: S-box Representation (in hexadecimal)

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

NPTEL

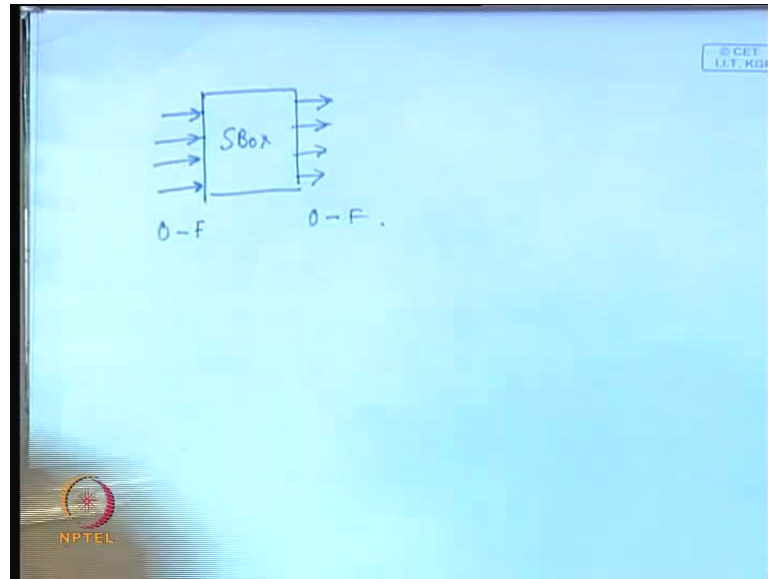
So, this is the, this is the substitution of that of the inputs. So, this is the s box table. So, you can represent the table in the form of an, I mean you can represent the s box in the form of a table, right? So, the permutation table also will look like this. So, permutation acts is the just transposition on the bits, and how many bits do you have? You have got 1 to 16 bits, right? And therefore, you perform a transposition, and **this is how...**

**(( ))** so, we will take 0 1 2 perform S-Box transposition.

Yeah

**(( ))**

(Refer Slide Time: 12:30)



No, see, I mean if you just see the table, observe the table. It should be a substitution of elements from 0 to 15, right? Your s box acts upon how, [ma/many] how many elements. I think this should be, this which are clear till now. So, we have an s box, right? So, your s box acts upon how many elements? Four, four bit elements and it produces another 4 bit output. So, I am considering a bijective s box. So, what, what are the possible values of your input? The input can run from 0 to f; the output can also be 0 to f, but the only thing is that when I am considering a bijective mapping, it has to be a one to one mapping, right?

(Refer Slide Time: 13:00)

### Example: GPig Cipher

- $l=m=Nr=4$
- Thus plain text size is 16 bits
- It is divided into 4 groups of 4 bits each.
- S-Box works on each of the 4 bits
- Consider a S-Box (substitution table)

Table 1: S-box Representation (in hexadecimal)

input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

NPTEL


(Refer Slide Time: 13:14)

### GPig (contd.)

- The Permutation Table is as follows:

input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

- Permutation is the transposition of bits
- There are  $1m=16$  bits, which are transposed using the above table




So, therefore, you, now, if you see the table, it is exactly a one to one, one to one mapping. So, there are no repetitions. So, each, each, each of [f]- I mean this row and this row all of them are just permutations of the numbers from 0 to f, is it clear? So, this is your **[sub/substitution]** s box table and permutation table, ok.

(Refer Slide Time: 13:21)

### Modifications or Variations of the SPN Structure

- Examples: DES, AES
- Different S-Boxes instead of a single one
  - As done in DES, there are 8 different S-Boxes
- Have an additional invertible linear transformation
  - As done in AES
- Is the GPig Cipher secure?

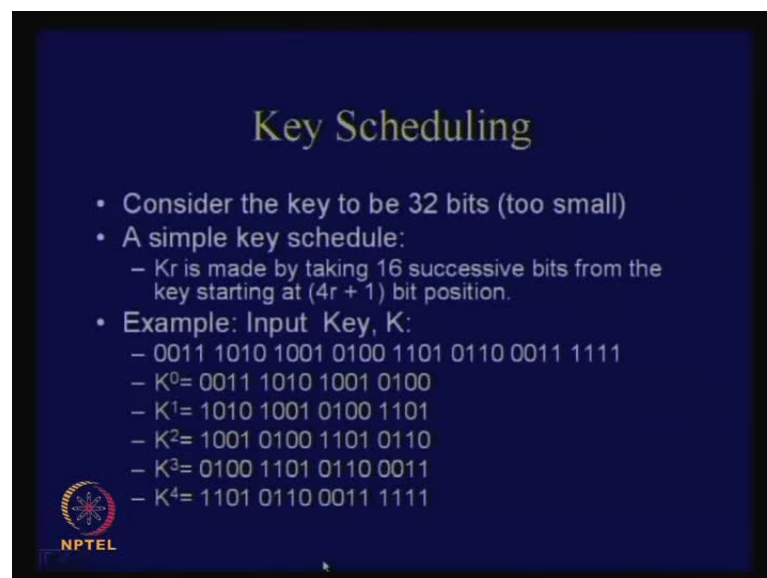


So, therefore, the, bit, bit reflection that your examples of d e s and a e s. So, these are different s boxes have been used. So, there are, I mean if you modify or you vary this structures slightly, you can actually describe your d e s and also your a e s algorithms.

So, [diff/different] in, in case of for example, d e s not only one s box are used but there are actually eight different s boxes, and may be in case of a e s, you do not have only one linear transformation but you have an additional invertible linear transformation. So, therefore, you know [than/that] a e s that are two linear transformation, right? One in the shift row and other one is a mix column.

But the, [ques/question] the point is that you can actually represent the a e s and the d e s in such kind of, using such kind of structure. So, therefore, the question that we will address in today's class is the gpig cipher secure.

(Refer Slide Time: 14:40)



**Key Scheduling**

- Consider the key to be 32 bits (too small)
- A simple key schedule:
  - $K_r$  is made by taking 16 successive bits from the key starting at  $(4r + 1)$  bit position.
- Example: Input Key,  $K$ :
  - 0011 1010 1001 0100 1101 0110 0011 1111
  - $K^0 = 0011 1010 1001 0100$
  - $K^1 = 1010 1001 0100 1101$
  - $K^2 = 1001 0100 1101 0110$
  - $K^3 = 0100 1101 0110 0011$
  - $K^4 = 1101 0110 0011 1111$

NPTEL

So, we will try to find out a technique through which we can analyze the security of the gpig cipher. So, it is a toy example, no doubt, but we can probably scale; we can easily scale this approach, and the security of the cipher whether it is good or bad depends upon, how, how good it fairs again this attack. They are very important class of attack. So, [con/consider] so, another thing which is given to be told is that when I told about the cipher, I told about the round structure, but we have to also address the key scheduling, right?

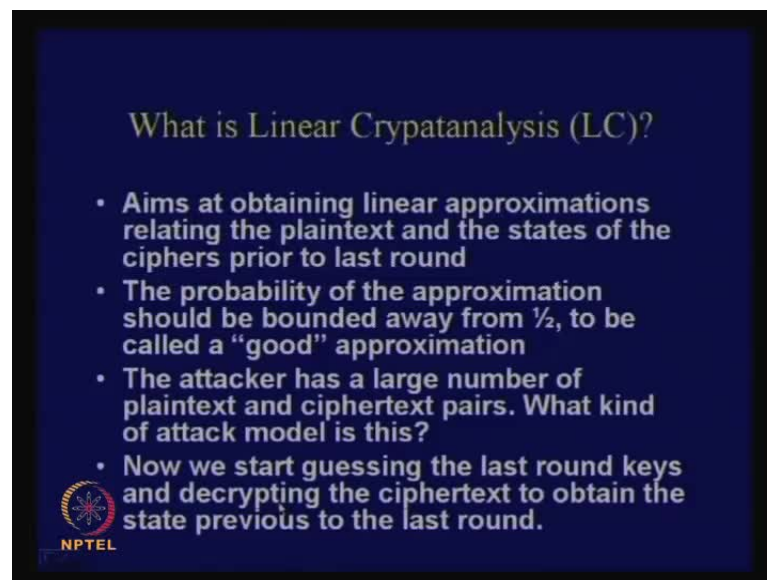
So, imagine that you have a thirty two bit key, it is too small but it is ok for a toy cipher. So, what you do is that as simple, you take a simple key schedule, and what you do is that, you, you know that  $k_r$  is made of, [six/sixteen] is made by taking sixteen successive bit from the key starting at four  $r$  plus one bit position. So, what does it mean? Suppose

this is your input key, right? Therefore, input keys are thirty two bits, so what you do is that you assign your  $k_0$  and you start from the 0, from the, from the first bit. So, you take this and you assign, see you assign sixteen bits. So, you have to assign this, this, this and this to your  $k_0$ . (Refer Slide Time: 15:18) For your  $k_1$ , you start from here, you take 1 0 1 0 and you start it again, you take four words.

So, you take 1 0 1 0 1 0 0 1 0 1 0 0 and 1 1 0 1 and you assign that to your  $k_1$ . Similarly, for  $k_2$ , you start from, this, this point, this location. Therefore, you see that you are starting from four  $r$  plus 1, because when your round is 0, you are starting from one first bit. When your round is one, that is,  $r$  is equal to 1, you start from the fifth bit location. So, that is this point. Therefore, you start from the next word essentially.

So, you can assign an, an, and you can do this and you can obtain all the round keys. You can, you can obtain  $k_0$ ,  $k_1$ ,  $k_2$ ,  $k_3$  and  $k_4$ . So, key scheduling is just that, right? Therefore, you just take the input, input key, and from there, somehow you have to create the round keys, ok?

(Refer Slide Time: 16:26)



This is very simple key scheduling algorithm but it is ok for our purpose. So, all of us follow the key scheduling algorithm. So, now, what is linear cryptanalysis? So, the [li/linear] objective of linear cryptanalysis is to obtain linear approximations relating the plaintext and the states of the cipher prior to the last round. So, for example, in our toy cipher, we have we have four rounds.

So, what we will try to do is that we will try to create linear approximations. Why do I say linear approximations? Because the s box is non-linear, right? There are no linear. So, it is, if the s box is properly defined or properly designed, then there are no linear expressions which can actually, I mean linear expressions in terms of the input variables and the output variables of your s box, ok?

You can only at best say that this linear approximation holds with so much probability but not with a probability of one, because if it holds with probability of one; it means your s box is linear, but as I told you in from the beginning that your s box is a non-linear component, right?

So, therefore, our objective will be to take the inputs, that is, the plaintext and the states of the cipher prior to the last round and obtain a linear approximation. The probability of the approximation, so, there is a probability should be bounded away from half to be called a good approximation.

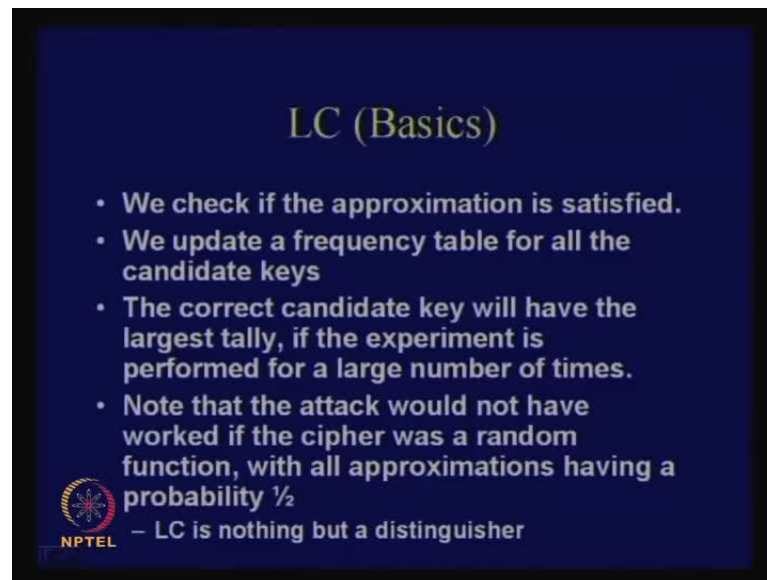
So, therefore, if your probability is say  $p$ , so in ideal case, for, from the point of view an attacker, the probability should be one, right? So, your, the, so a difference from a half is actually equal to half, so that is the maximum. Similarly, if there is an expression in which holds a linear expression, which holds with the probability of zero, what does it mean? The naught of that holds with the probability of one. Therefore, what we are interested is the is on the deviation from half, ok?

So, the attacker has got a large number of plaintext and cipher text. So, therefore, the attacker [ob/obtains] obtains a large number of plaintext and cipher text and carries on, carries the, carries on the attack. So, what kind of attack module is this? This is the known plaintext attack.

So, now, we start guessing the last round keys and decrypting the cipher text to obtain the state previous to the last round. So, what we do is that once we have got this approximation, we start to guess the last round key and, we, we decrypt the cipher text and see whether the, and we obtain the state of the last round, and see, now, since we have the plaintext and also the state [out/output] output of the last round, we see whether the expression that we have found out holds or not. If it holds, we keep it as a possible key or otherwise we throw away the key. So, this is our strategy, ok?



(Refer Slide Time: 18:59)



LC (Basics)

- We check if the approximation is satisfied.
- We update a frequency table for all the candidate keys
- The correct candidate key will have the largest tally, if the experiment is performed for a large number of times.
- Note that the attack would not have worked if the cipher was a random function, with all approximations having a probability  $\frac{1}{2}$

NPTEL – LC is nothing but a distinguisher

So, we will see more details on this. So, we check if the approximation is satisfied, we update a frequency table for all the candidate keys. The correct candidate key will have the largest tally if the experiment is performed for a large number of times. So, this is the statistical analysis. So, therefore, you know in statistics, we have to take large number of samples for something to hold, right?

So, therefore, we have to keep on repeating this for large number of times. You note that the attacker would, the attack would not have worked if the cipher was a random function with all approximations having a probability of half, ok?

So, you note that if your cipher was actually a random function, suppose random function is the same that from the input, it [provide/produce] it produce your random output. Then these approximations would have held with because what is the output of your linear, of, of your of your linear approximation? It is a, binary, binary expression, right? So, the output is either 0 or the output is either 1.


So, in case of a random function, half of the times it will be 0 and half of the times it will be 1, right? Therefore, the probability that the attack, [I mean the, that, that, if], if the cipher was modeling actual random function, the expression, the probability that an expression would be either 0 or 1 was actually equal to half, ok?

So, what linear cryptanalysis does, is tries, it tries to distinguish the cipher from a random function and that is the underlying principle of all class of attacks. It tries to distinguish a given instance from a random function. So, a, [ran/random] say a random function is very hard to actually obtain, ok?

(Refer Slide Time: 20:54)

**Piling Up Lemma**

- Consider independent random variables:
  - $X_1, X_2, \dots$
  - let  $\Pr[X_1=0]=p_1 \Rightarrow \Pr[X_1=1]=1-p_1$
  - let  $\Pr[X_2=0]=p_2 \Rightarrow \Pr[X_2=1]=1-p_2$
  - Thus,  $\Pr[X_1 \wedge X_2]=0$  is  $p_1 p_2 + (1-p_1)(1-p_2)$
  - Not let  $\epsilon_1=p_1-1/2$  and  $\epsilon_2=p_2-1/2$   
(these are called bias values of the rv.s)
  - Thus,  $\Pr[X_1 \wedge X_2]=0 = 2\epsilon_1\epsilon_2$

 NPTEL

So, what we are trying to do is that we are trying to make it as look like random, right? and therefore, the objective or property or underlying principle of any attack is to find out that property which distinguishes the corresponding cipher from a random function and this is one way of doing it and very interesting way of doing it, but in order to do that, we require a lemma and we call that lemma as a piling up lemma. So, we will try to see what is the piling up, what is the statement of a piling up lemma, ok?

So, you consider independent random variables. So, first of all consider two independent random variables, and I denote them by  $x_1$  and  $x_2$ . So, so, imagine that suppose assume that the probability that  $x_1$  is equal to 0, this random variable takes the values of 1 and 1.

So, imagine that your probability that your  $x_1$  is equal to 0 is equal to  $p_1$ . So, [there/therefore] therefore, the probability that  $x_1$  is equal to 1 is automatically equal to  $1 - p_1$ , right. So, also assume that your probability that your  $x_2$  is equal to 0 is equal to  $p_2$ , and therefore, at the probability that  $x_2$  is equal to 1 is equal to  $1 - p_2$ .

So, what is the probability that  $x_1$  ex-or with  $x_2$  is equal to 0? It is equal to, so when can  $x_1$  ex-or  $x_2$  to be equal 0? Both of them are 0 or both of them are 1. So, it is equal to  $p_1$  multiplied with  $p_2$  plus  $1$  minus  $p_1$  multiplied with  $1$  minus  $p_2$ .

So, if you rearrange this or rather if you substitute that epsilon 1, so, since we are interest in the deviation of the probability from half, so, you just substitute that epsilon 1 and make epsilon 1 equal to  $p_1$  minus half and epsilon 2 is equal to  $p_2$  minus half.

(Refer Slide Time: 23:00)

The image shows a handwritten derivation on a blue background. At the top right, there is a small logo for '© CET IIT, KGP'. The derivation starts with the definitions of bias:  $p_1 = \epsilon_1 + \frac{1}{2} \Rightarrow 1 - p_1 = \frac{1}{2} - \epsilon_1$  and  $p_2 = \epsilon_2 + \frac{1}{2} \Rightarrow 1 - p_2 = \frac{1}{2} - \epsilon_2$ . The next line shows the expansion of the XOR probability formula:  $p_1 p_2 + (1 - p_1)(1 - p_2)$ . This is then expanded into  $(\epsilon_1 + \frac{1}{2})(\epsilon_2 + \frac{1}{2}) + (\frac{1}{2} - \epsilon_1)(\frac{1}{2} - \epsilon_2)$ . The result is  $2\epsilon_1\epsilon_2 + \frac{1}{2}$ , where  $2\epsilon_1\epsilon_2$  is circled. Below this, a diagram shows  $x_1 \oplus x_2$  with an arrow labeled 'bias' pointing to  $2\epsilon_1\epsilon_2$ . In the bottom left corner, there is an NPTEL logo.

So, this deviation from half, we often referred to as the bias value. So, it is the bias of the random variable  $x_1$  and this is the bias of the random variable  $x_2$ . So, therefore, if I, represented, represent now this expression in terms of epsilon 1 and epsilon 2, this is what I obtain - it is equal to two epsilon 1 into 2 epsilon 2; I mean it is equal to two epsilon 1 into epsilon 2. It is quite easy to understand. Do you see why? See, your  $p_1$  is equal to epsilon one plus half and your  $p_2$  is equal to epsilon 2 plus half, and what we had was  $p_1 p_2$  plus  $1$  minus  $p_1$  into  $1$  minus  $p_2$ .

So, what is  $p_1$  into  $p_2$ ? It is epsilon 1 plus half into epsilon 2 plus half, and what is  $1$  minus  $p_1$ ?  $1$  minus  $p_1$  is equal to half minus epsilon 1, and what is  $1$  minus  $p_2$ ? It is also equal to half minus epsilon two.

So, you plug these values, it is half minus epsilon one into half minus epsilon two. So, what do you obtain? You see that epsilon 1 epsilon 2 states and actually it is equal to

2epsilon 1 plus 2 epsilon 2 because you collect from here and here, and other thing which you get is equal to plus half, the other, other, terms cancel.

So, now, what is the deviation of this corresponding output from half? It is equal to 2 epsilon 1 2 epsilon 2, right? So, the output, that is, the bias of the expression x 1 ex-ored with x 2 the corresponding bias value is equal to 2 epsilon 1 into epsilon 2, right? So, if you repeat this, you can, you can apply mathematical induction and you can obtain it for x 1 ex-or x 2 ex-or three also, right? And you can continue in this fashion for n variables, right?

(Refer Slide Time: 24:45)

**Generalized lemma**

**Lemma 1** [1] For  $n$  independent, random binary variables  $X_1, X_2, \dots, X_n$ , with bias  $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ .

$$Pr(X_1 \oplus \dots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^n \epsilon_i$$

Thus if  $X_1, X_2, \dots, X_n$  are  $n$  linear approximations then the bias of the linear approximation made out of these  $n$  equations is denoted by [2]:

$$\epsilon_{1,2,\dots,n} = 2^{n-1} \prod_{i=1}^n \epsilon_i$$

Note that if there is one bias on the RHS which is 0, then LHS is also 0

NPTEL

So, if you do that, this is what you obtain that the probability that x 1 ex-or x 2 ex-or x 3 ex-or so until x n, this should be equal to half plus 2 to the power of n minus 1 into product of all the [bi/bias] individual bias values, ok?

So, here bias, if you are considering say n number of n variables is actually equal to two to the power of n minus 1 into product of all the individual biases. So, you can just plug the value of n equal to 2 and check that. This is actually equal to the previous case, because n if I plug n equal to 2, this becomes equal to 2 and there are two product terms, ok?

So, thus if x 1 x 2 and x n are linear approximations, then the bias of the linear approximation made out of these n equations is denoted by this formula. Now, imagine

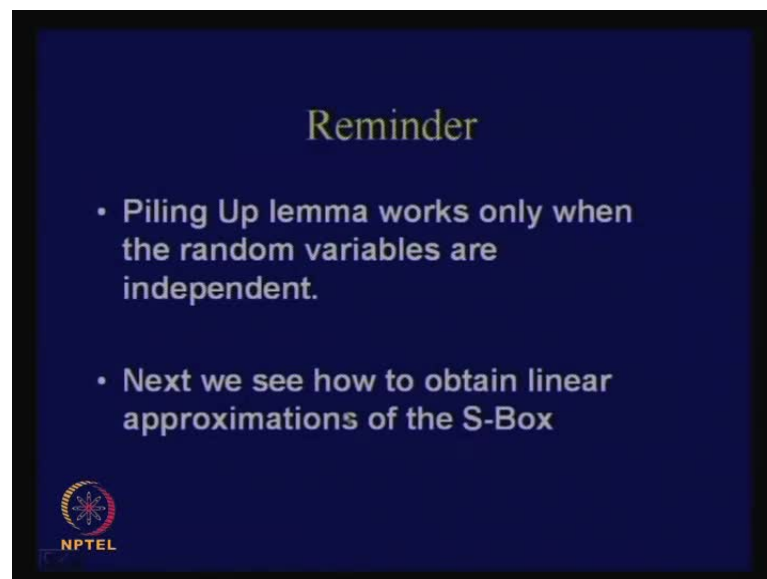
that all of these  $x$  random variables like  $x_1, x_2$  and so on. Till  $x_n$  and nothing but linear expressions, and what I am interested is, is, in computing the bias of the ex-or of this linear expressions. So, what is the corresponding bias value for that? So, in order to do that, what we do is that we compute the individual bias values of the expressions and then multiply the bias values, and then multiply that with two power of  $n$  minus one and that should give me the bias of the corresponding output, ok?

Note that if one of the bias values is 0, then your corresponding bias becomes equal to 0. Therefore, what does bias being equal to 0 mean?

(( ))

Yeah. So, probability is half. So, probability equal to half means what? It is a linear expression, so you can imagine, you can just try with some examples and you will figure out that. So, that means if there is one single linear value and if you combine that, then you obtain again a linear expression. So, you just figure out that if one of the bias values is equal to 0, then the left hand side also computes to 0.

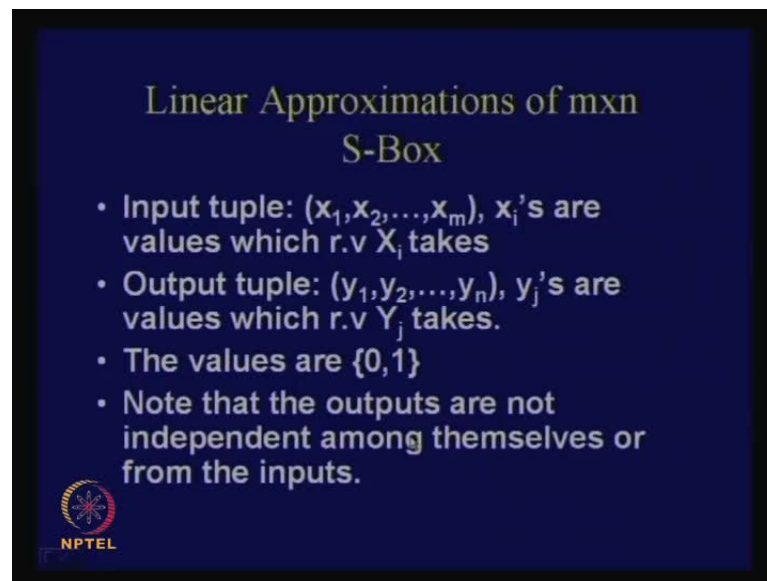
(Refer Slide Time: 26:54)



So, one reminder that we have to keep in mind is that the piling up lemma works only when the random variables are independent. So, we have, because we have multiplying the probabilities, right? So, see, in our case, we will assume this and go ahead, because essentially the idea is that since we are doing a key scheduling. If the key scheduling is

proper, then all the round keys should be independent, and after I ex-or with the round keys, this state variables which I obtain, all of them are independent. So, they were is an assumption and this assumption works in practice. So, next we see how to obtain linear approximations of the s box. So, this is actually quite central to our discussion. So, please pay attention to this.

(Refer Slide Time: 27:38)



The slide has a dark blue background with yellow and white text. The title is 'Linear Approximations of mxn S-Box'. Below the title are four bullet points. In the bottom left corner, there is a small circular logo with a star-like pattern and the text 'NPTEL' below it.

### Linear Approximations of mxn S-Box

- Input tuple:  $(x_1, x_2, \dots, x_m)$ ,  $x_i$ 's are values which r.v  $X_i$  takes
- Output tuple:  $(y_1, y_2, \dots, y_n)$ ,  $y_j$ 's are values which r.v  $Y_j$  takes.
- The values are  $\{0, 1\}$
- Note that the outputs are not independent among themselves or from the inputs.

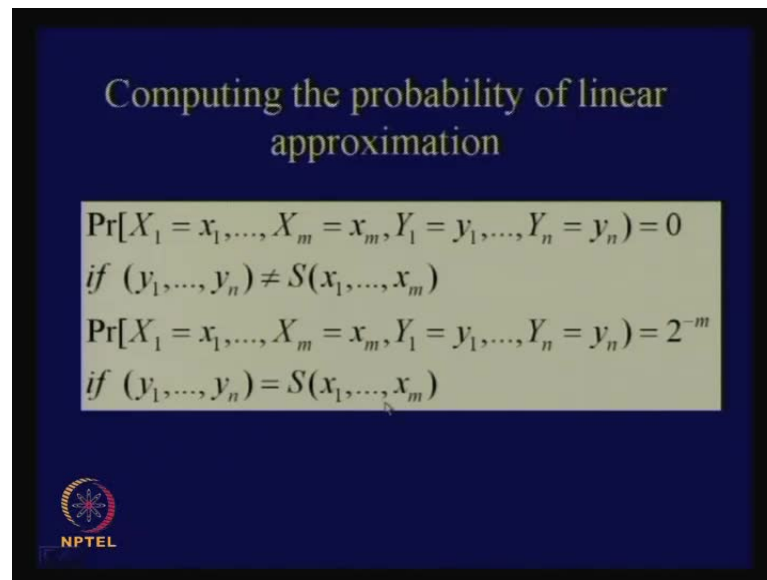
NPTEL

So, therefore, you can actually represent your s box as an m cross n table, right? So, what we do is that you can represent your input as [fro/from] running from  $x_1$  to  $x_m$ , and  $x_i$ 's are the values which the corresponding random variable  $x_i$  or capital  $x_i$  take.

Similarly, your output, you can represent, in, in the form of  $y_1$  to  $y_n$  and each of these  $y_j$ 's are the values which the corresponding random variable denoted by capital  $y_j$  take, right? So, your s boxes essentially a mapping from  $x$  to  $y$ , right? And these values, that is, random variables can take values 0's and 1's.

So, note that the outputs are not independent among themselves or from the inputs; that means that if you fix the input, the output is decided, right? So, in, in the s box, if I fix the input, the output is automatically decided, right?

(Refer Slide Time: 28:34)



Computing the probability of linear approximation

$$\Pr[X_1 = x_1, \dots, X_m = x_m, Y_1 = y_1, \dots, Y_n = y_n] = 0$$

if  $(y_1, \dots, y_n) \neq S(x_1, \dots, x_m)$

$$\Pr[X_1 = x_1, \dots, X_m = x_m, Y_1 = y_1, \dots, Y_n = y_n] = 2^{-m}$$

if  $(y_1, \dots, y_n) = S(x_1, \dots, x_m)$

NPTEL

So, therefore, if I would like to compute the probability of this fact, that is, your  $x_1$  or random variable  $x_1$  is equal to the [va/value] value  $x_1$ . Similarly,  $x_m$  the random variable  $x_m$  is equal to the value  $x_m$ , and your  $y_1$  is equal to  $y_1$  and so on;  $y_n$  is equal to  $y_n$ . Now, I, I need to compute this particular probability.

So, there are two cases - since I have fix the values of the input  $x$  the corresponding  $y$  that I am interested in may be a real output of the  $s$  box or, may, may not be the output of the  $s$  box, right? So, if it is actually I, if, if I fix this values in into the  $s$  box, if this output does not occur, then this probability computes to 0, right? This can never happen, and if this happens, that is, if, indeed if I fix the values of input to  $x_1$  to  $x_m$ , if the output is really  $y_1$  to  $y_n$ , then what is the probability? The probability is 2 to the power of minus  $m$ . You see, we are not doing two to the power of minus  $m$  plus  $n$ , why? Because these variables, that is, the  $y$  is are dependent upon these values of  $x$ , ok.

Once we have fixed these values, these gets automatically decided, right? So, therefore, this probability is equal to either 0 or 2 power minus  $m$  depending upon whether your, [in/indeed] whether your, indeed, indeed fixing upon the  $x$  is gives you the corresponding wise.

(Refer Slide Time: 30:01)

**S-Box in terms of the random variables**

$X_1$	$X_2$	$X_3$	$X_4$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

What is the bias of  $X_1 \oplus X_4 \oplus Y_2$ ?


There are 8 cases when  $X_1 \oplus X_4 \oplus Y_2 = 0$

Thus the probability is  $8/16 = 1/2$

So, the bias is zero.

Consider,  $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$

The bias turns out to be  $-3/8$



So, let us consider a very simple example. So, the idea is that your inputs like  $x_1, x_2, x_3$ , and  $x_4$ , and your corresponding outputs of the s box and your outputs are denoted by  $y_1, y_2, y_3$ , and  $y_4$ .

So, this is the table, and I am highlighted certain portions of this table because you will be concentrating on these columns and rows. So, I am interested in computing an example of bias value, say  $x_1$  ex-or with  $x_4$  ex-or with  $y_2$ , ok?

So, what is the bias value of this expression? What does it mean? It means I am interested in computing the number of cases when  $x_1$  ex-or  $x_4$  ex-or  $y_2$  is equal to 0. So, you can observe that in this truth table, these yellow lines, that is, the yellow rows that that have drawn with, they are the cases that are interesting to be, ok?

Why you can see that in all these cases, here  $x_1$  is equal to 0; it is one over here, and it is one over here, right? So, if I take an ex-or, I obtain 0. What about this row? This is 0; this is 0, and this is 0; so, it is still zero, right? (Refer Slide Time: 31:27) Similarly, you can work it for the other rows also.

So, let us do with this it is 0. This is 1 and this is 1. So, if I ex-or, I obtain 0, right? So, for all these yellow lines, and there are how many yellow lines? One, two, three, four, five, six, seven, eight; so, in this eight lines, these expressions is satisfied, but what about the other lines? The expression is not satisfied.



So, we can see for example, the first [colu/column], the first row, it is 0 here; it is 0 here and it is 1 here; so, the expression is not satisfied, right? So, you see that this is our linear expression and your table although it is a non-linear mapping. There are some cases, for which, it will get satisfied, and for some cases, it does not get satisfied, ok?

So, what is the probability that it get satisfied? It is actually equal to eight by sixteen that is equal to half. So, what is the bias value? The bias is 0. So, this, the [pro/probability], the bias of this linear expression is actually equal to 0. So, consider  $x_3 \oplus x_4 \oplus y_1 \oplus y_4$ .

So, this is another example which you can work, and I am, although I am not showing it actually, the bias in this case will turn out to be straightly more complicated. It is actually equal to minus three by eight. So, which means the probability is half minus three by eight, ok?

So, you can just check whether it is or not. So, so,  $x_3 \oplus x_4 \oplus y_1 \oplus y_4$ , therefore, out of these two expressions, from the point of view of the attacker, these expression is more interesting to be than this expression, because the bias value of these expression is very small, whereas, the bias value of this expression is quite high, so, actually, equal to minus 3 by 8.

(Refer Slide Time: 33:36)


**Representing the Approximations**

- Any expression can be written in the form:

$$\left( \bigoplus_{i=1}^4 a_i X_i \right) \oplus \left( \bigoplus_{i=1}^4 b_i Y_i \right)$$

- Here  $a_i \in \{0,1\}$  and  $b_i \in \{0,1\}$
- Thus each of a and b can be denoted by hexadecimal numbers from 0 to F

They can be stored in a table



I mean so note that I am interested in the absolute value of the bias, and the bias in this case is minus 3 by 8. So, the absolute value is 3 by 8 which is quite high, and the bias in this case is 0, right? So, with this observation, we see that I can actually obtain expressions of this form.

(Refer Slide Time: 34:18)

### S-Box in terms of the random variables

$X_1$	$X_2$	$X_3$	$X_4$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

What is the bias of  $X_1 \wedge X_4 \wedge Y_2$ ?


There are 8 cases when  $X_1 \wedge X_4 \wedge Y_2 = 0$

Thus the probability is  $8/16 = 1/2$

So, the bias is zero.

Consider,  $X_3 \wedge X_2 \wedge Y_1 \wedge Y_4$

The bias turns out to be  $-3/8$



(Refer Slide Time: 34:21)


$$X_1 \oplus X_4 \oplus Y_2 = 0$$

$$\Leftrightarrow (X_1 \oplus X_4) \oplus (Y_2) = 0.$$

$$\rightarrow a = (1001)$$

$$\rightarrow b = (0100).$$

$$\bigoplus_{i=1}^n a_i x_i \oplus \bigoplus_{i=1}^m b_i y_i = 0$$



So, so any expressions can be written in this form like you take, you take ex-ors and you do a  $x_i$  ex-or with ex-ors  $b_i y_i$ . So, here  $a_i$  belongs to 0 1, and  $b_i$  also belongs to 0 1. Thus each of a and b can be denoted by a hexadecimal number starting from 0 to f and

they can be stored in the form of a table. Do you see that? So, you see when I am talking about  $x_1$  XORed with  $x_4$  XORed with  $y_2$  equal to 0. I could have written this expression like this also, right?  $x_1$  XORed with  $x_4$  XORed with  $y_2$  equal to 0.

So, now, if you take the s box and number them like  $x_1, x_2, x_3, x_4$ , and  $y_1, y_2, y_3$ , and  $y_4$ , so, [thi/this] this particular expression  $x_1$  XORed  $x_4$ , I can represent that as, you can take a another hexadecimal value which is equal to 1 0 0 1, and you can take b which is equal to 0 1 0 0, and now, consider sigma or rather you consider a big sigma of  $a_i \oplus x_i$  XORed with a big sigma  $b_i \oplus y_i$  equal to 0. So, I can represent my expression in this format also, right? Where  $i$  runs from 1 to 4 or and this expression also has got a similar sigma.

So, therefore, this runs from  $i$  equal to one to four, you see that? So, therefore, now, I can actually represent this in the form of a table and all such kind of approximation, that is, all such kind of linear expressions, I can represent in the form of the hexadecimal values  $a$  and  $b$ .

So, for example, this expression is for  $a$  and  $b$ , for the choice of  $a$  and  $b$ , right? So, I can find similar kind of expressions by varying this values of  $a$  and  $b$ , and how many values of  $a$  is possible? There are sixteen values of  $a$ , and how many values of  $b$  are possible? Sixteen values of  $b$ . So, if I represent all these expressions in the form of a table, how many, [ex/expressions] what is the size of the table? It is a sixteen cross sixteen table, right? And each and what I can do is that in the table, each element will represent the number of times it actually matches and number of times it does not matches, match. So, what I do is that conveniently I just store the only the bias value and that is number of times it deviates from half.

(Refer Slide Time: 37:08)

### Linear Approximation Table (LAT)

a	b															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	6	4	6	6	8	10	8	8	6	12	6	6	6	8	10	8

for  $X^3 \oplus X^4 \oplus Y^1 \oplus Y^4$   
 $a=(0011)=3$   
 $b=(1001)=9$   
 Thus  $T[3,9]=2$   
 $Bias = 2/16 - 1/2 = -3/8$

Thus Bias  
 $= (T[a,b]/16) - 1/2$

So, therefore, if I represent this in the form of a table, this will look like this. So, you see that this is the value of b being varied and this is the value of a being varied. So, now, we again, if we look back into this expression  $x^3 \oplus x^4 \oplus y^1 \oplus y^4$  will be 0 0 1 1 by the [simi/similar] similar example as, what, what we have done for the previous case, and  $y^1 \oplus y^4$  will be 1 0 0 1. So, what is value of a 3 and 9? You just consider the third row and the ninth column.

So, what is the value? 2. So, you see that  $t_{3,9}$  is equal to two, and therefore, the bias in this case is actually 2 by 16 minus half. So, I can represent this in various formats. I can either store the value of 2 or I can store the bias value itself. So, in this case, I have actually not stored the bias value and this bias value comes out to 2 by 16 minus half and that is equal to minus 3 by 8. So, you, you remember the bias that we computed in the previous case.

So, I can obtain a bias in this fashion, and for a any other expression or any other value of a and b, the bias value is equal to  $t_{a,b}$  by sixteen minus half, and actually, there are lot many beautiful properties of this table. I, I can just show you certain things like may be if your, if your choice of a and b are both 0, then you can see that this values equal to sixteen. I guess you can easily figure out why, because if your a is zero, then all your cases you will get that value.

(Refer Slide Time: 38:55)

$X_1$	$X_2$	$X_3$	$X_4$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
0	0	0	0	.	.	.	.
0	0	0	1	-	.	.	.

$a=0$   
 $b=0$

So, you will get all of the expressions all of the sixteen values will get satisfied, right? So, what are you doing it choose an a and b and we are adding the excess, right? So, therefore, in your, [tru/truth] so, therefore, your, if your truth table was like this, so, here  $x_1, x_2, x_3$  and  $x_4$ , and your output was equal to  $y_1, y_2, y_3$  and  $y_4$ . So, what we were doing is that we were varying this - 0 0 0 0 0 1. So, this is the corresponding truth table and I have got individual values of the y's, right? So, note that if your a is equal to 0, then that means what? Your, it is always satisfied, right?

So, therefore, if I take a is equal to 0 and b is equal to 0, then all the time my corresponding expression is satisfied, right? So, you see that. So, there are how many possible times? There are sixteen possible times, because I am varying the excess and all the sixteen times, I find that there is a match, right?

(Refer Slide Time: 39:44)

### Linear Approximation Table (LAT)

a \ b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	6	4	6	6	8	10	8	8	6	12	6	6	6	8	10	8

for  $X^3 \wedge X^4$   
 $Y^1 \wedge Y^4$   
 $a=(0011)=3$   
 $b=(1001)=9$   
 Thus  $T[3,9]=2$   
 $Bias = 2/16 - 1/2 = -3/8$

Thus Bias  
 $= (T[a,b]/16) - 1/2$

(Refer Slide Time: 39:54)

© CET I.I.T. KGP

$x_1$	$x_2$	$x_3$	$x_4$	$y_1$	$y_2$	$y_3$	$y_4$
0	0	0	0	.	.	.	.
0	0	0	1	.	.	.	.

1 1 1 1

$\oplus a_i x_i = 0$

$a = 0$   
 $b = 0$   
 $N_L(a_i, 0)$   
 $a > 0$

So, therefore, this value being equal to sixteen is quite obvious, but can you figure out why these values are eights? So, let us consider this column. So, what I am, what I am interested, is, is in this value 1, I call that say for example, n 1, that is, number of times you have got actually a your column is fixed to 0 but your a is varying, right? And only thing is that your a is actually greater than 0.

So, so, what you have what are you doing? Here, in this case, you are you are keeping the a greater than 0 and you are varying the value of a, right? So, you note that, in, in the

corresponding truth table, that is, since you have got all possible values of four bit values, if I fix the value of a, then you will find that an expression of these type, that is, a  $\oplus$  b will be **act/actually** actually equal to 0 half of the times, and half of the times it will be equal to 1, right? So, the moment I have fixed the value of b to 0, so in my expression, the second part goes to 0, right?

So, what was my expressions format? It was like this - a  $\oplus$  b, right? so if I fix the value of b to 0, this term goes to 0, right? And I am interested in how many cases, this is equal to 0, and so, therefore, we, if, from here, we note that if I vary, there are total 16 number of possible cases, out of them, eight times this expression will hold and eight time this expression will not hold. So, therefore, this value of n l a, 0 should be equal to 8.

(Refer Slide Time: 41:31)

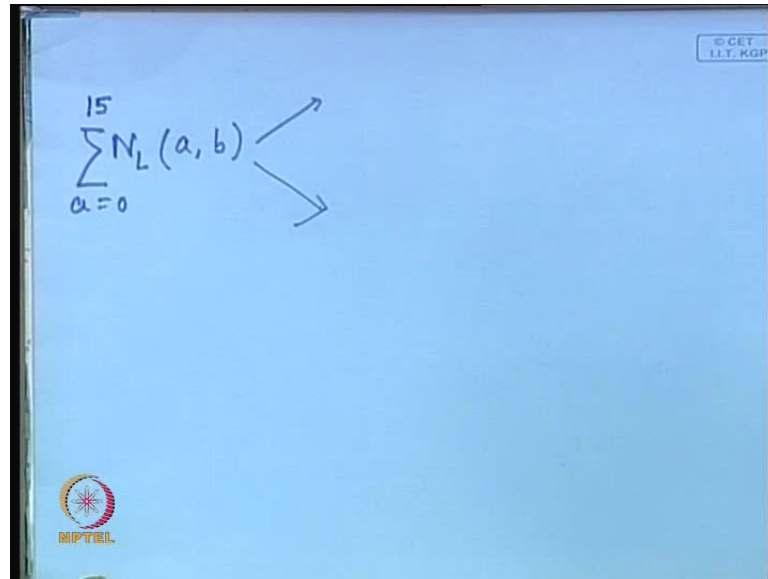
### Linear Approximation Table (LAT)

a \ b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8	8

for  $X^3 \oplus X^4 \oplus Y^1 \oplus Y^4$   
 $a=(0011)=3$   
 $b=(1001)=9$   
 Thus  $T[3,9]=2$   
 Bias =  $2/16 - 1/2 = -3/8$

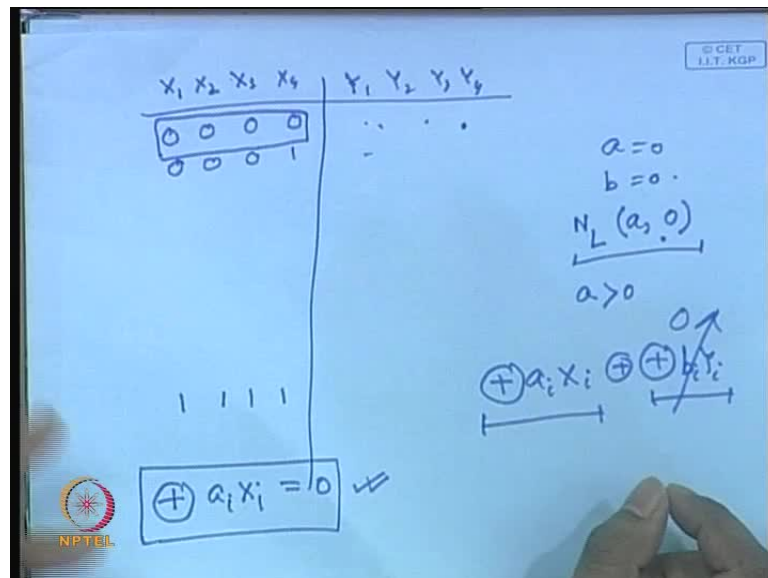
Thus Bias  
 $= (T[a,b]/16) - 1/2$

(Refer Slide Time: 41:43)



Another interesting property you can also find out. If I start adding up the columns, so you will find that if I start adding up the columns, that is, I keep on adding, so therefore, you will find that what I am interested is in computing  $n$  l a comma b and I do a sigma over all possible values of a which runs from 0 to 15.

(Refer Slide Time: 42:13)



So, in this particular case, you will find that your x that I can divide this particular case into two parts, that is, imagine that I am considering only the excess. So, therefore, you can imagine that your, so what your, so, you can imagine that suppose your x value



is equal to zero, the moment your x value is equal to zero, the assignment for y gets fixed because your s box is a fixed table, right?

So, therefore, the expression that you are considering is in this case, this becomes equal to zero, and this is some value, right? So, therefore, you have for various values of b, this will be certain things, right?

So, therefore, what I am interested is whether this is equal to zero or not, so that depends upon how your s box has been constructed, right? So, either in all the cases, so that that depends upon whether in all the cases, this can be equal to 0 or in none of the cases, this will be equal to 0, right?

(Refer Slide Time: 42:56)

$$\sum_{a=0}^{15} N_L(a, b)$$

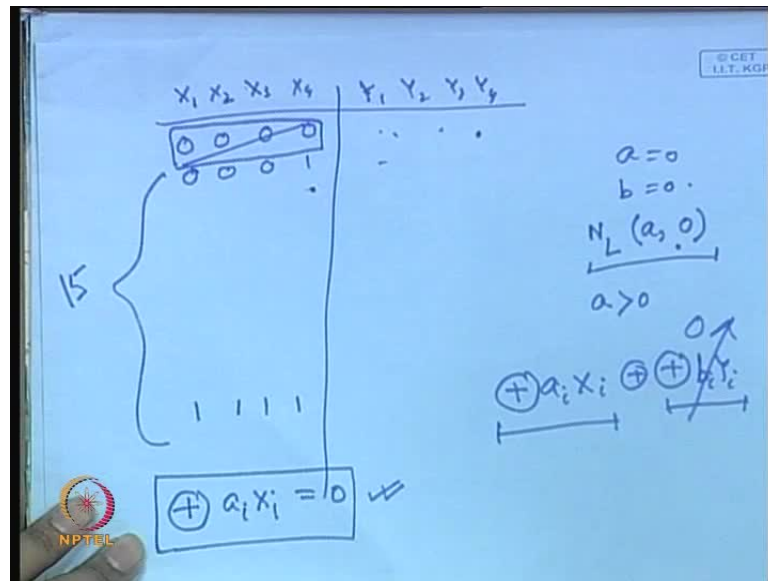
$x \neq 0$

$x = 0 \rightarrow 0 \text{ or } 16$

So therefore, that means that this value, if your corresponding x that you choose is 0, is either equal to 0 or it is equal to half of the cases, right?

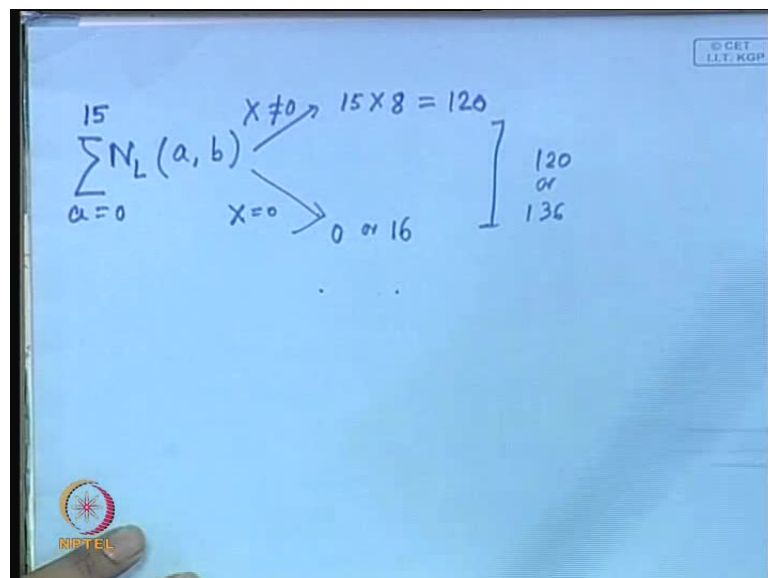
Therefore, it is either equal to zero or you will find that not half actually for all the cases. It will be satisfied for all the cases. So, it is either equal to 0 or it is equal to 16. That depends upon your s box table, right? So, the moment I have fixed your x to be 0, your output y is decided, right? So, what I am interested is that whether sigma or rather ex-or of b i y i is equal to 0, so that depends upon your s box. So, if the, if that is, so, then, it will happen for all the possible values or it will never occur.

(Refer Slide Time: 43:56)



So, therefore, yours therefore, there are, there are, there are two possibilities - either it will be equal to 0 or it will be equal to 16. What about when  $x$  is not equal to 0? So, if  $x$  is not equal to 0, then you will find out that actually, if I, so therefore, this is gone. So, I am not considering these cases. So, how many possible non-zero values are there? There are fifteen non-zero values. So, you take, one, one value of  $x$  and you vary all the all the choices of  $a$ .

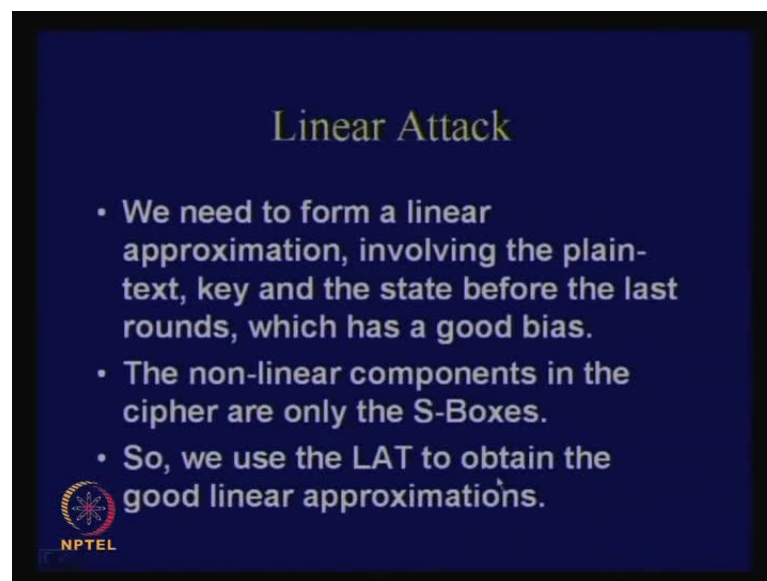
(Refer Slide Time: 44:48)



So then, you can apply the similar theory as this only replace a by b, and still have, still have got the same result, right? Right? Therefore, how many choices of possible a's are there? There are sixteen choices. So, again, in half of the cases, this equation will be satisfied, and half of the cases, this will not be satisfied, because the moment you have fix the value of x, your y gets fixed, right? So, either that  $\sum b_i$  or rather ex-or  $b_i$  will be equal 1 or will be equal to 0. So, half of the cases, this is satisfied, and half of the cases, it is not satisfied, right? And therefore, you will find that, so, half of the cases means 8, and since there are fifteen non-zero values. So, you have got 15 multiplied with 8 and thus works out to 120, right? So, therefore, you will find that if you add up, your sum is either equal to 120 or it is equal to 136. That you can check.

So, your sum can be either equal to 120 or will be equal to 136. So, this result can also be generalized, but I just showed it, because with the example, things becomes easier. If it is, if you are not really convinced, may be you can go back and look into this.

(Refer Slide Time: 45:23)



The slide has a dark blue background with a black border. The title 'Linear Attack' is centered at the top in a yellow font. Below the title are three bullet points in white text. In the bottom left corner, there is a circular logo with a red and blue design, and the text 'NPTEL' below it.

### Linear Attack

- We need to form a linear approximation, involving the plain-text, key and the state before the last rounds, which has a good bias.
- The non-linear components in the cipher are only the S-Boxes.
- So, we use the LAT to obtain the good linear approximations.

NPTEL

So, coming back to our linear attack, so there are lot of interesting properties of these table. So, what we need to do is that we essentially need to form a linear approximation involving the plaintext key and the state before the last rounds which has got a good bias. So, that is our objective, and non-linear components in the ciphers are only the s boxes, so, we require to do the l a t table or the linear approximation table to obtain the good linear, [approx/approximation] good linear approximations for the s boxes.


(Refer Slide Time: 45:52)

### Linear Approximations of the 3(=4-1) round Cipher

- Approximations of the S-Boxes with high values:

- In  $S_1^2$ , the random variable  $T_1 = U_8^1 \oplus U_7^1 \oplus U_8^1 \oplus V_8^1$  has bias  $1/4$
- In  $S_2^2$ , the random variable  $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$  has bias  $-1/4$
- In  $S_3^3$ , the random variable  $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$  has bias  $-1/4$
- In  $S_4^3$ , the random variable  $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$  has bias  $-1/4$

- If we assume that the 4 random variables are independent we can combine them by the Piling Up Lemma.



(Refer Slide Time: 45:54)

### Linear Approximation Table (LAT)

a	b															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	8	8
F	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8	8

for  $X^3 \wedge X^4 \wedge Y^1 \wedge Y^4$


$a=(0011)=3$

$b=(1001)=9$

Thus  $T[3,9]=2$

Bias =  $2/16 - 1/2 = -3/8$

Thus Bias =  $(T[a,b]/16) - 1/2$



(Refer Slide Time: 46:06)

Linear Approximations of the 3(=4-1) round Cipher

- Approximations of the S-Boxes with high values:
  - In  $S_1^1$ , the random variable  $T_1 = U_8^1 \oplus U_7^1 \oplus U_8^1 \oplus V_8^1$  has bias  $1/4$
  - In  $S_2^2$ , the random variable  $T_2 = U_8^2 \oplus V_8^2 \oplus V_8^2$  has bias  $-1/4$
  - In  $S_3^3$ , the random variable  $T_3 = U_8^3 \oplus V_8^3 \oplus V_8^3$  has bias  $-1/4$
  - In  $S_4^3$ , the random variable  $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$  has bias  $-1/4$
- If we assume that the 4 random variables are independent we can combine them by the Piling Up Lemma.

NPTEL

So therefore, what we do is that essentially from these table, if you find out those choices of a's and b's, for which, the bias value is large, and we keep those values of a's and b's. So,so, what we need to do is that we have got a four round cipher. So, we need we require to compute or rather find out the, [app/approximate] I mean a linear approximation for three round of the cipher, ok?

So, what we do is that we take certain linear approximations like this. So, these are some examples which I already know, but in order to do a real life attack, you have to really find out these approximations, ok?

So, approximations of the s boxes with high values are needed. So, you see that these are the various s boxes and these are the some, these are some linear expressions involving the inputs and the outputs of your s boxes, ok?

So, note that the variable u is, [deno/denoted] is suppose[d]-, is supposed to denote the input of the s box, and v is supposed to denote the output of the s box, and here, u one five - how we can read this or we can read this as follows: this is the first round and I am considering the fifth bit.


Similarly, u one seven means seven bit and first round. So, this is an ex-or and I am interested whether this value holds or not. Therefore, we see that there are bias values attributed 1 by 4 minus 1 by 4 minus 1 by 4 and minus 1 by 4. These are obtained from

the 1 a t table. So, if we assume that the four rounds variables are independent, we can combine them by the piling up lemma.

(Refer Slide Time: 47:31)

**Linear Approx (contd.)**


- So, the bias of:  $T_1 \oplus T_2 \oplus T_3 \oplus T_4$  is  $2^3(1/4)(-1/4)^3 = -1/32$
- This is by Piling Up lemma
- $T_1, T_2, T_3$  and  $T_4$  have the property that their input and output are expressible in terms of Plaintext, the key bits and  $u^4$  (the input to the last round of S-Boxes)



(Refer Slide Time: 47:41)

**Linear Approximations of the 3(=4-1) round Cipher**

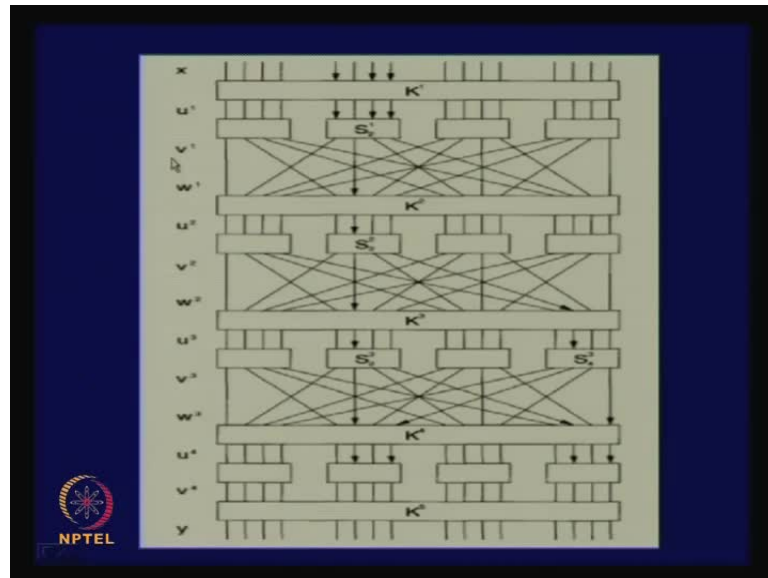
- **Approximations of the S-Boxes with high values:**
  - In  $S_1^1$ , the random variable  $T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$  has bias  $1/4$
  - In  $S_2^2$ , the random variable  $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$  has bias  $-1/4$
  - In  $S_2^3$ , the random variable  $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$  has bias  $-1/4$
  - In  $S_1^3$ , the random variable  $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$  has bias  $-1/4$
- If we assume that the 4 random variables are independent we can combine them by the Piling Up Lemma.



So therefore, what we can start doing is that we can start combining them by the piling up lemma, and we will see that **[suppo/suppose]** suppose I need to find out the bias of  $t_1$  ex-or  $t_2$  ex-or  $t_3$  ex-or  $t_4$ , so  $t_1$  and  $t_2$  and  $t_3$  and  $t_4$  - where the four individual random variables, and I need to find out the ex-or the bias of  $t_1$  ex-or with  $t_2$  ex-or  $t_3$  ex-or  $t_4$ .

So, you see that this is the straight forward example of or application of the piling up lemma. If we assume that the  $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$ , all of them are independent. So, you that this involves the first round; this involves the second round; this involves the third round, and this also involves the third round. (Refer Slide Time: 48:02)

(Refer Slide Time: 48:14)



So therefore, so, we, we often call this as a linear trail. It is refers to as a linear trail. You can observe from this diagram that this is a linear trail. So, you see, find out observe those, those arrows, you observe there are some arrows.


So therefore, you see that these this is what I am bothered with. So, I am considering the approximation of this s box; approximation of this s box; approximation of this s box, and similarly, I am finding out the approximation of this s box also, why? Because the output of this is affecting the input of this s box.

So, I am finding out the active s boxes, those x boxes which are disturbed by my initial disturbance, right? So therefore, I, I, observe these s boxes and I find out their corresponding approximations. So, imagine that approximation of this is  $t_1$ ; approximation of this is  $t_2$ ; approximation of this is  $t_3$ , and approximation of this is  $t_4$ , and each approximation involves the input of the s box and the output of the s box, right? The input of the s box is been denoted by the variable u and the output of the s box is been denoted by the variable v, similarly, for the all other s boxes also.

(Refer Slide Time: 49:22)

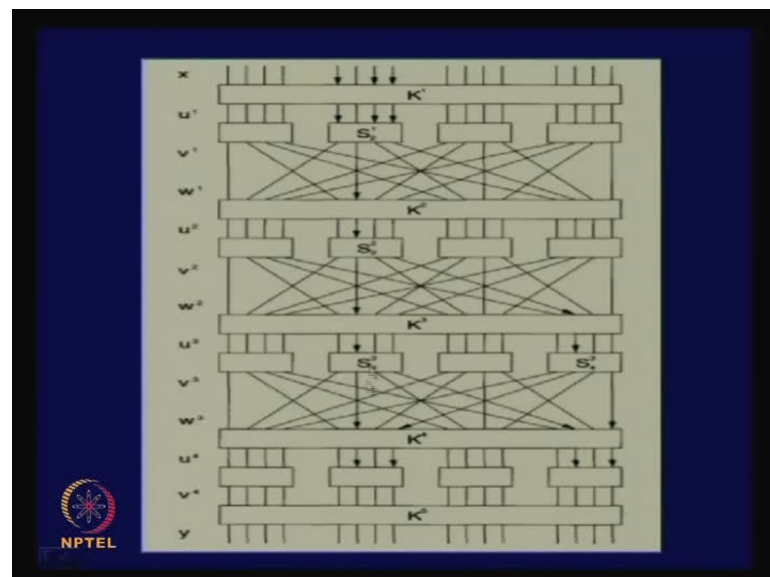
### Linear Approx (contd.)

- So, the bias of:  
$$T_1 \oplus T_2 \oplus T_3 \oplus T_4$$
is  $2^3(1/4)(-1/4)^3 = -1/32$
- This is by Piling Up lemma
- $T_1, T_2, T_3$  and  $T_4$  have the property that their input and output are expressible in terms of Plaintext, the key bits and  $u^4$  (the input to the last round of S-Boxes)



So, if we combine the ex-or, I mean obtain the s box, then we will see that applying the piling up lemma. We obtain a bias of minus 1 by 32, and  $t_1, t_2, t_3$  and  $t_4$  have the property that the input and output are expressible in terms of plaintext, the key bits and  $u^4$  which is the input to the last round.

(Refer Slide Time: 49:53)



So, you see that the variables that are there in  $t_1, t_2, t_3$  and  $t_4$ . I can express this them in terms of the plaintext, in terms of the key bits, and in terms of  $u^4$  which is the input to the last round of the s box. So, I am in, what am I interested in? I am interested in finding



out good linear approximations which involve the plaintext, the key bits and the output of the ultimate round of the input to the last round. So, therefore, this value, this variable  $u_4$ . So, I am interested in forming approximations which involve the plaintext the key bits and the  $u$  four values.

So, you see that we have come till this point and I can apply the key mixing techniques and I can express these variables as a function of your key and the plaintext, and similarly, I can do it for this also. I can express this, as a, as a linear [appro/approximation] linear ex-oring between the between the variable  $u$  four and the corresponding key  $k$  four.

(Refer Slide Time: 50:41)

**Linear Approx (contd.)**

- In  $S_1^1$ , the random variable  $T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$  has bias  $1/4$
- In  $S_2^2$ , the random variable  $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$  has bias  $-1/4$
- In  $S_2^3$ , the random variable  $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$  has bias  $-1/4$
- In  $S_4^3$ , the random variable  $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$  has bias  $-1/4$

↓

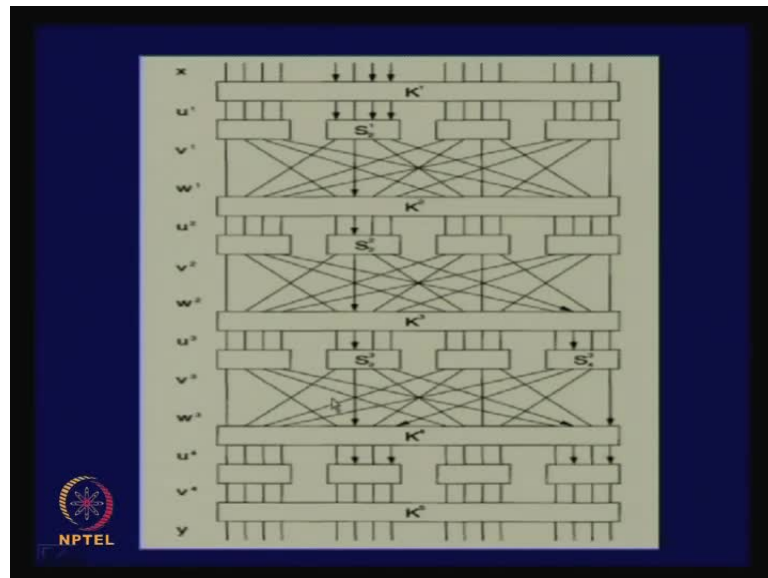
$T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$	$= X_5 \oplus K_5^1 \oplus X_7 \oplus K_7^1 \oplus X_8 \oplus K_8^1 \oplus V_6^1$
$T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$	$= V_6^1 \oplus K_6^2 \oplus V_6^2 \oplus V_8^2$
$T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$	$= V_6^2 \oplus K_6^3 \oplus V_6^3 \oplus V_8^3$
$T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$	$= V_8^2 \oplus K_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$

NPTEL

So therefore, what we do is this, that is, so these are your previous, I have just copied the previous thing -  $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$ , but I can express  $t_1$  in this form;  $t_2$  in this form;  $t_3$  in this form, and  $t_4$  in this form.

So, you see that  $t_1$  is just a copying of this, I have copied this here, and I can actually express  $u_{51}$  as  $x_5$  ex-ored with  $k_{51}$ , because  $u_{51}$  was the ex-oring of the, I can obtain  $u_{51}$  by ex-oring your input. The fifth bit of your plaintext with with the fifth bit of your first round key.

(Refer Slide Time: 51:38)



Similarly, I can do it for the other things also, and for  $t=2$  also what I do is that I take  $u_6 \oplus 2$ , so I can easily do that by  $v_{16} \oplus 1$  ex-ored with  $k_{16} \oplus 2 \oplus 1$ . You can see this from the corresponding diagram. So, actually will find that there are certain other, you, you need to take care of this permutations also, so therefore, you have to be careful.

(Refer Slide Time: 51:47)

### Linear Approx (contd.)

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4 =$$

$$X_5 \oplus X_7 \oplus X_8 \oplus V_6^3 \oplus V_8^3 \oplus V_{14}^3 \oplus V_{16}^3$$

$$\oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_8^3 \oplus K_{14}^3$$

has a bias of  $-1/32$ .

The following equations are substituted in the above equation:

$$V_6^3 = U_6^4 \oplus K_6^4$$

$$V_8^3 = U_{14}^4 \oplus K_{14}^4$$

$$V_{14}^3 = U_8^4 \oplus K_8^4$$

$$V_{16}^3 = U_{16}^4 \oplus K_{16}^4$$

So, finally, if you take this ex-ors like  $t=1$  ex-or  $t=2$  ex-or  $t=3$  ex-or  $t=4$ , you will find that you obtain an expression of these forms. You see that your expression involves now the plaintexts the corresponding values of  $v$ 's and the key values.

So, you, again you, since I need to express in terms of the variable u four, so, I will express again this v six values in terms of the u u 4 values, the, rather the v 3 values, in terms of the u four values. The u, I, I am expressing my random variable v 3 in terms of u 4.

(Refer Slide Time: 52:31)

**Linear Approx (contd.)**

- Note that the final expression involves the plaintext, key bits and  $u^4$ :

$$X_5 \oplus X_7 \oplus X_8 \oplus U_0^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4$$


$$\oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_0^2 \oplus K_0^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$$

- Note that the bias of the expression is  $1/32$ .
- Also note that the term,

$$K_1^1 \oplus K_1^2 \oplus K_5^2 \oplus K_2^3 \oplus K_3^3 \oplus K_{14}^3 \oplus K_2^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$$

can either be 1 or 0.

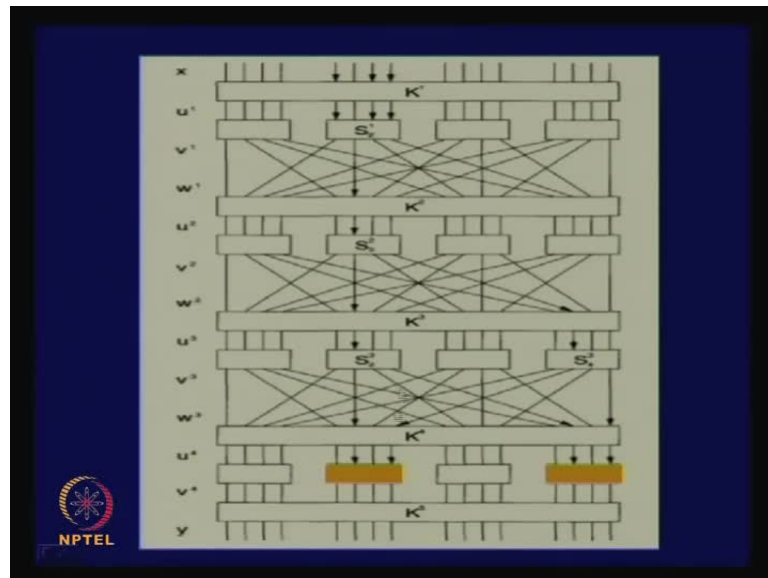
Hence the bias of  $X_5 \oplus X_7 \oplus X_8 \oplus U_0^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4$  is  $\pm 1/32$



So, I can do that and therefore, since I just substitute some linear [va/values] expressions, then your bias values equal to minus 1 by 32. So, you can observe that the final expression that will essentially involve the plaintext key bits and u 4. So, that was my objective, and the corresponding bias value of this is was equal to 1 by 32.

So, note that there are, there is now inside this expression. There is one term or rather one part of the term which involves only the key bits, and what can the key bit be equal to? It can be either equal to 0 or 1, because I have, I know that the key is fixed for all the various plaintext and cipher text pairs that I have obtained, the key was fixed. Therefore, this particular ex-or will either be equal to 0 or 1.

(Refer Slide Time: 53:22)



So therefore, the bias of the expression apart from the key bits will either be equal to plus 1 by 32 or minus 1 by 32. So, so therefore, we can obtain expressions of these form and we can see. So therefore, what we have done is that we have obtain this approximations.

(Refer Slide Time: 53:32)

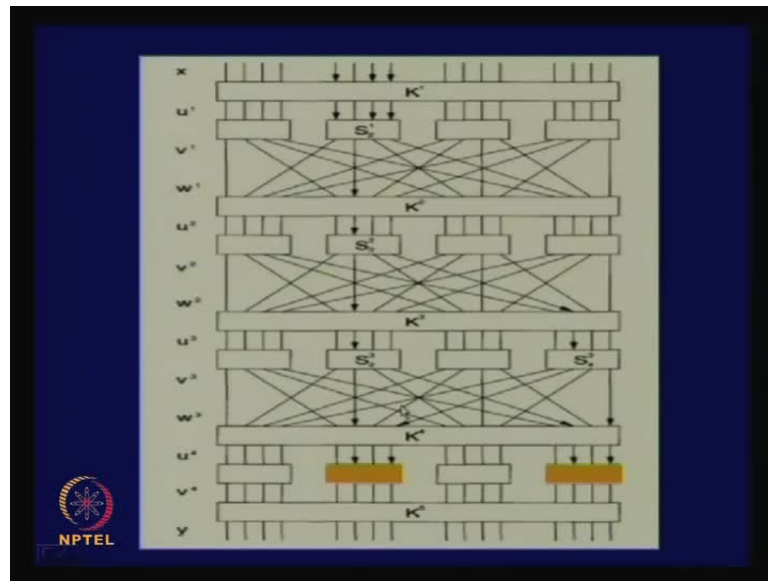
### The Attack

- Note that the expression has bits in  $U^4$ , which are there in the second and fourth S-Box of the last round.
- The attacker obtain large number of ciphertexts from the plaintexts he knows.
- Then he guesses 8 key bits,  $K_5[5-8]$ ,  $K_5[13-16]$
- He makes a frequency table, where for each key a count is stored to denote the number of cases the above expression is satisfied.
- If we inspect  $T$  plaintext, ciphertext pairs then for a wrong guess in  $T/2$  cases the expression will be satisfied.

For a correct guess, in case of about  $T/2 \pm T/32$ , the expression is satisfied.

Roughly,  $T=8000$ .

(Refer Slide Time: 53:35)

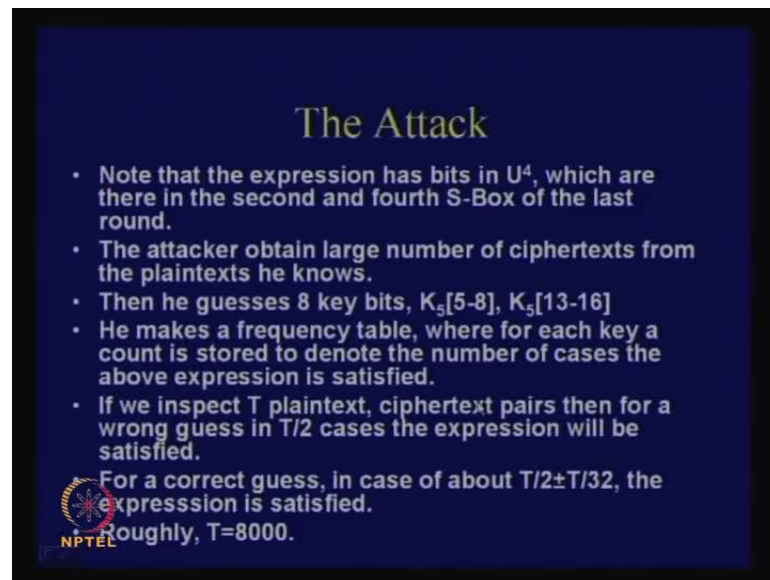


So, now what we do is that the final step of the attack is this, that is, what we do is that since we have obtained these approximations, you see that the bits that are there in your expression from the 4 variables lie in this  $S$  box and this  $S$  box.

So, in order to see whether your key guess is correct, you require to guess only this part of the key and this part of the key, not the entire key part. So, if I require the entire key guessing, my complexity would have been  $2^{32}$ . That is the brute force such complexity.

But in this case, what we do is that I just guess these 4 bits, and these 4 bits, that is,  $2^4$  complexity. The workload is  $2^4$ , and then, I, [obtain] I know the cipher text, I go back and I go to this point and check whether the corresponding expressions are being satisfied or not.


(Refer Slide Time: 54:23)



### The Attack

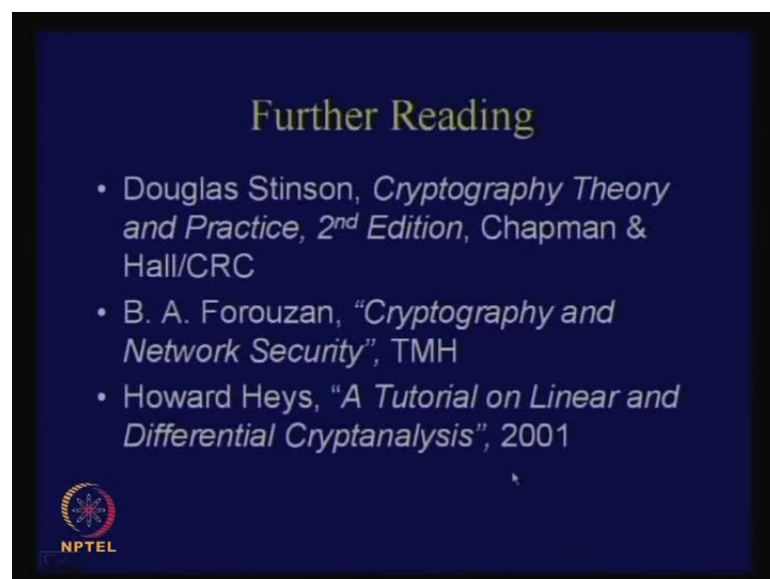
- Note that the expression has bits in  $U^4$ , which are there in the second and fourth S-Box of the last round.
- The attacker obtain large number of ciphertexts from the plaintexts he knows.
- Then he guesses 8 key bits,  $K_5[5-8]$ ,  $K_5[13-16]$
- He makes a frequency table, where for each key a count is stored to denote the number of cases the above expression is satisfied.
- If we inspect  $T$  plaintext, ciphertext pairs then for a wrong guess in  $T/2$  cases the expression will be satisfied.

For a correct guess, in case of about  $T/2 \pm T/32$ , the expression is satisfied.  
Roughly,  $T=8000$ .




So that I store in a form of a table and I keep on doing it for the large number of cases. The idea is that if my key guess is wrong, then half of the times expression will be satisfied and half of the times it will not be satisfied, but for the correct guess, we will find that almost always that expression holds, and therefore, you can form a distinguisher from a random guess and a correct guess, and in this case, we have seen that if I just try for say eight thousand cases, the actual key comes out.

(Refer Slide Time: 55:07)



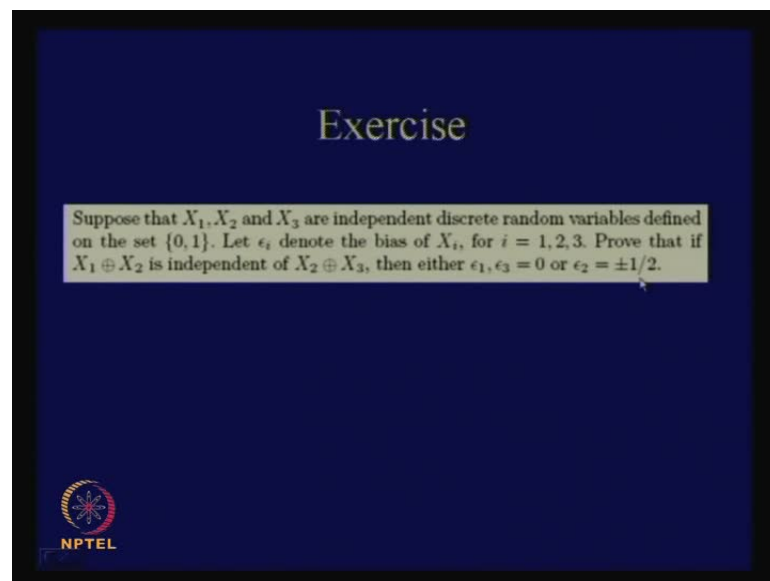
### Further Reading

- Douglas Stinson, *Cryptography Theory and Practice, 2<sup>nd</sup> Edition*, Chapman & Hall/CRC
- B. A. Forouzan, "*Cryptography and Network Security*", TMH
- Howard Heys, "*A Tutorial on Linear and Differential Cryptanalysis*", 2001



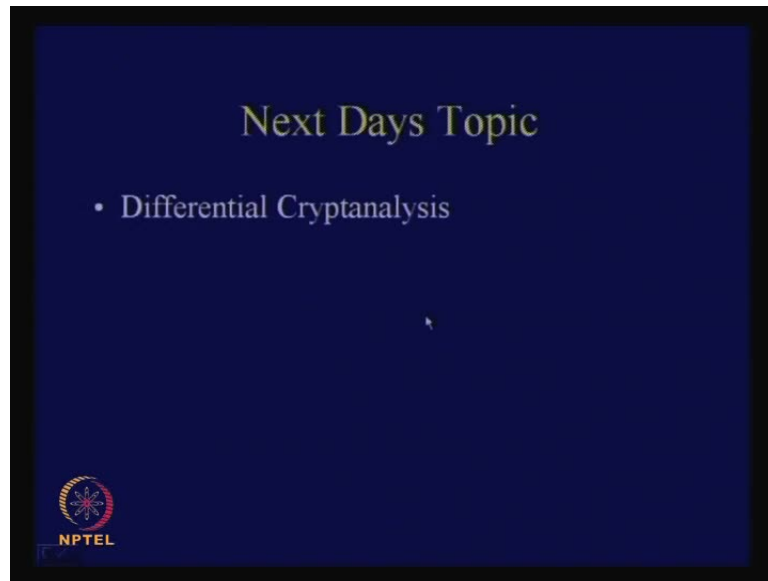
So, you see that we are actually stimulated this also, it works in real life. So, we will find that using eight thousand guesses and with the complexity of two power sixteen, you can actually break the cipher. So, the references that are followed are as again the Stinson's book, Forouzan book, but there is a very good tutorial it is available online is written by Howard Heys and is called a tutorial on linear and differential cryptanalysis, and you can just Google it out and find out. So, it is available freely on net; it is written by Howard Heys.

(Refer Slide Time: 55:33)



So, you can solve this [ex/exercise] exercise it says that suppose that  $x_1, x_2$  and  $x_3$  are independent discrete random variables defined on the set  $0, 1$ . Let  $\epsilon_1$  denote the bias of  $x_i$  for  $i$  equal to  $1, 2, 3$ . Prove that if  $x_1$  ex-ored  $x_2$  is independent of  $x_2$  ex-ored  $x_3$ , then either  $\epsilon_1, \epsilon_3$  is equal to  $0$  or  $\epsilon_2$  is equal to plus minus half. So, you can take this as an exercise and solve this. This is the straight forward application of the piling up lemma.

(Refer Slide Time: 56:06)



So, next day's topic is differential cryptanalysis and we leave (()).