

Cryptography and Network Security
Prof. D. Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No. # 01
Lecture No. # 15
Differential Cryptanalysis

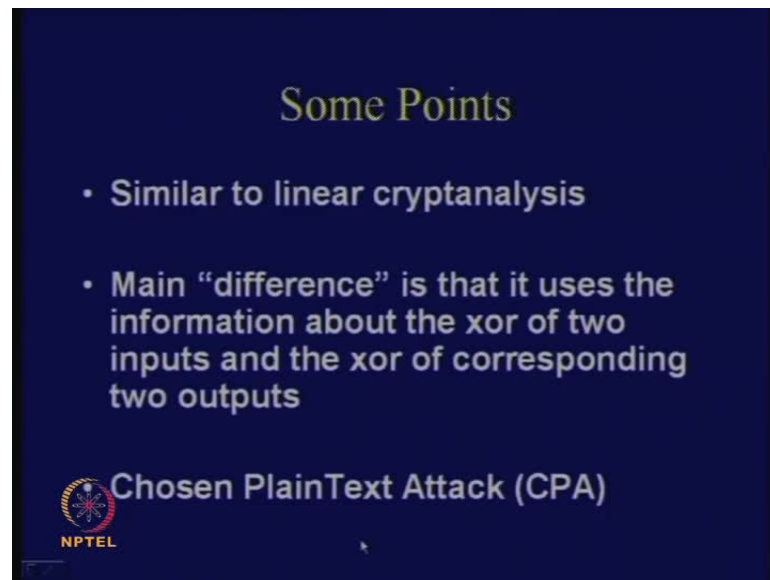
So, today as we we we were discussing about cryptanalysis, so we will continue with cryptanalysis or modern block ciphers. And last day's class we had seen in a method, which was named as linear cryptanalysis, ok, so it was supposed to find out linear approximations of ciphers, which also has non-linear components. So, in today's class, we will rather see another method, it is called as differential cryptanalysis, so it is supposedly even more powerful tool in a linear cryptanalysis.

(Refer Slide Time: 00:50)



Now, we will see, a understand what is the working principle. So, first of all we will try to understand the concept of differentials and then define something it is called a propagation ratio.

(Refer Slide Time: 01:16)



Some Points

- Similar to linear cryptanalysis
- Main “difference” is that it uses the information about the xor of two inputs and the xor of corresponding two outputs

Chosen PlainText Attack (CPA)

NPTEL

So, we will define this particular quantifier and see how to calculate this propagation ratio, ratio of a given cipher. And then discuss about the original attack, **at the** the actual attack, that is the differential attack, how do we carry forward the differential attack with given good propagation ratios. So, we will take them one by one.

Some points, this method is quite similar to the linear cryptanalysis method, the main difference, so the keyword is difference actually. So, the main difference is that it actually uses the information about the xor of two inputs and the xor of the corresponding two outputs.

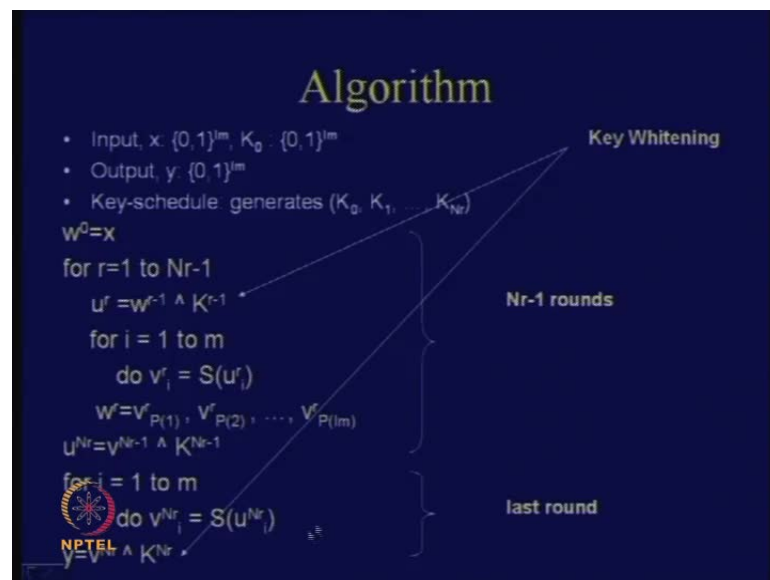
Therefore, it **uses the actual...** So, the main difference is in the difference itself, therefore you were using the difference to calculate, to find out certain properties of your given cipher. So, the xor essentially here symbolizes of difference of your inputs, **as well as...** So, you calculate **the the** the differences of the inputs and the differences of the corresponding outputs. Therefore, you take two ciphers, see, imagine there are two instance of the cipher, you feed the first instance by a given plaintext, you feed the second instance by another plaintext, so I call them p and p^* .

The only property being that p and p^* maintain a fixed difference, therefore $p \oplus p^*$ is given fixed values say a p^{dash} . And then you get corresponding output cipher text like c and c^* , therefore the idea is that this difference which you obtain in the output ok.

So, idea is that whether **it is...** So, there are some expected values of these difference and **you** basically what you do is that you guess some bit of the key and you check whether the output xor is as expected or not, so that is the broad idea of behind the attack. So, this is the chosen plaintext attack, right, not a known plaintext attack, why, because here you are obtaining plaintext which maintain a fixed difference, therefore you are not only **you know** the plaintext, but you are actually choosing a plaintext, ok.

And therefore, supposedly this is an even more powerful attack compared to the known plaintext attack, that is the linear attack, but, therefore this has got from **the designer's from from** the attacker's point of view there are stronger assumptions, right.

(Refer Slide Time: 03:22)



The assumption is that you can actually choose the plaintext, therefore, again we will go back and look at the algorithm that we cryptanalyzed last day, so we will take the same algorithm as our **(())** and continue with our discussions. So, therefore, if you remember in the last day's class we has been discussing about an s p n cipher, which **has** essentially which works on lm number of bits, right, so the block length was lm , where l and m are both integers, ok.

So, your keys also an lm bit value and your output is also an lm bit value, therefore **you** what you do is that there are nr number of rounds, the first round, I mean, the first nr minus 1 rounds, **there is there is a** what you do there, you just do key a xoring.

And then you continue doing the shift or rather, you continue doing **the** the sub bite, the **the the** S- box operation and how many S- boxes are there? There eight S- boxes or there are n S- boxes. So, what you do is that you apply the S- boxes that are **the first** the first S- box, the second S- box and so on, till the nth s box and obtain the corresponding outputs. The outputs as stored in the variable v and this particular variable is now essentially storing the output of the s box step, **so what about the**, so what about the layer? The next layer is the permutation layer, ok.

So, what you do there is that you take the l m number of bits and you do a permutation on them, right. So, **as we** as we discussed that the permutation is nothing but the transposition of the bits, **so what you**, so therefore that you can represent by this particular function called p and you see that p 1, p 2 and until p l n just represents the transposition of the bits, ok.

So, you obtain the corresponding output w and you do that **for say** for all the n r minus 1 rounds and the last round, you just do a plain substitution, therefore there is no permutation in the last round. And then finally, you again do a key xoring step, so as **as** I told in the last day's class this step and this step are known as the key whitening steps, ok.

(Refer Slide Time: 05:25)

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

So, again a recap at our original **the** cipher that we were discussing, so **we are** we are here essentially, we have got l and m, both of them are equal to 4 and n r is also equal to

4, what does it mean? It means that we are appointing on 16 bits of values, ok. And there are 4 S- boxes and the number of rounds is also 4, so this divided into, therefore, this 16 bits is divided into 4 groups and each of them are of 4 bits. So, the S- box works on each of the 4 bits, ok.

So, consider a substitution box, this is **an** an example of the S- box, so **so** therefore since S- box operates on 4 bit values, the inputs can go from 0 to 15, right, similarly the outputs also, but since we are considering a one to one S- box, that is a bijective S- box, therefore both the input and the outputs are just plain permutations of the values from 0 to 15, ok.


(Refer Slide Time: 06:25)

GPig (contd.)

- The Permutation Table is as follows:

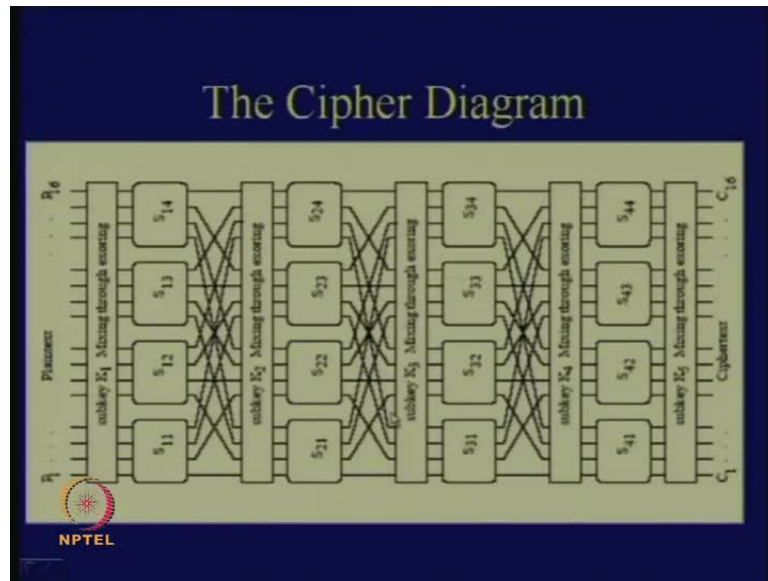
input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

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



So, it is just a map, therefore that is very simple description of S-box. I have seen **the** what about the permutation, the permutation means, that means, transposition of 16 bits, so you take the entire 16 bit block and you just transpose the bits, ok. So, therefore, **it is** this details how the transposition takes place, so if I implement, for example, in hardware, this will be just a plain hardwiring, right. So, you take some wires and just reboot them.

(Refer Slide Time: 06:49)



So, in a pictographically this was how it was looking, so it take a plain text and you obtain a cipher text and there are key xoring layers in between, which we call as a key mixing layers. And there are 16 bits, which we are divided into 4 groups and each of them have a 4 bits, right.

So, you again obtain these outputs and you pass them through 4 days boxes. So, these s boxes again, **after** after that the output that you obtain from the S- boxes, they have to be transposed, right. Therefore, this is just rewinding, which you do is a hardwire, you can hardwire this part. So, it is just transposition, you can see that the first bit comes here and the second bit goes to the one, two, three, four and five; fifth location, ok.


(Refer Slide Time: 07:30)

GPig (contd.)

- The Permutation Table is as follows:

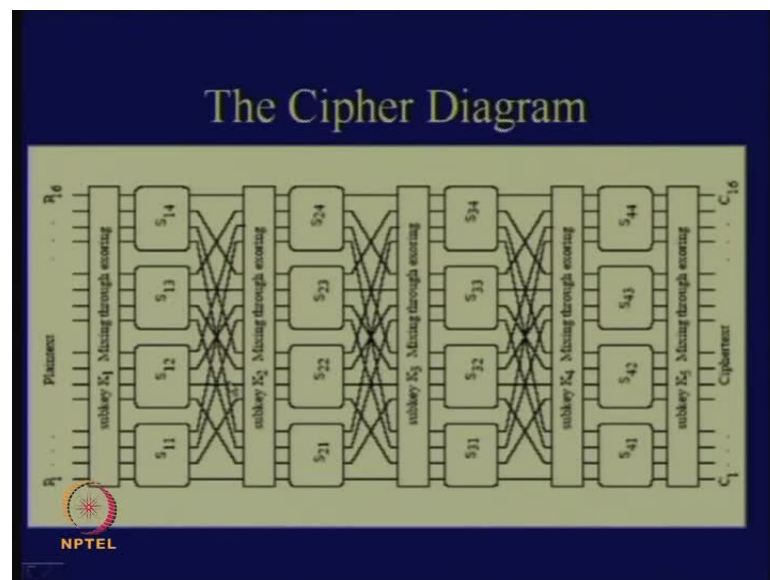
input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

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



So, you can follow that from, **the** from the permutation table also that the first bit goes to their first location and the second bit goes to the fifth location and so on. So, you can relate the diagram with this table.

(Refer Slide Time: 07:40)




So, similarly, **you obtain** there are four rounds and finally, you obtain the corresponding cipher text. So, this are all **(())** cipher which has been provided and **to the last like** last day we had performed linear attack on this, today we will see how a differential attack works on this, right.

(Refer Slide Time: 08:00)

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?



NPTEL

So, this is **again...** So, again now I am considering problem that is the g pig cipher secure, then only today I am not concern with the linear attack, so today **i am** as a designer I am evaluating whether the cipher that I am proposing is secured against a differential attack or a given chosen plaintext attack.

(Refer Slide Time: 08:22)

Key Scheduling

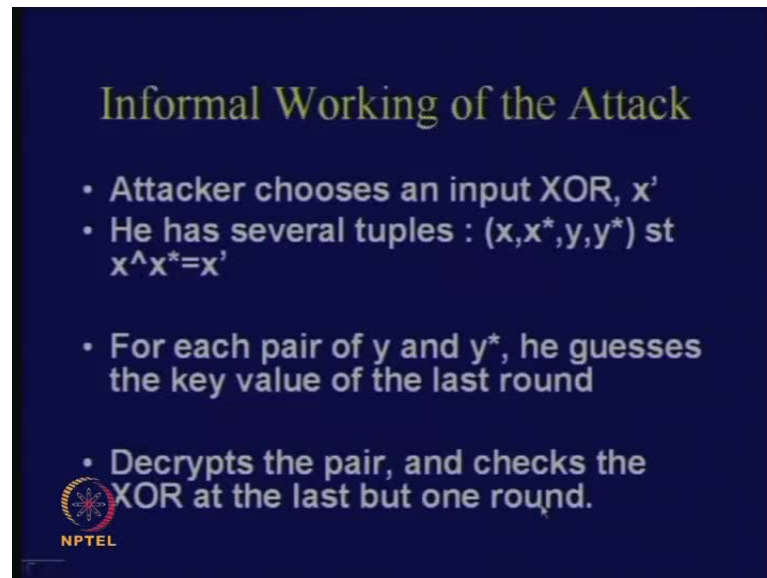
- Consider the key to be 32 bits (too small)
- A simple key schedule:
 - K_r is made by taking 16 successive bits from the key starting at $(4r + 1)$ bit position.
- Example: Input Key, K:
 - 0011 1010 1001 0100 1101 0110 0011 1111
 - $K^0 = 0011 1010 1001 0100$
 - $K^1 = 1010 1001 0100 1101$
 - $K^2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


NPTEL

So, even at the key scheduling, so the key scheduling was as follows, you take all the bits and he starts from the $4r + 1$ bit position, if r is your corresponding round number. Therefore, you start for the first bit, you start from here and you collect 16 bits, then


again you start from this point and again collect 16 bits and next round you again start from here and collect 16 bits, ok.

(Refer Slide Time: 08:48)



Informal Working of the Attack

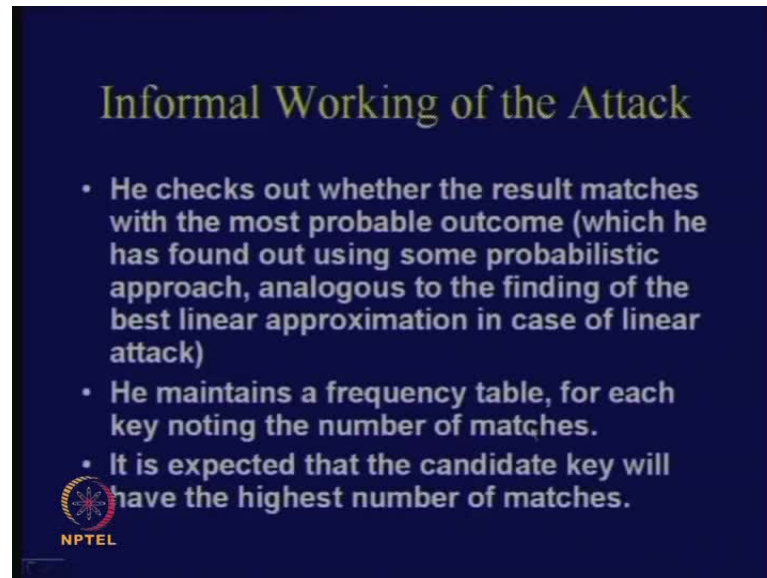
- Attacker chooses an input XOR, x'
- He has several tuples : (x, x^*, y, y^*) st $x \oplus x^* = x'$
- For each pair of y and y^* , he guesses the key value of the last round
- Decrypts the pair, and checks the XOR at the last but one round.

 NPTEL

These are the very simple substitution, I mean simple key scheduling technique, now we start formally what is known as a differential attack and we start actually informally. So, it says that the attacker chooses an input xor, x dash, therefore you fix an input xor and I name that to be x dash. So, he has several tuples, say for example, he can obtain tuples like x , x star where x and x star if you xor them, you get the value of x dash. So, you obtain the corresponding ciphers from x , that is **you obtain the output of**, if you encrypt x you obtain y and if you encrypt x star you obtain y star, ok.


So, for each pair of y and y star, he guesses the key value of the last round, so what he does is that he has obtained y and y star which are the corresponding cipher texts, so the last round he guesses some key values, ok.

(Refer Slide Time: 09:52)



Informal Working of the Attack

- He checks out whether the result matches with the most probable outcome (which he has found out using some probabilistic approach, analogous to the finding of the best linear approximation in case of linear attack)
- He maintains a frequency table, for each key noting the number of matches.
- It is expected that the candidate key will have the highest number of matches.

 NPTEL

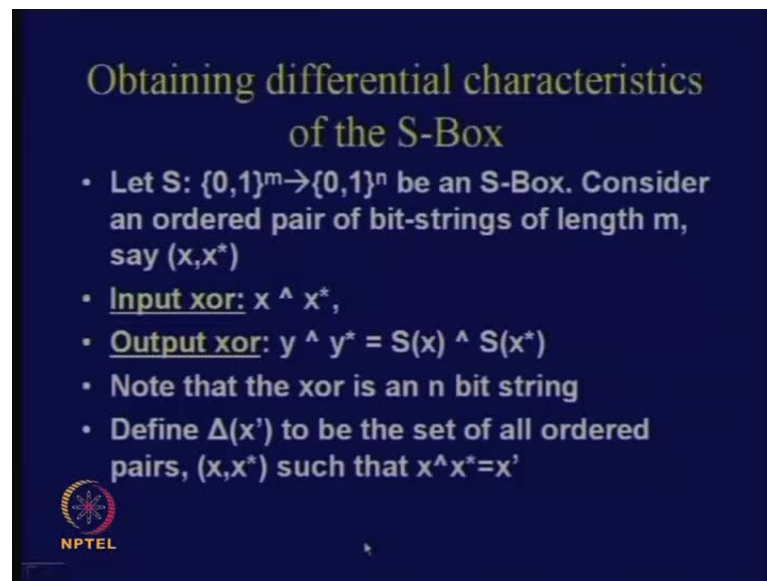
And what he does is that he decrypts the last round, so decrypts the both of the pairs and he checks the xor at the last but one round. So, he goes back and he finds out what is the corresponding xor at the **last** last but one round. And he checks out whether the result matches with the most probable outcome, **so which he has**, so which has basically found by some probabilistic technique, therefore we will discuss about the probabilistic method that is to be followed.

So, based upon this probabilistic approach, if the attacker **the the attacker** has first studied the cipher, so **and** based upon his study he has actually computed that which are the corresponding differentials which are **supposed to be** supposed to occur with the large number of probability, so this is the large probability. So, **find out**, therefore, he checks out whether the result matches with the most probable outcome, which he has found out using some probabilistic approach analogous to the finding of the best linear approximation in case of linear attack.

So, in case of linear attack, the attacker had previously studied the cipher and obtain linear approximation of the cipher, right. So, in this case, actually there is a different strategy, he has performed differential analysis of the cipher and he has obtained something which is known as more differential, I mean more probable differentials in output, therefore, so this is the, I think it will become more clear as we proceed, ok.


So, therefore, what he does is that he maintains a frequency table for each of the key noting the number of matches, therefore he finds out, so he guesses one key and finds out the number of matches. And similarly, keeps on doing that it is expected that the candidate key will have the highest number of matches, so that is the broad strategy behind that attack, ok.

(Refer Slide Time: 11:26)



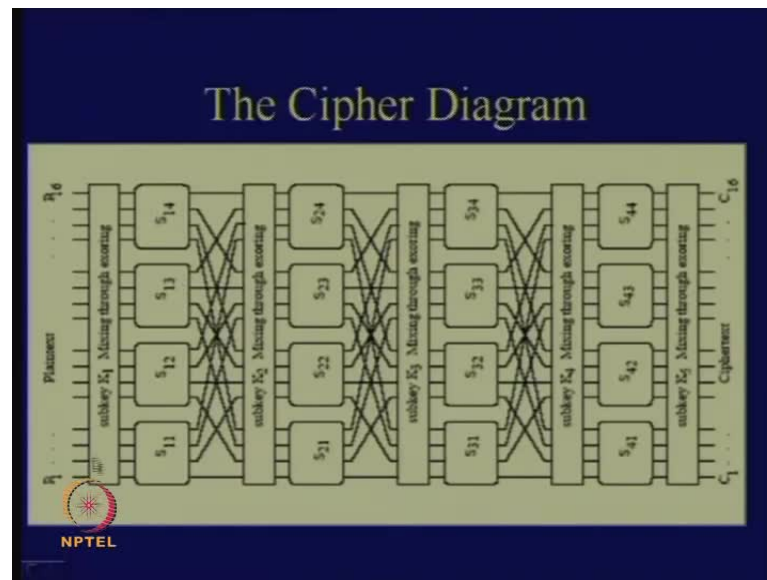
Obtaining differential characteristics of the S-Box

- Let $S: \{0,1\}^m \rightarrow \{0,1\}^n$ be an S-Box. Consider an ordered pair of bit-strings of length m , say (x, x^*)
- **Input xor:** $x \wedge x^*$,
- **Output xor:** $y \wedge y^* = S(x) \wedge S(x^*)$
- Note that the xor is an n bit string
- Define $\Delta(x')$ to be the set of all ordered pairs, (x, x^*) such that $x \wedge x^* = x'$

 NPTEL

So, now the **the** next question comes is **how** how do we really carry forward and really obtain those probability functions. Therefore, suppose **we are we** in this case we are considering three rounds of the cipher, **so** because there totally there are four rounds, so the exclude, the last round, we have three rounds, right, now we are interested in obtaining **the** the output after three rounds which is more probable, ok.

(Refer Slide Time: 11:52)



So, therefore, the idea is this that is coming back to our diagram, if we setup a differential at this point, I am interested in finding out what differential should occur here with a larger probability. Therefore I set differential input here and I am interested in finding out what is the corresponding differential here among all possible differentials, which should occur with the large number of (()) ok. So, how do I carry forward in doing this analysis? So, we have the reasonably complicated structure, right, so how do we carry forward when do that?

(Refer Slide Time: 12:28)

Obtaining differential characteristics of the S-Box

- Let $S: \{0,1\}^m \rightarrow \{0,1\}^n$ be an S-Box. Consider an ordered pair of bit-strings of length m , say (x, x^*)
- **Input xor:** $x \wedge x^*$,
- **Output xor:** $y \wedge y^* = S(x) \wedge S(x^*)$
- Note that the xor is an n bit string
- Define $\Delta(x')$ to be the set of all ordered pairs, (x, x^*) such that $x \wedge x^* = x'$

The slide also includes a small NPTEL logo in the bottom left corner.

So, **let** therefore, again your problem is mainly the S- box, why, because apart from the S- box all other layers are linear layers, ok. So, now, if I fix an input differential to a linear layer, I can easily predict what should be the corresponding differential at the output of the linear layer, but the real **(())** from the point of view the attacker is again the S- box, ok.

So, he needs to take care specially for the S- box, right. So, what he does is this, that is he takes the S- box and he does **does** more detailed analysis on the S- box, like we had done for the liner attack we also had done analysis for the S- box, ok. So, in this case also we do analysis for the S- box in terms of differentials. So, what he does is as follows. So, what you do is that he consider an ordered pair of bit strings of length m , therefore suppose your S- box operates on m bit values and results in n bit values.

So, this is an example of an m cross n S- box, so you consider an ordered pair of bit strings of length m , say I call them x , x^* and I say that the input, that is input xor is x xor with x^* . So, that is my input differential actually and what is the output differential, you compute were the value of s x , you compute the value of s x^* and you take an xor bit with them, right, therefore you obtain y and y^* and that is the output xor. So, note that the xor is an n bit string, right, because your resulting is finally an n bit values, so your xor is also an n bit value, right.

So, now let us define particular term called delta and delta is Δ , why do I call them Δ because I have fixed this value of x xor x^* supposedly, so that means that x xor with x^* is a fixed value of Δ , ok. And you obtain all such ordered pairs x , x^* , which satisfy the equation that x and x^* when taken xor results in Δ . So, it is called like a differential set, so you form a differential set or a delta set such that, **so** that is defined to be the set of all ordered pairs x , x^* such that the xor of x and x^* is equal to Δ . So, that is the very simple definition, ok.


(Refer Slide Time: 14:47)

The Delta Set

$$\Delta(x') = \{(x, x \oplus x') : x \in \{0, 1\}^m\}.$$

- Observe that the number of elements in the set is 2^m .
- For each pair in the set, the number of values which the output xor can take is 2^n .
- Thus the 2^m output pairs are distributed among 2^n values.

The non-uniformity in the distribution is exploited in the attack.



So, how do I define more? Therefore, this is how it looks like formally, right, you take delta, delta x dash, delta x dash is equal to all ordered pairs x, xor with x dash, right and x you vary from for all the possible m bit values, because that is **that is your** the entire space of your x, right. So, how many number of elements can you obtain in this fashion?

You can obtain 2 power of m values, right, **but you i** because all of us see that that there are 2 power of m values in this set, right, the coordinate of the set is 2 power m right, but how many values, how many distinct values can output take? It can take 2 power m values, ok. So, therefore, in this distribution, if I note here in a form of a distribution **or a** or a table then this 2 power m output pairs have to be distributed among 2 power n values, correct. So, the problem is that if you take a non-linear s box, then this distribution is actually not uniform, ok.

And the attacker exploits this property to mount an attack, so the idea is that **if you** if this distribution would have been a uniform distribution, then this wouldn't have worked, but then you will that **an it is** it is actually an impossible thing to obtain a fully uniform distribution if you have a non-linear transformation, ok. So, on one hand, you do not want non-linearity, but on the other hand actually non-linearity is very much required. So, you saw that in the previous attack when we talk about linear cryptanalysis, then we require non-linearity, right.

Or in this case we see that because of this distributions non-linearity or rather non uniformity I am actually mounting an attack, but this does not mean that if I replace the S- box by a linear transformation, I would have avoided this **this**, know, even then this would have worked, ok.

X dash is fixed, so what you do is that x dash is fixed and then what you do is that you start guess values, guessing values at the first **first** element in the pair, so that is x, so how many values of x are there? There are 2 power m values of x immediately when you fix x, the second values becomes fixed automatically, because x dash is fixed. That means, if you take xor between these two elements in ordered pair you obtain x dash, correct, so that is the way how this set has been generated, ok, so that I called as a delta set, right, is it clear.


(Refer Slide Time: 17:31)

An Example Set

• $\Delta(1011) = \{(0000, 1011), (0001, 1010), \dots, (1111, 0100)\}$

**Distribution of the
S-Box
output XOR
for the input
XOR = 1011**

x	x*	y	y*	y'
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010
0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

 NPTEL

Therefore, this is an example; I think it will be clear with this example. So, you considered this example, say I fix **the input** the input differential to 1011, ok. So, what does it mean, what I do is that how many possible values are there for 4 bit values, there are 16 values, right. So, what I start doing is that I start changing the first one like all zeroes, then 0001 and so on till all ones, ok.

Immediately when I fix this one the second one gets fixed, right, because your delta is same. So, how many possible elements are there in this set? There are 16 elements and

the output is also in this case 16 that means 16 output pairs have now to be distributed among 16 values, ok.

But, you observe the distribution, **you will find that this have**, so this is for your S- box, therefore **the** this for same S- box that we have described in **the** the few slides back. So, you take x , you take x^* , you operate and you operate the S- box and you obtain y and y^* and you note the differential y^Δ . So, what is y^Δ ? y^Δ is a xor of y and y^* , ok.

So, now, in an uniform distribution you would have expected that all the values would have occurred ones, right, but unfortunately it does not occur. So, we will find that there are some values which are completely absent here and there are some values which are repeated. So, there are some output differentials, you fix the input differentials to this and then there are some output differentials which are more expected, ok.

And there are some output differentials which are not expected, which are impossible, which cannot occur, so this general idea you will find appear in various kind of cryptanalytic methods, ok.

So, today we will discuss about differential attacks, where there are other classes of similar kind of attacks also, like something which is called higher order differentials, impossible differentials and so on **and also, therefore**, this idea is very central, therefore let us try to understand this properly, ok.


(Refer Slide Time: 19:29)

Non-uniform distribution of the output XORs of an S-Box

0000	0001	0010	0011	0100	0101	0110	0111
0	0	8	0	0	2	0	2
1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0	0	2	0	2

- Frequency Distribution of the Output XORs show that only 5 out of the 16 possible XORs occur
- Non-uniform distribution
- In an uniform distribution, all the output XORs would have occurred once.

This attack exploits this property, which serves as the distinguisher



So, this now, you can actually obtain a frequency table in this fashion, you will find that, so this is a simple tally table, so similar to a frequency table and you will find that there are certain differentials which have never occurred and certain differentials which occurred for a quite quite a significant number of times, ok.

So, you see that which the highest number is of rather, which is the output differential which is occurred for the maximum number of times, it is this value and so 0010 is probably the most expected. So, the frequency distribution of the output xors show that only five out of the sixteen possible xors occur, the others do not occur. So, this is a widely non uniform distribution that we have. In an uniform distribution all the output xors would have occurred exactly once, right, because we have a bijective map here. So, this as so, what. So, what if the map was not bijective, then we would have expected an uniform distribution, each of the elements would have occurred 2 to the power of m minus n number of times, if m is greater than n , right.

So, if you have a if you for example, DES, your input was 6 bits and your output was 4 bits right. So, if we had a uniform mapping, then each of the elements in a differential table would have occurred 2 to the power of 6 minus 4 , that is 2 square, that is 4 number of times, but, unfortunately that does not takes place and therefore, you can try differential attacks, ok


So, this attack exploits this property, which serves as the distinguisher, therefore this is again **you** you see that the first objective is to identify a proper distinguisher and it has been proved historically that if you have been able to find out distinguishers, you can convert that into real **real** life attacks. So, a first objective of any cryptanalysis is to find out distinguishers.

(Refer Slide Time: 21:20)

Difference Distribution Table

$\Delta x'$	$\Delta y'$															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	0	2	0	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	2	4	2	0	0	0	6	0	0	0	0	0	0	2	0
F	2	0	0	6	0	0	0	0	4	0	2	0	0	0	2	0

Any entry is denoted by $N_D(\Delta x, \Delta y)$
Thus $N_D(B, 2) = 8$

 NPTEL

So, now, you can actually do that for all possible differentials, so same thing you can do it for all possible differentials and obtain table which is known as the DDT or difference distribution table. So, you see in a difference distribution table what we have noted is in the rows, we have given the input differentials and in the output we have noted the output differentials and all these are the frequencies, right.


(Refer Slide Time: 21:51)

An Example Set

- $\Delta(1011) = \{(0000, 1011), (0001, 1010), \dots, (1111, 0100)\}$

Distribution of the S-Box output XOR for the input XOR = 1011

x	x^*	y	y^*	y'
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010
0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101




(Refer Slide Time: 22:01)

Non-uniform distribution of the output XORs of an S-Box

0000	0001	0010	0011	0100	0101	0110	0111
0	0	8	0	0	2	0	2
1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0	0	2	0	2

- Frequency Distribution of the Output XORs show that only 5 out of the 16 possible XORs occur
- Non-uniform distribution
- In an uniform distribution, all the output XORs would have occurred once.

This attack exploits this property, which serves as the distinguisher



See for example, we had worked out in the previous case of the case of b , 2, right. So, b , 2 was this, that this input differential was equal to b , right, it is b right, yeah, so what about the output differential, it is equal to 2.

(Refer Slide Time: 22:06)

Difference Distribution Table

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

Any entry is denoted by $N_{a'}(\Delta x, \Delta y)$
Thus $N_{0'}(B, 2) = 8$

NPTEL

(Refer Slide Time: 22:17)

Non-uniform distribution of the output XORs of an S-Box

0000	0001	0010	0011	0100	0101	0110	0111
0	0	8	0	0	2	0	2
1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0	0	2	0	2

- Frequency Distribution of the Output XORs show that only 5 out of the 16 possible XORs occur
- Non-uniform distribution
- In an uniform distribution, all the output XORs would have occurred once.

This attack exploits this property, which serves as the distinguisher

NPTEL

(Refer Slide Time: 22:21)

Difference Distribution Table

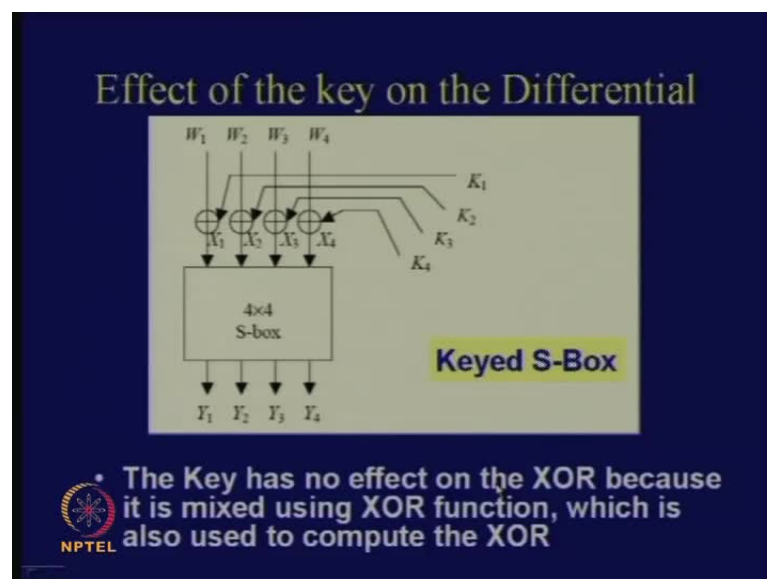
a'	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	6	0	0	2	0	0	4	0	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	2	4	2	0	0	0	6	0	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	4	0	2	0	0	0	2	0

Any entry is denoted by $N_o(\Delta x, \Delta y)$
Thus $N_o(B, 2) = 8$

NPTEL

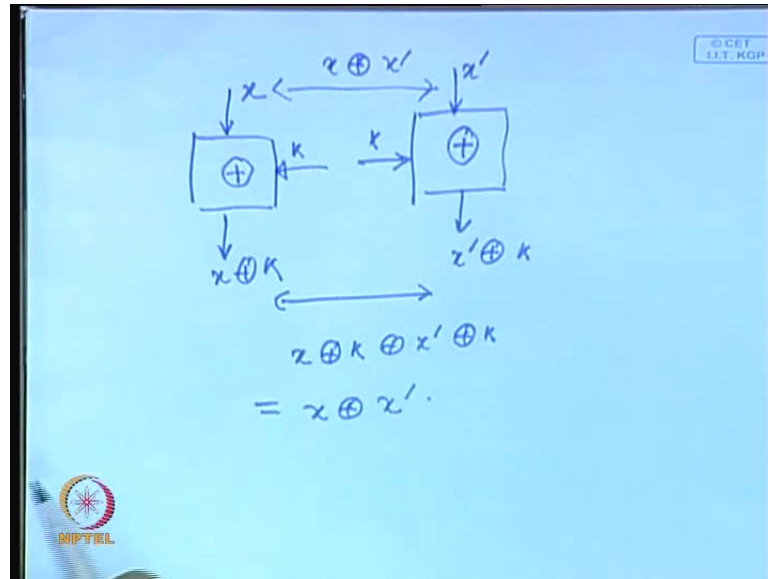
So, if I look into this table and observe the entry of b, 2, you see b and 2, you get the value 8. So, that is what we found out here, right, that is it will be stored here, similarly for the other values also you note that there are lot of zeroes, right, which means that there are lot of differentials which actually do not take place. So, if you fix the input differentials to a particular value and then you note that certain differentials which never occur and some differentials we are more expected, ok.

(Refer Slide Time: 22:47)



Similarly, you understand that **if you** if you have a table like this, you can also obtain the probabilities, right, therefore you can find the most probability, least probability and so on, right. Therefore, what you do is this, but before that let us also observe another important thing, that is the key has got more effect on the differential, why **in a** because we are mixing the key bias by symbol xor, right.

(Refer Slide Time: 23:09)



And therefore, if you take two such differentials then the key gets cancelled out, do you see that. So, what is your key xoring operation, you take x and you mix the key, right and you obtain x xor with k , so what happens when other instance of the cipher? You take x dash and you mix the same key, right. So, what do you obtain x dash xor k , so what is the differential here? x xor with k xor with x dash xor with k , I can apply my previous properties of the xor and I can obtain that this is actually equal to x xor x dash, ok.

So, this differential here was also x xor with x dash, right, therefore **if I fix my**, I mean this differential at the input actually passes to the output, so the key has got no effect, right, all of us see that. So, if the key was mixed by some other function, **so** an integer addition, then this would not have been the keys, right. But, then if we can actually change the definition of the differential and can carry out similar kind of analysis, ok.

So, my differential is exactly the defined as the just exactly the opposite of my addition, right, so, in this case, both of them are xors, because I am doing **doing** AGF, two operation, my addition and subtraction, both can be implemented using an or, ok.

(Refer Slide Time: 24:51)

Effect of the key on the Differential

Keyed S-Box

- The Key has no effect on the XOR because it is mixed using XOR function, which is also used to compute the XOR

But, if you are done an integer addition, then you would have defined the differential by the **by by** subtraction, modular subtraction. So, the key has got no effect on the differential, so a key mixing has no effect on the xor, because it is mixed using a xor function which is also used to compute the xor, ok.

(Refer Slide Time: 25:06)

Propagation Ratio

- Propagation Ratio (Prop Ratio) is the probability that an input XOR a' gives an output XOR b'
- The pair (a', b') is called a Differential
- Thus Prop^r ratio for (a', b') :

$$R_p(a', b') = \frac{N_D(a', b')}{2^m}$$

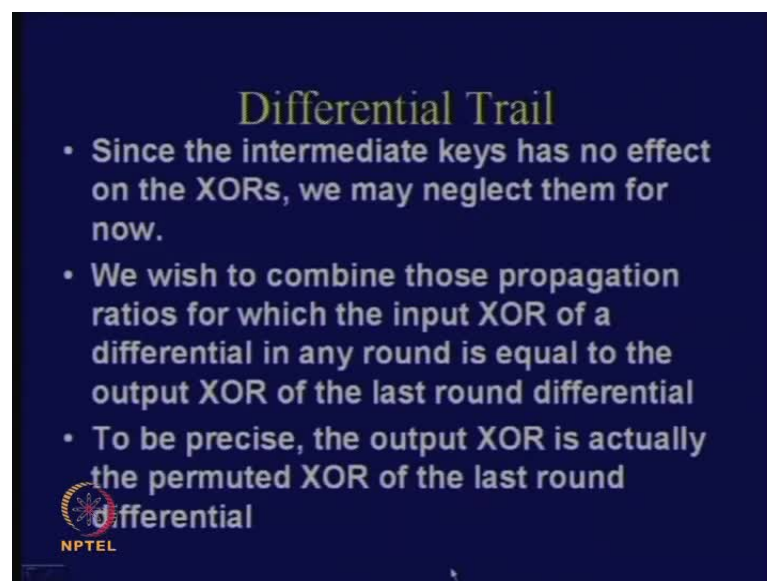
So, therefore the differential passes that is the point, now we can define something which is called a propagation ratio as follows. So, propagation ratio or prop ratio is the probability that an input xor a dash gives a corresponding output b dash, ok. So, we had

noted the table, right, so the table essentially had pairs at elements, at entries, at locations which are corresponding to the dash row and the b dash column, right. So, what does it mean? It means that if you set the input differential to a dash then the number of times b dash occurs an output differential is so on, is a n d a dash, b dash, ok.

So, now, what is the probability of the prop ratio? It is this divided by 2 power of m. So, 2 power of m are the number of possible inputs that you can provide, therefore you can obtain similar kind of probabilities and that we may refer to as the propagation ratio or the prop ratio.


So, **you observe that**, I think we have all of us **have** can say that this point can appreciate the fact that there are some values for which the propagation ratio will be actually equal to zero, because they never occur, ok.

(Refer Slide Time: 26:13)



Differential Trail

- Since the intermediate keys has no effect on the XORs, we may neglect them for now.
- We wish to combine those propagation ratios for which the input XOR of a differential in any round is equal to the output XOR of the last round differential
- To be precise, the output XOR is actually the permuted XOR of the last round differential

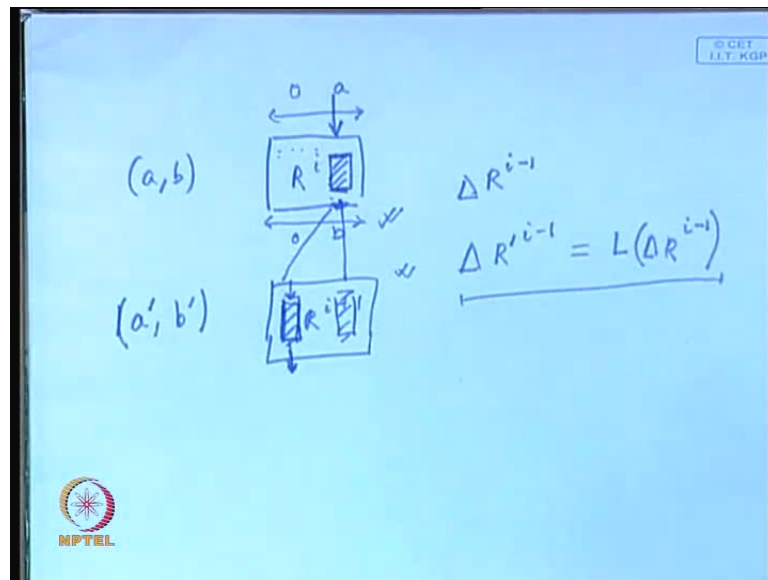
 NPTEL

So, now, if you have the idea of the propagation ratio similar to something which we have called as a linear trail in our linear attack, we can define something which is called a differential trail. So, what is the differential trail, **see you** say **the** since the intermediate keys has got no effect on the xors in our analysis, we can actually neglect the keys, right.

Because they do not have on the differential, right, so you can just simply throw away the keys and that is actually one reason differential attack is so powerful, so where the input points, the keys has got no effect, ok.

So, if you do a differential attack, the key has got no effect on your analysis that makes your analysis quite simple, right and ugly simpler. So, what we wish to combine is those propagation ratios for which the input xor of a differential in any round is equal to the output xor of the last round differential.

(Refer Slide Time: 27:10)



So, what it means is this, that is so you see that your cipher essentially composes of rounds like, so you idea is that suppose you take round R^i and you observe round R^i plus 1. So, what you have done is that may be a targeted one S-box here, ok and your targeted one another S-box here and we have found out that the propagation ratio of these and these is a high for a particular value, I call, I refer them as a dash and b dash and the propagation ratio for these S-box is high, for say a value of a, b, ok.

So, now, if I want to combine these two, that means I require to find out the propagation ratio of a and say b dash here, then that means I need to see that these actually these difference is actually influencing this differential here or is the same differential as this, only then I can combine, you see that. So, the moment I fix, say for example, say for example I need a difference here, so that means what? That means, for all the other other points I fix a zero zero difference, now that means if I denote the differential here in a word format, then this would be all zeroes followed by a, right.

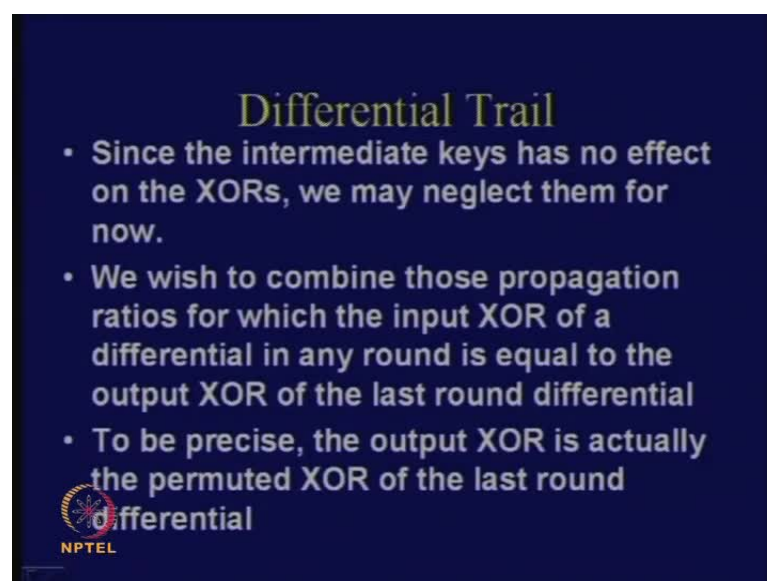
This is the input differential of this round, now I obtain by by this analysis, suppose I will say that in the output you have a large probability that say all zeroes and b will occur

as a differential. Now, here, after this what do you have, you have a transposition, right. So, you will observe that where does this **disturbed by** disturbed bits go, say for example, this disturbed bits, say one bit goes here, one bit goes here and so on, right. So, the moment what does it means that if there is one disturbed S- box, that is actually influencing not one S- box, but may be two S- boxes, right.

So, therefore, now after you have observed that, you need to find out the difference propagation ratio of this S- box as well as this S- box, right, therefore you need to see that the difference propagation which you obtained here is actually equal to the difference propagation here if there was no permutation layer. But, if there is permutation layer, it is equal to the permuted differential, so it is equal, so if your difference here is ΔI call that ΔR_i , something like this, ΔR_i or rather Δ output of R_i and **here** here it is actually $\Delta R_i - 1$.


And here, it is equal to say suppose $\Delta R_i - 1$, then **that** since you have a linear transformation here, you can actually write the $\Delta R_i - 1$ is equal to the linear function operated upon $\Delta R_i - 1$. So, mathematical I can represent this in this format, right, so I mean this is a mathematical representation, but if you do not bother about this, is just **(())** that it should be such that the difference, that when **you** in your differential analysis you take care of how that wires are getting transposed, right.

(Refer Slide Time: 30:36)



Differential Trail

- Since the intermediate keys has no effect on the XORs, we may neglect them for now.
- We wish to combine those propagation ratios for which the input XOR of a differential in any round is equal to the output XOR of the last round differential
- To be precise, the output XOR is actually the permuted XOR of the last round differential

 NPTEL

So, I think it would be, again I will repeat this point in our analysis, so we says that, now you come to this point, so we wish to combine those propagation ratios for which the input xor of a differential in any round is equal to the output xor of the last round differential, ok.

(Refer Slide Time: 31:04)

• The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

- In $S_2^1, R_p(1011, 0010) = 1/2$
- In $S_3^2, R_p(0100, 0110) = 3/8$
- In $S_2^3, R_p(0010, 0101) = 3/8$
- In $S_3^3, R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:

$$R_p(0100000000, 0000010101010000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$

To be precise the output xor is actually the permuted xor of the last round differential, so that is what I said and I will again repeat this in our example. So, the prop ratios are assumed, so another assumption that we make is that about prop ratios that is the individual prop ratios are actually independent, so this is exactly similar to the assumption behind the pilling up lemma.

So, again this comes from the fact that my key scheduling is supposed to provide the independent keys for each round and if we does so, then after this xoring, you get independent random variables. Therefore, you can assume fairly as the prop ratios are **are** independent, therefore you can actually multiply the corresponding prop ratios, ok.

So, now you see a real life example and I think example is always important in this cases. So, **you see that in your**, you see that we've targeted some s box, like may be this s box, so you observe the arrows again. So, you observe this arrow, this arrow and this arrow, so what does it mean that we have set the input differential to certain value, ok.

So, if there are no arrows, that means my differential is actually equal to zero. So, you have basically disturbed this points, so immediately if you have disturb this points, you see that this key xoring does not change the differential, so it just passes its differentials, right. Then what happens is that this s box produces some output, right, so here you can actually have lot of possible outputs, so what the attacker does is that from the table he says that if this be the input differential, then that is the larger probability of the particular differential to occurred as an output, ok.

See for example, here we have said this as a 1011, so it is expected that in the output you will have **so** 0010 as an output. So, that was a larger number that is occurring for a larger number of cases, right. So, this now you see because of the transposition, this, the disturb bit is this one, right. So, this one is a disturb bit, so where does it go, it goes to this particular point, so it actually makes this s box disturbed, right. So, what happens to the other output of this S- box, again we have set the input to say 0100 and again from the table you find out which output differential is more probable, ok.

And then you observe that so in this case for example 0110 is more probable and you find that where does this go, this bit goes here, this bit goes here, therefore now two S-boxes are disturbed, right. And similarly, you obtain the corresponding expected output differential here, so finally you see that **in** in the cipher text **this s box and this s box**, in the final round, this S- box and this S- box are getting disturbed, right.

So, now I am interested in finding out, if I said my input differential to this value and I expect that my output differential is this value, what is the corresponding probability formed? So, what I do for that is very simple, I take the individual probabilities and assuming that fact all of them are independent, I multiply the probabilities, ok.

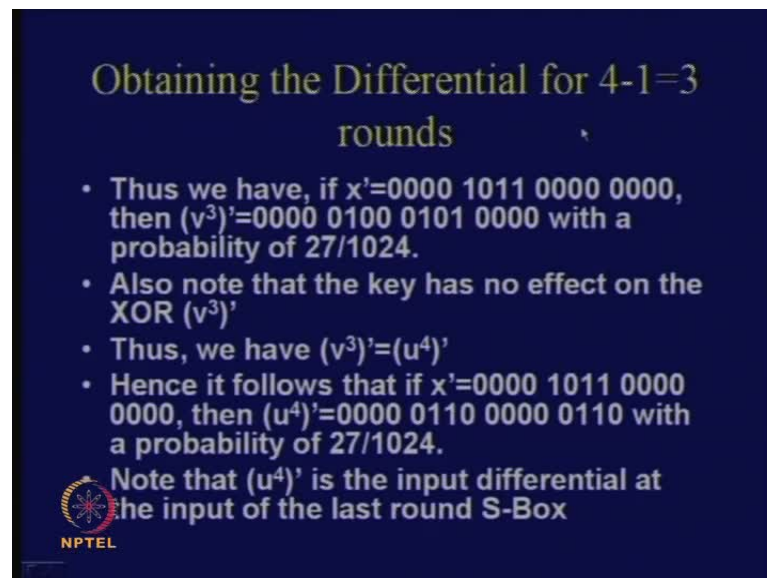
So, that is the broad idea, now let us see the little bit more details. So, you see that here if I set this value to this then your prop ratios is half, why, it was equal to 8 by 16, right, so is equal to half. Similarly, for the tables you can obtain this values also, there 3 by 3, 3 by 8 and 3 by 8, so you see that, so it is exactly the same way, as I mean you can see that why I am interested in this differential, ok. So, that is precisely, exactly the same, as I told you that you have take care of this differentials and also the transposition of the words. So, finally, you see that your output differential which you are expecting here are

the resultant prop ratio is obtained as this, your input is four zeroes followed by 1011 followed by all zeroes, ok.

And what is the differential here? It is four zeroes, 01, so **you** you observe here actually, here, so 0000, then you have got 0101 and then you have got 0101 and then you have got four zeroes, ok. So, I am finding out, if I set my input differential to these values and output differential is expected to be this value, what the corresponding propagation ratio for this bit differential is, in order to do that I multiply this probabilities, ok.

Because all of them has to hold simultaneously, right, all the events has to hold simultaneously, therefore If I assume that all these events or the all these prop ratio are independent then I can multiply them, right. So, if I multiply them then I obtain for example half into 3 by 8 whole cube and I obtain 27 divided by 1024, ok.


(Refer Slide Time: 36:13)



Obtaining the Differential for 4-1=3 rounds

- Thus we have, if $x^1=0000\ 1011\ 0000\ 0000$, then $(v^3)'=0000\ 0100\ 0101\ 0000$ with a probability of $27/1024$.
- Also note that the key has no effect on the XOR $(v^3)'$
- Thus, we have $(v^3)'=(u^4)'$
- Hence it follows that if $x^1=0000\ 1011\ 0000\ 0000$, then $(u^4)'=0000\ 0110\ 0000\ 0110$ with a probability of $27/1024$.

Note that $(u^4)'$ is the input differential at the input of the last round S-Box

 NPTEL

So, that is my probability of this particular difference occurring, right. Now, what we do is that we obtain the differential of a three round approximation of the cipher, right.

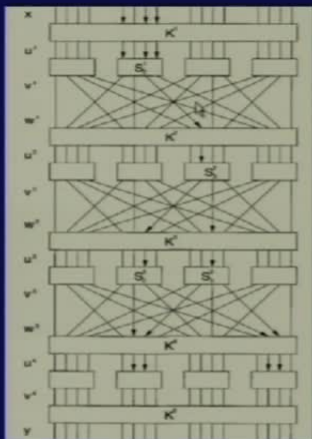
(Refer Slide Time: 36:22)

• The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

- In S_2^1 , $R_p(1011, 0010) = 1/2$
- In S_3^2 , $R_p(0100, 0110) = 3/8$
- In S_2^3 , $R_p(0010, 0101) = 3/8$
- In S_3^3 , $R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:

$$I_p(0000101100000000, 0000010101010000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$



The diagram illustrates a 4-round Feistel network. It shows four rounds of computation. Each round consists of a function block (F) and an XOR operation with a round key (K). The input differential x is shown at the top. The output differential v is shown at the bottom. The diagram shows how the differential propagates through the rounds, with the final output differential v being the result of the propagation through three S-boxes and XOR operations.

(Refer Slide Time: 36:29)

Obtaining the Differential for 4-1=3 rounds

- Thus we have, if $x'=0000\ 1011\ 0000\ 0000$, then $(v^3)'=0000\ 0100\ 0101\ 0000$ with a probability of 27/1024.
- Also note that the key has no effect on the XOR $(v^3)'$
- Thus, we have $(v^3)'=(u^4)'$
- Hence it follows that if $x'=0000\ 1011\ 0000\ 0000$, then $(u^4)'=0000\ 0110\ 0000\ 0110$ with a probability of 27/1024.

Note that $(u^4)'$ is the input differential at the input of the last round S-Box



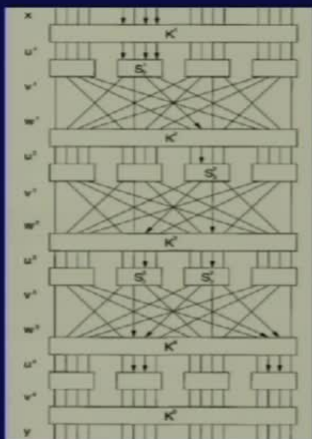
(Refer Slide Time: 36:42)

• The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

- In $S_2^1, R_p(1011, 0010) = 1/2$
- In $S_3^2, R_p(0100, 0110) = 3/8$
- In $S_2^3, R_p(0010, 0101) = 3/8$
- In $S_3^3, R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:

$$I_p(0000101100000000, 0000010101010000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$



So, is this part clear to us? In this part now we obtain a differential three rounds that is 4 minus 1, 3 rounds. So, we have got x dash is equal to this, that is four zeroes 1011 and four zeroes and therefore you see that v 3 dash.

(Refer Slide Time: 36:52)

Obtaining the Differential for 4-1=3 rounds

- Thus we have, if $x'=0000\ 1011\ 0000\ 0000$, then $(v^3)'=0000\ 0100\ 0101\ 0000$ with a probability of 27/1024.
- Also note that the key has no effect on the XOR $(v^3)'$
- Thus, we have $(v^3)'=(u^4)'$
- Hence it follows that if $x'=0000\ 1011\ 0000\ 0000$, then $(u^4)'=0000\ 0110\ 0000\ 0110$ with a probability of 27/1024.

Note that $(u^4)'$ is the input differential at the input of the last round S-Box



So, what was v 3 dash, v 3 dash was this particular location, right, therefore this is the differential at this location. So, we will see that v 3 dash was equal to this value, that is four zeroes 0100 0101 and four zeroes and this probability is actually equal to 27 by 1024, ok.

So, also note that the key has got no effect on the xor v 3 dash and thus we are actually v 3 dash is equal to u 4 dash, therefore it passes, hence it follows that if x dash, if we set the value of x dash to this, then u 4 dash is a expectation is this with the probability of 27 by 1024, ok.

(Refer Slide Time: 37:32)

• The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

- In S_1^1 , $R_p(1011, 0010) = 1/2$
- In S_2^2 , $R_p(0100, 0110) = 3/8$
- In S_2^3 , $R_p(0010, 0101) = 3/8$
- In S_3^3 , $R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:

$$R_p(0000\ 1011\ 0000\ 0000, 0000\ 0101\ 0101, 0000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$


So, note that u 4 dash is the input differential at the input of the last round s box, that is **what my** what my objective was, right. So, I have actually found out an expected differential at this location, so you see that this is equal to 0000 0110 then 0000 and again 0110, ok.

(Refer Slide Time: 37:47)

Obtaining the Differential for 4-1=3 rounds

- Thus we have, if $x' = 0000\ 1011\ 0000\ 0000$, then $(v^3)' = 0000\ 0100\ 0101\ 0000$ with a probability of $27/1024$.
- Also note that the key has no effect on the XOR $(v^3)'$.
- Thus, we have $(v^3)' = (u^4)'$.
- Hence it follows that if $x' = 0000\ 1011\ 0000\ 0000$, then $(u^4)' = 0000\ 0110\ 0000\ 0110$ with a probability of $27/1024$.

Note that $(u^4)'$ is the input differential at the input of the last round S-Box



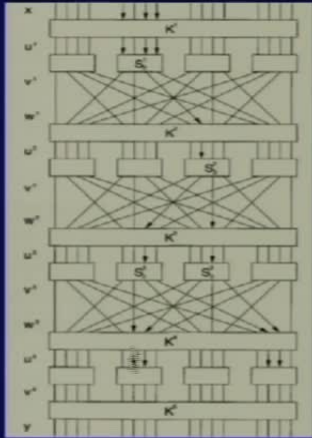
(Refer Slide Time: 37:54)


• The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

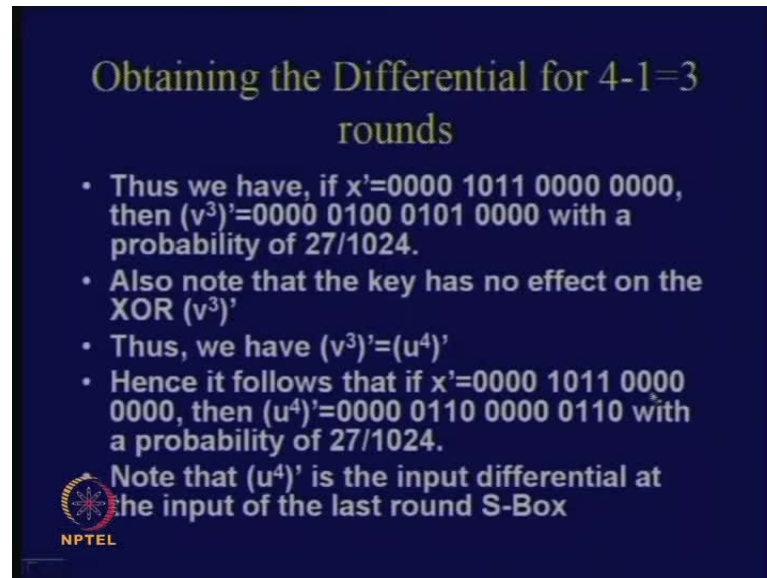
- In S_2^1 , $R_p(1011, 0010) = 1/2$
- In S_3^2 , $R_p(0100, 0110) = 3/8$
- In S_2^3 , $R_p(0010, 0101) = 3/8$
- In S_3^3 , $R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:



$$R_p(0000\ 1011\ 0000\ 0000, 0000\ 0100\ 0101\ 0000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^2 = \frac{27}{1024}$$



(Refer Slide Time: 37:57)



Obtaining the Differential for 4-1=3 rounds

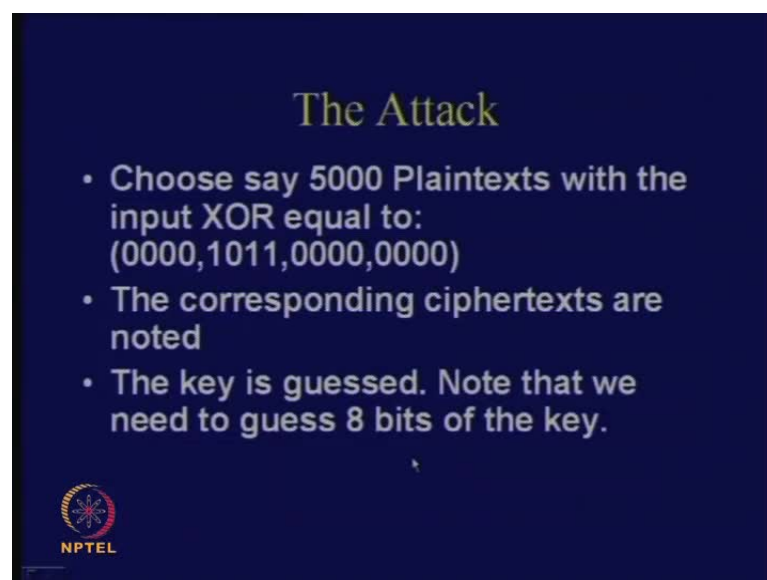
- Thus we have, if $x^1=0000\ 1011\ 0000\ 0000$, then $(v^3)^1=0000\ 0100\ 0101\ 0000$ with a probability of $27/1024$.
- Also note that the key has no effect on the XOR $(v^3)^1$
- Thus, we have $(v^3)^1=(u^4)^1$
- Hence it follows that if $x^1=0000\ 1011\ 0000\ 0000$, then $(u^4)^1=0000\ 0110\ 0000\ 0110$ with a probability of $27/1024$.

Note that $(u^4)^1$ is the input differential at the input of the last round S-Box




So, that is my corresponding output differential, 0000 0110 all the four zeroes, so in this case it is told to be at this point, so you can see that if I propagate that to the output I can obtain similarly the differential, ok, so that is called straight forward. Therefore, this is this value corresponding, you have got four zeroes 0110 four zeroes and 0110 and your probability is of this is according 27 by 1024, ok.

(Refer Slide Time: 38:17)



The Attack

- Choose say 5000 Plaintexts with the input XOR equal to:
(0000,1011,0000,0000)
- The corresponding ciphertexts are noted
- The key is guessed. Note that we need to guess 8 bits of the key.



So, now the real attack, therefore we have done our pre I mean our analysis, what we do is that suppose you take large number of possible plaintexts, so with keeping in mind that


your input xor is this, so you set this input xor to this and you take large number of possible plaintexts. So, in this case, we have taken 5000 plaintexts and then what you do is that you obtain corresponding cipher texts and you guess the last round key.

(Refer Slide Time: 38:47)

Obtaining the Differential for 4-1=3 rounds

- Thus we have, if $x'=0000\ 1011\ 0000\ 0000$, then $(v^3)'=0000\ 0100\ 0101\ 0000$ with a probability of $27/1024$.
- Also note that the key has no effect on the XOR $(v^3)'$
- Thus, we have $(v^3)'=(u^4)'$
- Hence it follows that if $x'=0000\ 1011\ 0000\ 0000$, then $(u^4)'=0000\ 0110\ 0000\ 0110$ with a probability of $27/1024$.

Note that $(u^4)'$ is the input differential at the input of the last round S-Box



(Refer Slide Time: 38:50)

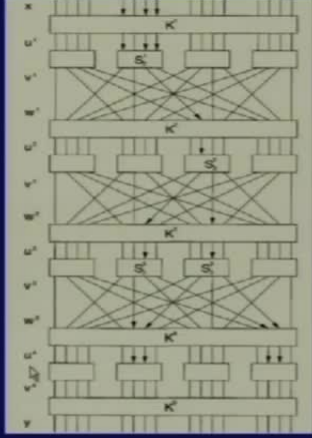

- The Prop ratios are assumed to be independent

Thus we may multiply the prop-ratios.

- In $S_2^1, R_p(1011, 0010) = 1/2$
- In $S_3^2, R_p(0100, 0110) = 3/8$
- In $S_2^3, R_p(0010, 0101) = 3/8$
- In $S_3^3, R_p(0010, 0101) = 3/8$

Thus resultant Prop-ratio is obtained as:

$$P_3(0000\ 1011\ 0000\ 0000, 0000\ 0100\ 0101\ 0000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$

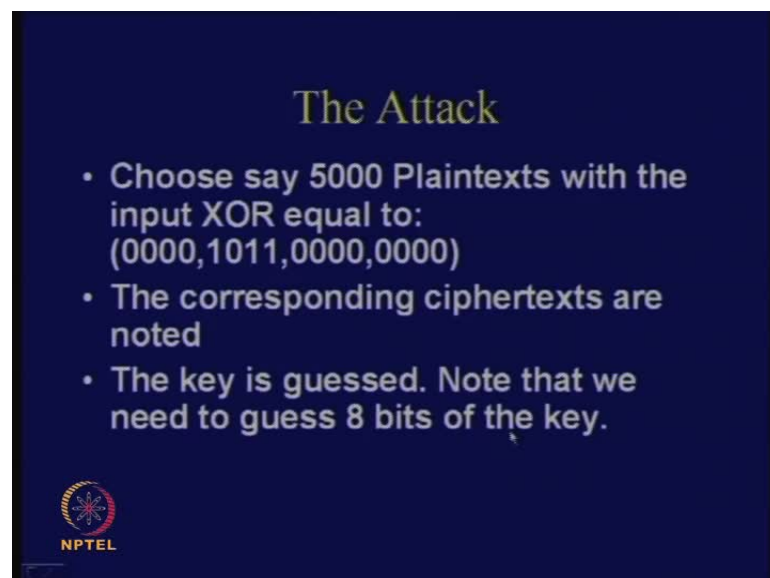



So, now you see that like a linear attack, we needed so here also **you** you do not require to guess the entire key, you require to guess only these 4 bits and these 4 bits, right, therefore in a linear attack also **we are** if you remember we are guessed only this part of the key, ok. So, **in a** if your total complexity of doing a, I mean good force complexity of

doing the attack would have been 2^{16} , right, there are 4 4 4 and 4, but in this case you are just guessing the only this part and this part, so how many possible guesses are there? 2 by that is 256 therefore you are guessing only 256 keys and what you do is that from the cipher texts you go back and you check whether the differential meets the expected value or not, ok.


And similarly you do frequency analysis for all the possible keys, it is expected that the correct key will actually give you larger number of probability, I mean will have the highest frequency in the table, ok.

(Refer Slide Time: 39:45)



The Attack

- Choose say 5000 Plaintexts with the input XOR equal to:
(0000,1011,0000,0000)
- The corresponding ciphertexts are noted
- The key is guessed. Note that we need to guess 8 bits of the key.




NPTEL

(Refer Slide Time: 39:50)

The Attack

- Decrypt the last round and verifying whether the differential at the input of the last round S-Box is 0000 0110 0000 0110
- Make a frequency table for the keys




So, what you do is that you guess the key, note that just need to guess eight bits of the key and what you do is that you decrypt the last round and verify whether the differential at the input of the last round S-box is actually equal to four zeroes 0110 four zeroes 0110, ok.

(Refer Slide Time: 40:11)

Result

<i>partial subkey</i> [$K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}$]	prob	<i>partial subkey</i> [$K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}$]	prob
1 C	0.0000	2 A	0.0032
1 D	0.0000	2 B	0.0022
1 E	0.0000	2 C	0.0000
1 F	0.0000	2 D	0.0000
2 0	0.0000	2 E	0.0000
2 1	0.0136	2 F	0.0000
2 2	0.0068	3 0	0.0004
2 3	0.0068	3 1	0.0000
2 4	0.0244	3 2	0.0004
2 5	0.0000	3 3	0.0004
2 6	0.0068	3 4	0.0000
2 7	0.0068	3 5	0.0004
2 8	0.0030	3 6	0.0000
2 9	0.0024	3 7	0.0008

From this observation we conclude 24 is the correct key, with a probability around $27/1024=0.0264$, which is close to the experimental value of 0.0244



So, now you make the frequency table for the keys, right. So, what you do is that you make a frequency table for the keys and if you make a table this is how a sample table would, **may be** look like. So, you see that we have guessed 4 4 keys here and four keys

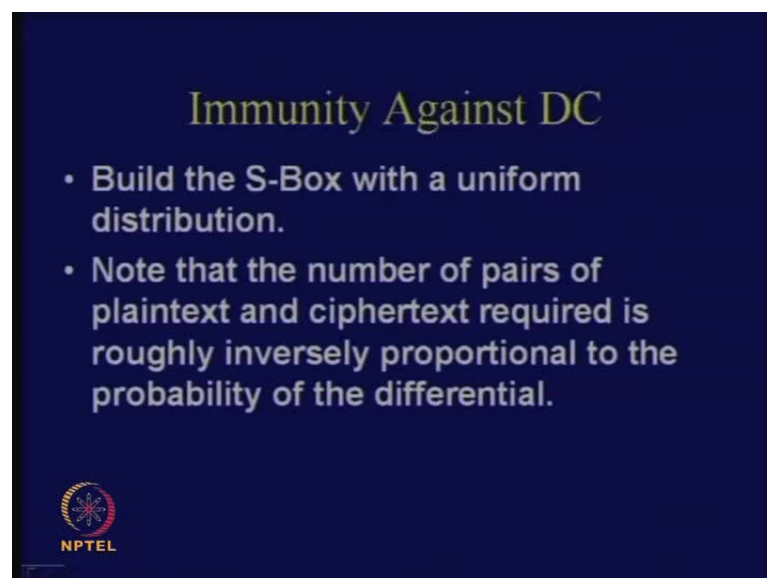
here. So, you see that 2 4 was the actual key, therefore **assume that 2 i mean** believe that 2 4 was the actual key and the probability of that occurring is actually equal to 0.0244, whereas the others has got lesser probability than that, ok.

So, from this observation we can conclude that 2 4 is the correct key with a prop ratio of around 27 by 1024, which is actually equal to 0.0264 and you see that it matches fairly with your observed value, which is close to the experimental value of 0.0244, ok. So, **you can you can** you can immediately understand that since this is a statistical analysis, doing **one one** one plaintext, two plaintext, three plaintext, it will become more and more close to your actual expected value if you take large number of plaintexts, right.

So, therefore, you have to take large number of plaintexts, so you see that in this case we have been able to do the attack with only five thousand cases, so you have to keep on doing that until or unless you find that there is significant key, there is a key which essentially, I mean goes close to the expected value, ok.


So, the idea is that you have to find out, ensure that you take large number of texts and then only you can confirm that this particular key is correct, right. So, that is the broad idea behind the differential attack, ok.

(Refer Slide Time: 41:40)



Immunity Against DC

- Build the S-Box with a uniform distribution.
- Note that the number of pairs of plaintext and ciphertext required is roughly inversely proportional to the probability of the differential.


NPTEL

So, therefore, immediately when we think of attacks, you also should think about immunity, right, because we are essentially not bad people, right, we are god people,

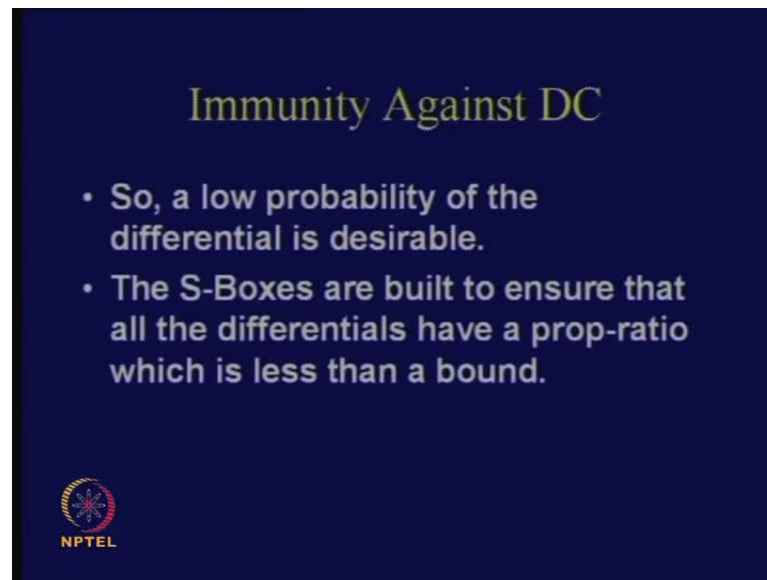
therefore we should think about how to make over cipher strong, right. So, therefore, the idea is essentially it (()) down to the fact that how do I make the s box good, right, that is the central idea. So, in this particular case when I am thinking about how to make an immune against differential cryptanalysis, I have to build the S- box with as much as uniform distribution as I can, right.

So, you can see that the I mean there are some papers which talks about uniform uniform mappings, there was a paper by neighbor, which I told you in context to linear attack- i think in contact to (()) design. So, that paper essentially talks about many such transformations and reflects upon the point of various properties which those mappings has and which can be probably used as cryptographic mappings, ok.

So, one of them was the inverse function in finite field, which was adopted by (()) designers and made and what works as the AES S- box, what works central to the AES S- box. So, note that the number of pairs of a plaintext and cipher text required is roughly inversely proportional to the probability of the differential, therefore if I observe like my probability is say p then roughly the number of pairs that we will require will proportional to one by p , ok.

So, you need, therefore you see that your probability that you occur, I mean that you obtain is high, then you require lesser number of plaintexts. Therefore, you that is the idea why you require, those prop you indicate to identify those prop ratio, ratios which has got a higher values. So, only then you will you can do the attack which smaller number of plaintexts, ok.

(Refer Slide Time: 43:38)

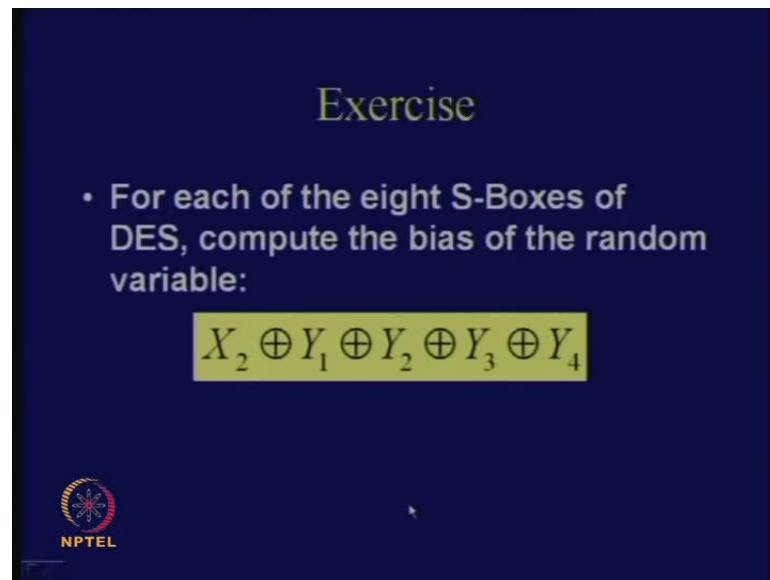


So, low probability of the differential, from the point of the attacker I will try to make the probabilities as small as possible, right, from the point from the point of the designer I will make the probability as small as possible, ok. Then the attacker will require large number of texts to carry forward the attack, right, so the s boxes are built to ensure all the differentials have a prop ratio which is less than a bound, therefore if I can provide as a designer that the prop ratios will be lesser than such values, then I am providing an upper bound to this prop ratios, ok.

So, that is required from the point of, so when I design S- boxes I require to apply commutative theory or theory, I mean mathematical principles and really provide such kind of upper bounds, ok. So, therefore, designing of designing s boxes is not so trivial, therefore you have to take care of lot of commutative and other properties and you will see that there are lot of trade off also, if you if you can save this then the other things will be exposed, ok.

So, we have got a very small piece of cloth and you have to actually cover a very big space, therefore we will see some kind of ideas about which are require to design an S-box, certain properties which we need to satisfy for when we are talking about the S- box construction in our next classes, ok.

(Refer Slide Time: 44:54)

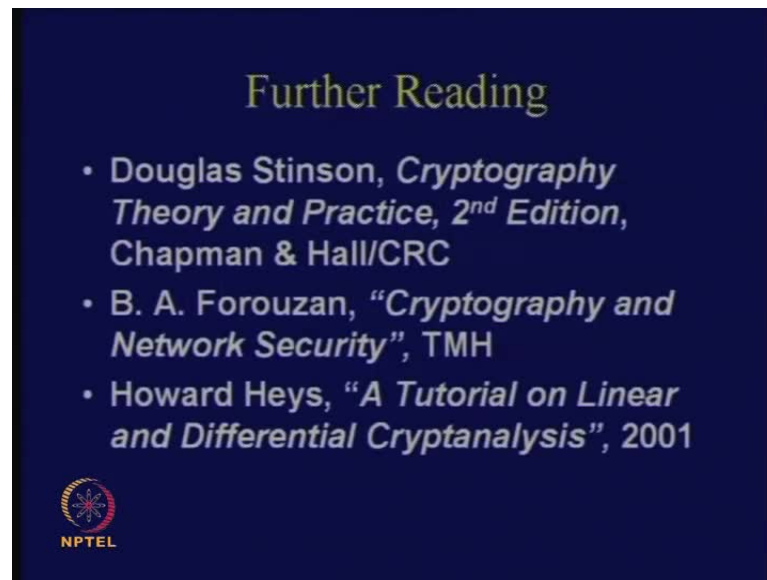


The slide has a dark blue background. At the top center, the word "Exercise" is written in a light green serif font. Below it, a white bullet point is followed by the text "For each of the eight S-Boxes of DES, compute the bias of the random variable:". Centered below this text is a light green rectangular box containing the mathematical expression $X_2 \oplus Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$. In the bottom left corner, there is a circular logo with a red and blue design and the text "NPTEL" below it.

But, for now you can just take a small exercise, so what you can do is that for each of the eight S- boxes of DES you can compute the bias of the given random variable, which is like this. So, you can do that for **say** atleast for some of the s boxes, ok, and you can you can just try to find out that whether your DES S-boxes was really good or not, therefore you can do an analysis. What you can do for this is that maybe you can write a piece of c code or an matlab code, through which you can actually evaluate the differential table of the DES S-boxes, ok.

And then **I give** I leave it to an exercise, you can actually submit reports on that also, I mean, hand written reports would do, that among the eight S- boxes which S- box do you think is the most strong, ok.

(Refer Slide Time: 45:46)



So, some of the books that I have again followed are same, so again **this** this tutorial is very good, so you can follow this tutorial, therefore this clears lot of doubts which generally appears in when you see this kind of attack for the first time, ok.

(Refer Slide Time: 46:02)



So, our next **next** day's topic we will talk about some other cryptanalytic methods, so these cryptanalytic methods means that there are large number of cryptanalysis methods which are there, but we will try to understand some of them.