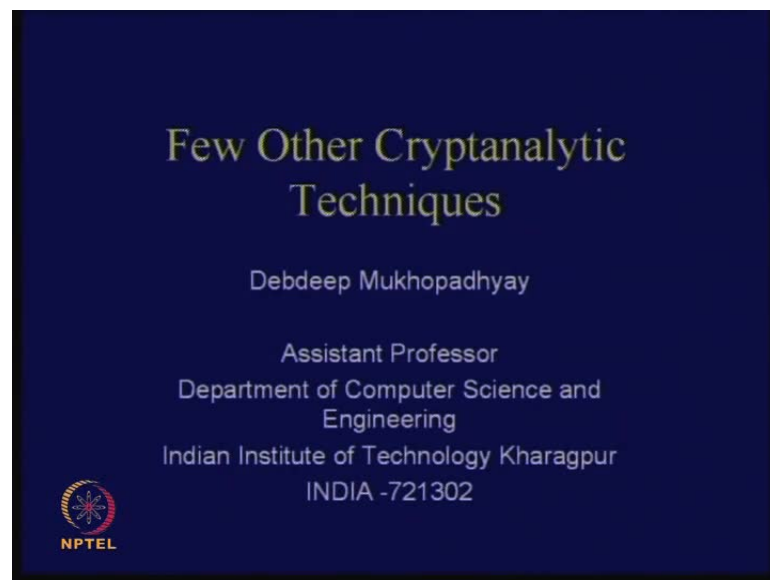


**Cryptography and Network Security**  
**Prof. D. Mukhopadhyay**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Module No: # 01**  
**Lecture No: #16**  
**Few other Cryptanalytic Techniques**

Today's class will be on some other cryptanalytic techniques or we shall be discussing about something more developed kind of attacks against block ciphers. So, we have been discussing about block - I mean, block cipher cryptanalysis. We discussed about 2 major cryptanalytic techniques: one is called linear attacks and other one is called differential cryptanalysis.

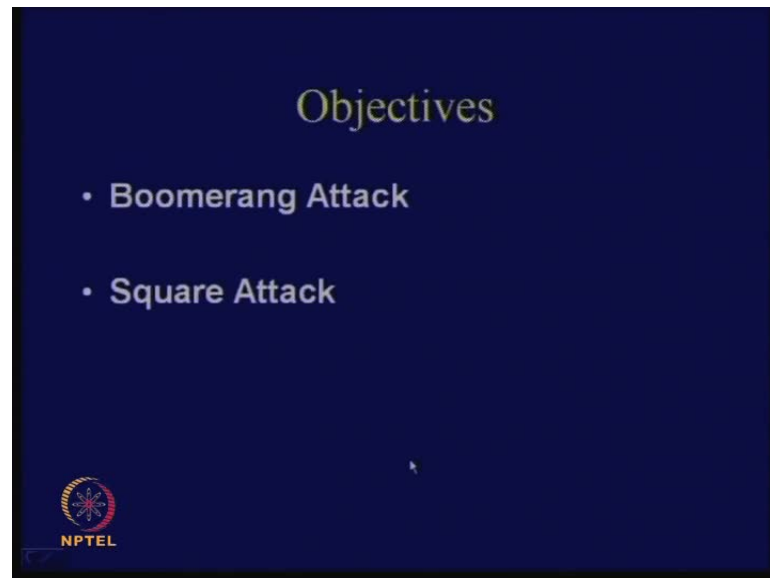
(Refer Slide Time: 00:47)



We will now see something which is more advanced and something which is more developed compared to them. **and actually some I mean it is about.** You have got a set of tools and you have to apply a tool to a given class of ciphers, right. Therefore, you do not know the cipher will succumb against which kind of attack which means that have to be knowledgeable about all possible techniques to cryptanalyze and the moment you propose a block cipher, you have to certify that the block cipher actually passes through so many such number of tests. But these tests are not very strictly formulated. Therefore, it will vary depending upon the properties of the cipher. So, we study the properties of

the cipher and know how the attack works and then we have to design or rather tailor the attack according to the given cipher.

(Refer Slide Time: 01:32)



So it is a therefore, that is the whole so in In today's class, we will discuss about 2 major classes of attacks. One is called boomerang attack and other one is called square attack. Now, why we have chosen these 2 attacks, I will come to that very shortly.

(Refer Slide Time: 01:45)



But before we go into these 2 attacks, this is an overview of some common cryptanalytic techniques. We can say that there is a whole lot of repository. We have been discussing

about linear and differential cryptanalysis. We also have differential-linear cryptanalysis and then there is something called impossible differential attacks and there is something called truncated differential attacks, higher order differential attacks, probabilistic higher order differential attacks and something which is called integral attacks.

(Refer Slide Time: 02:15)



Then there is boomerang attack, rectangle attack, slide attack, interpolation attack, square attack and we also have something, which is called side channel or fault attack. Therefore, these are not very conventional kind of cryptanalytic methods, but rather exploit the implementation weaknesses. Side channel attack falls into that class and fault attacks are those, I mean in the cipher **there is sudden fusion of or rather either due to intentional** intentionally the fault is induced or accidentally the fault can be induced. When we study the faulty cipher text and fault free cipher text, from that you try to deduce the key.

Fault is also a very important issue nowadays because the moment, you are going into smaller and smaller technologies like nano scales and things like that, faults become more obvious. So, that amount of study is also done and there is some class of attacks called correlation attacks or statistical attacks also. This is actually something like you study various kinds of properties; this is something which we have done previously also. We have seen in case of classical ciphers, we have been doing various kinds of statistical

techniques, evaluating various like the [cususky] test, the mutual index coincidence test; all of them were essentially statistical tests.

So there is something which is called correlation. I mean it is very hard to draw line between one attack and the other attack. Actually, you can see the differential attacks and correlation attacks are vastly connected. Then there is more advanced kind of attacks called algebraic attacks. What it does is that, it studies the boolean functions of each obvious ciphers and tries to see whether your boolean functions can be [expanded] in some technique or can be solved essentially.

Therefore, you form an algebraic set of equations and try to solve those system of equations and there was quite a very significant progress in algebraic attacks and lot of stream ciphers were shown to essentially succumb against algebraic attacks. It was supposed to pose a threat against AES also at some point of time, but it has really not sustained and AES is probably got strong against algebraic attacks.

(Refer Slide Time: 04:25)



The slide has a dark blue background with yellow and white text. At the top, the title 'Recap about Differential Cryptanalysis' is written in yellow. Below the title, there is a bulleted list in white text. The first bullet point is 'We have seen in our discussion on Differential Cryptanalysis:', followed by three sub-bullets: '- eliminating high probability differentials guarantees security.', '- if p is the upper bound on the probability of any differential for the cipher, at least 1/p texts are needed to break the cipher.', and '- so to increase the security, reduce p.'. In the bottom left corner, there is a small circular logo with a star and the text 'NPTEL' below it.

### Recap about Differential Cryptanalysis

- We have seen in our discussion on Differential Cryptanalysis:
  - eliminating high probability differentials guarantees security.
  - if  $p$  is the upper bound on the probability of any differential for the cipher, at least  $1/p$  texts are needed to break the cipher.
  - so to increase the security, reduce  $p$ .

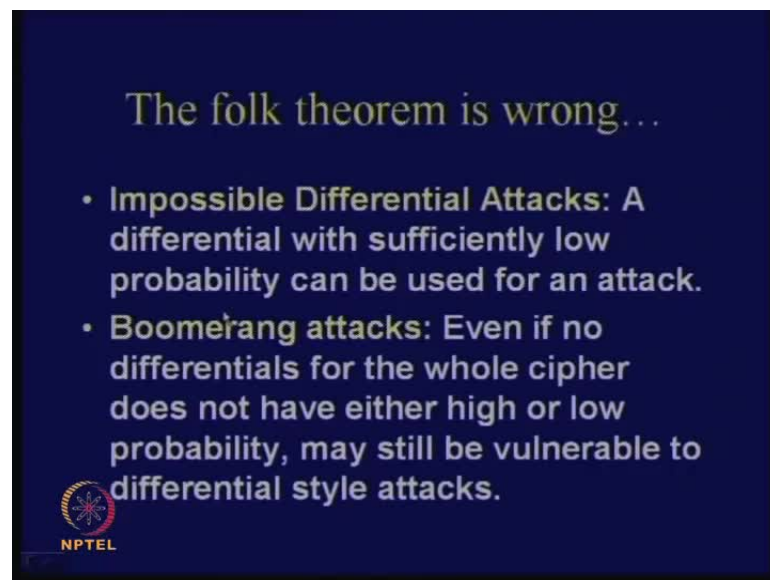
NPTEL

So, that is a debatable issue, but in today's class, we will essentially discuss more about 2 classes of attacks, but why are we discussing about boomerang attacks? First, I will try to stress on that point. So, we have been seeing in our previous discussions on differential attacks that our objective was to eliminate high probability differentials. Therefore, the idea was that if we can eliminate high probability differentials, then we should guarantee security.

So, for example, we can show that if the upper bound of any possible differential is a  $p$  that means all differentials are having a probability which is less than  $p$ . So, then thumb rule is that if you try for say  $1/p$  number of texts, then you can break the cipher. So, what does it mean? **If I am able to if I am** If I would like to increase the security then I would try to reduce the value of  $p$ .

So, differential cryptanalysis gives us this sort of idea that if I am able to reduce the probability  $p$  of any differential of the complete cipher, if the cipher has got  $R$  rounds and if I am able to reduce this probability for say,  $R$  minus 1 rounds, that was the idea of differential attacks, then we should be able to guarantee more security.

(Refer Slide Time: 05:46)

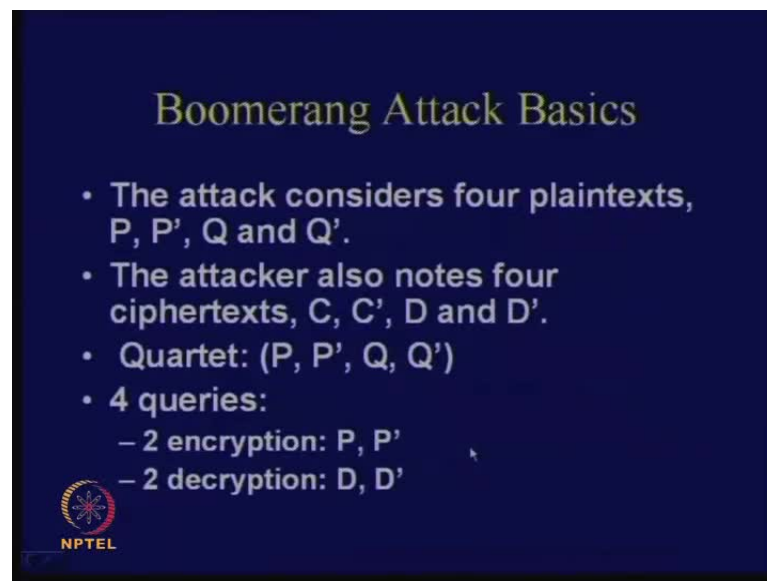


We will see actually this folk theorem is wrong. Therefore, there are 2 classes of attacks which exploit the incorrectness of these assumptions. One is called impossible differential attacks. Although, we will not be discussing on this attack, it mainly shows that if there is a differential whose probability is actually sufficiently low, then also, you can exploit that for your attack. Therefore, it says that if you can show that certain probability, certain differentials will never occur then also, you can use that as a pruning methodology.

Therefore, again you guess the key and again see that whether a particular differential holds or not. So, in that case, it is just the opposite of a differential attack, kind of. Therefore, certain differential can never occur; that is the idea, but the boomerang

analysis is more interesting because what it says is that even if no differentials for the whole cipher has got either high or low probability, then also you can break the cipher using a differential style technique. So, that is more alarming. Therefore, it shows that actually, although your total cipher or complete cipher is quite secure in the sense of differential attacks or **that means if an** even in a sense of impossible differential attacks because that means that it has neither high differential nor very low differential, but even then you can try to mount a differential kind of attack to break the system.

(Refer Slide Time: 07:13)



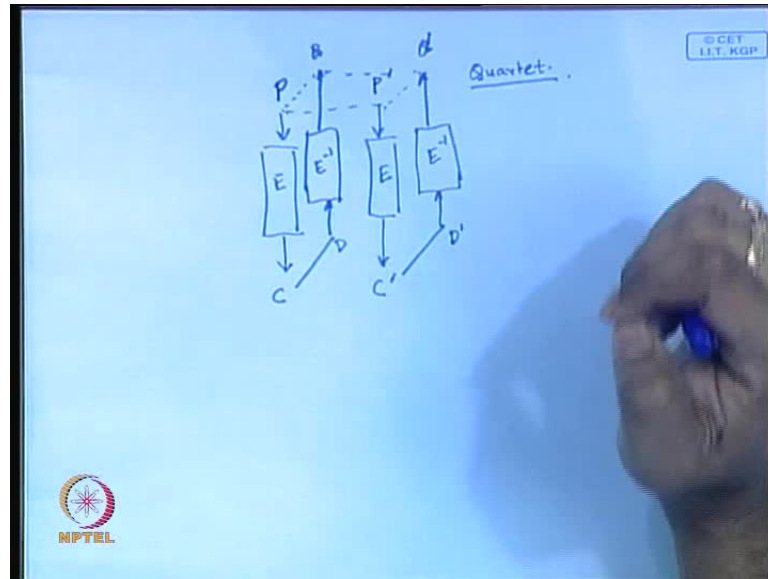
**Boomerang Attack Basics**

- The attack considers four plaintexts, P, P', Q and Q'.
- The attacker also notes four ciphertexts, C, C', D and D'.
- Quartet: (P, P', Q, Q')
- 4 queries:
  - 2 encryption: P, P'
  - 2 decryption: D, D'

NPTEL

So therefore, we will be discussing about this attack which is called the boomerang attack. **Therefore, the boomerang attack basics are essentially upon I mean in** In case of differential, you are considering 2 such plaintexts but in this case, we will consider 4 plaintexts. So, we will be considering P, P dash, Q and Q dash and we will say that P, P dash, Q and Q dash form something which is called as a Quartet. So, the attacker what he does is that he essentially chooses first of all P and P dash and does a decryption and then what he does is that he obtains Q and Q dash by decrypting 2 corresponding cipher texts. Therefore, the idea is as follows.

(Refer Slide Time: 07:58)



So, what you do is that you take  $P$  and  $P'$ . These are essentially 2 plaintexts and then what you do is that you apply the encryption function and I denote that by  $E$  and then you obtain 2 cipher texts. So, you obtain say for example,  $C$  and  $C'$  and from here, you do certain thing and arrive at another 2 cipher texts. I call them  $D$  and  $D'$  and then you decrypt  $D$  by  $E^{-1}$  which is the decryption function and you also decrypt  $D'$  and you obtain 2 plaintexts and I call them  $Q$  and  $Q'$ . Here, this  $P$ ,  $P'$  and  $Q$  and  $Q'$  form something, which I call as a Quartet.

()...

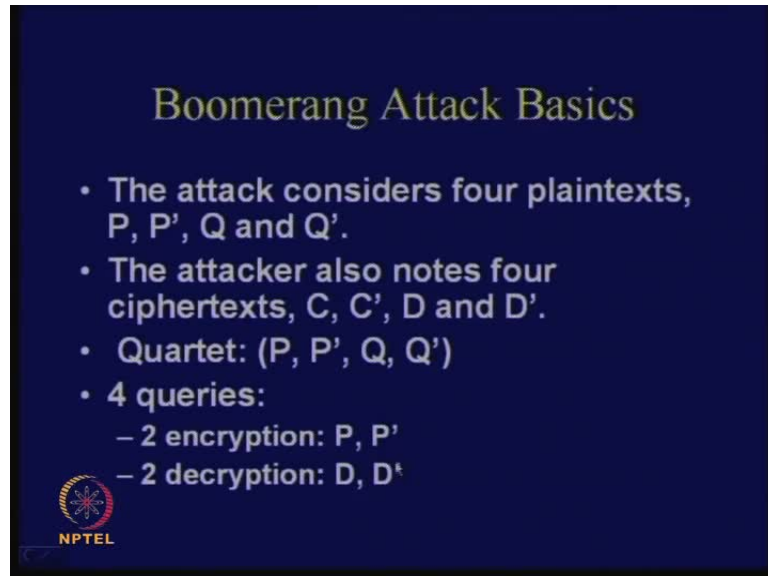
No, you don't know.

Then how do we get the quartet? ()...

So, you know that we have been discussing about various models of attacks. There are some models of attacks, which are called like chosen plaintext attacks, known plaintext attacks. In those cases, what we do assume that we can essentially encrypt or decrypt even if I do not know the key and if you remember that I am giving you certain practical scenario where that is possible as well. So, these are various models of attacks. In linear attacks and differential attacks also, we have assumed that we know the plaintext and cipher texts. How? Because that belongs to certain model like known plaintext attack or a chosen plaintext attack. Therefore, in this case, it is also an example of chosen plaintext


cipher text attack because in this case, we are not only choosing the plaintext, but we are choosing the cipher text also.

(Refer Slide Time: 10:10)



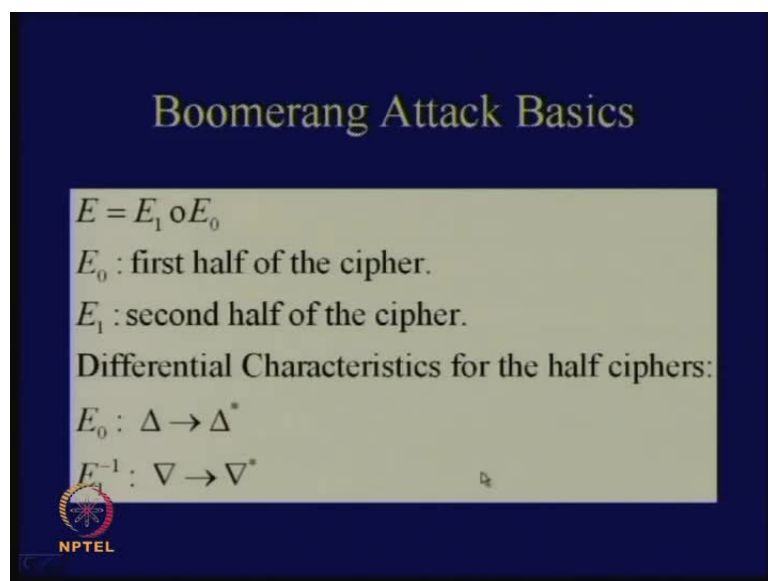
**Boomerang Attack Basics**

- The attack considers four plaintexts, P, P', Q and Q'.
- The attacker also notes four ciphertexts, C, C', D and D'.
- Quartet: (P, P', Q, Q')
- 4 queries:
  - 2 encryption: P, P'
  - 2 decryption: D, D'

 NPTEL

Therefore, this P, P dash, Q and Q dash form something which I called as a quartet. Now, coming back to your slide, you see that there are 4 queries that we have done. You are doing 2 encryption queries for P and P dash and you are doing 2 decryption queries for D and D dash. Therefore, how many queries you are doing? You are doing 4 queries.


(Refer Slide Time: 10:28)



**Boomerang Attack Basics**

$E = E_1 \circ E_0$   
 $E_0$  : first half of the cipher.  
 $E_1$  : second half of the cipher.

Differential Characteristics for the half ciphers:  
 $E_0 : \Delta \rightarrow \Delta^*$   
 $E^{-1} : \nabla \rightarrow \nabla^*$

 NPTEL

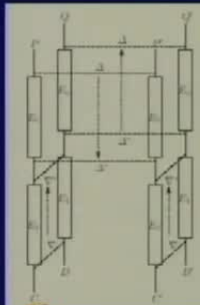


Therefore, now you can actually In our diagram, we have seen that we had the cipher E. What we can do is that we can think that this E is composed of 2 smaller ciphers. I call one of them as E naught; the other one as E 1. So, E naught is the first half of the cipher and E 1 is the second half of the cipher. Half does not mean exactly half; half just means portion - the first portion and the second portion. So, what you do is that you do differential characteristics for the half ciphers. What do you say? You take E 0 and you give that so before What you do is that you obtain differential characteristics for half of the ciphers. Even if you are not able to obtain good differential characteristics for the entire cipher, suppose you are able to obtain differential characteristics for half the cipher.

What you do is that you obtain characteristics for E 0; you obtain the characteristics for E 1 inverse. So, the characteristics for E 0 is denoted by this particular characteristic that is, you give delta and you obtain delta star and for E 1 inverse, you give lambda and you obtain lambda star. These are your individual half characteristics for half the cipher that you have done again by observation and suppose that these particular characteristics have got reasonably high probability. Although, you do not have good differential characteristics for the total cipher, but there are reasonably high probability for the half ciphers. Is this portion clear?

(Refer Slide Time: 12:04)

### Boomerang Attack Basics




$$\begin{aligned}
 E_0(Q) \oplus E_0(Q') &= E_0(P) \oplus E_0(P') \oplus E_0(P) \oplus E_0(Q) \oplus E_0(P') \oplus E_0(Q') \\
 &= E_0(P) \oplus E_0(P') \oplus E_1^{-1}(C) \oplus E_1^{-1}(D) \oplus E_1^{-1}(C') \oplus E_1^{-1}(D') \\
 &= \Delta^* \oplus \nabla^* \oplus \nabla^* \oplus \Delta^*
 \end{aligned}$$

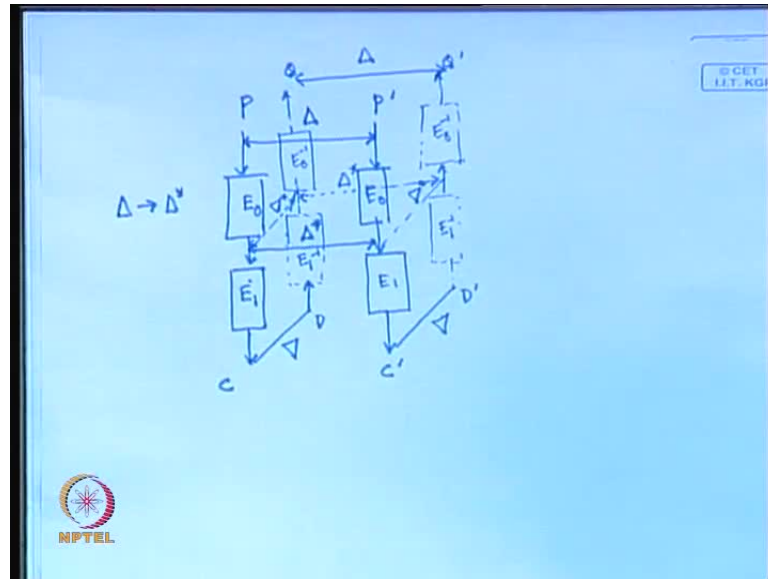
Note that this characteristic is the same as that of the inverse of  $E_0$ .

Thus, the difference in the plaintexts Q and Q' is the same as that in P and P'.

Hence, the name is "Boomerang".

 NPTEL

(Refer Slide Time: 12:26)



What you do after this is that you start observing something like this. Therefore, the same diagram you can observe by splitting up into 2 parts -  $E_0$  and  $E_1$  and also that for the entire quartet. **Now, you come to this diagram and I will show you with** What you do is this. Now, the same diagram what you do is that you see that  $P$  and  $P'$  and you observe by breaking up the cipher  $E$  into 2 components; one of them is  $E_0$  and the second component is  $E_1$  and you obtain the corresponding cipher; that corresponding cipher is denoted by  $C$ .

The same thing you have done for  $P'$  also.  **$P'$  also has** You operate  $E_0$  and you operate  $E_1$  and you obtain  $C'$ . Now, you assume that this  $P$  and  $P'$  maintain a constant differential,  $\Delta$ . In that case, if I assume that  $E_0$  passes a differential say,  $\Delta$  to  $\Delta^*$  that means that at this point, we have an expected differential of say,  $\Delta^*$ .

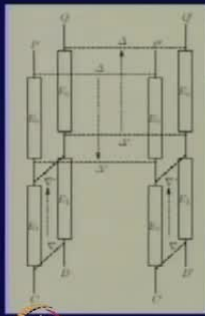
Now, what you do is that you have obtained  $C$  and  $C'$ , you apply an ex-or function you take the cipher text and you apply an ex-or function of say  $\lambda$  and you obtain 2 cipher texts called  $D$  and  $D'$  and then what you do is that you start the opposite operation; that is you start the decryption. So, you take this and you apply the decryption function which means you apply  $E_1^{-1}$  and also, apply  $E_0^{-1}$ . Same thing you do for this also. We apply  $E_1^{-1}$ ; we apply **and obtain**  $E_0^{-1}$  and then you obtain, what do you obtain here?  $Q_1$  and  $Q_1'$ . Now, you see that for  $E_1^{-1}$ , you

have basically given a differential of lamda and therefore, what you would have expected here is that the differential here is lamda star and same for here also. In that case, what is the corresponding differential existing at this point between these 2 things? What is the corresponding differential? It is the same as the delta star. Therefore, you obtain delta star here also.

Now, since we know that delta gives delta star; by symmetry, we know that delta star will also give delta. So, **what you obtain is that** you are expecting here a differential of delta again. So, it is something like a boomerang. You see that you are throwing a differential of delta and you are expecting back differential of delta. So, the idea is you know it is similar to **a what you like** a boomerang. Therefore, you throw something and you obtain back something.

(Refer Slide Time: 15:35)

### Boomerang Attack Basics



$$\begin{aligned}
 E_0(Q) \oplus E_0(Q') &= E_0(P) \oplus E_0(P') \oplus E_0(P) \oplus E_0(Q) \oplus E_0(P') \oplus E_0(Q') \\
 &= E_0(P) \oplus E_0(P') \oplus E_0^{-1}(C) \oplus E_0^{-1}(D) \oplus E_0^{-1}(C') \oplus E_0^{-1}(D') \\
 &= \Delta^* \oplus \nabla^* \oplus \nabla^* = \Delta^*
 \end{aligned}$$

Note that this characteristic is the same as that of the inverse of  $E_0$ .

Thus, the difference in the plaintexts  $Q$  and  $Q'$  is the same as that in  $P$  and  $P'$ .

Hence, the name is "Boomerang".

NPTEL

So, this is the entire basic principle behind the operation of boomerang attacks. You see that this same thing is being denoted here by some equations, but what is the idea is that at this point like  $E_0(Q)$  and  $E_0(Q')$ , I am expecting here a differential of delta star because if I know that here the differential is delta star - we discussed why it is delta star, then it implies that here, I am expecting a differential of delta.

So, note this characteristic is same as inverse of  $E_0$ . **Therefore, it is same as that of therefore, this we discussed already that** I mean if you have characteristic for  $E_0$ , we have the similar characteristics for  $E_0$  inverse also. Thus the difference in the plaintexts


in Q and Q dash is the same as that in P and P dash and hence the name boomerang. Therefore, you are throwing a differential and you are obtaining back the differential; you are getting back the differential.

(Refer Slide Time: 16:29)

**Example: COCONUT98**

- Designed to protect against DC.
  - full cipher provides no good differential characteristics.
- Uses a 256 bit key,  $K=(k_1, k_2, \dots, k_8)$

i	1	2	3	4
$k_i$	$k_1$	$k_1 \wedge k_3$	$k_1 \wedge k_3 \wedge k_4$	$k_1 \wedge k_4$
i	5	6	7	8
$k_i$	$k_2$	$k_2 \wedge k_3$	$k_2 \wedge k_3 \wedge k_4$	$k_2 \wedge k_4$



So we will study one cipher to understand how the boomerang attack works. The name of the cipher is called COCONUT98 and it was given by S Vaudenay and I will just give the reference for this particular work and it was given with the idea that they essentially give a theory which will always protect against differential cryptanalysis. So, it was very nice theory; it was known as the decorrelation theory and using a decorrelation theory they proved that actually you cannot obtain any good differential characteristics for the entire cipher.

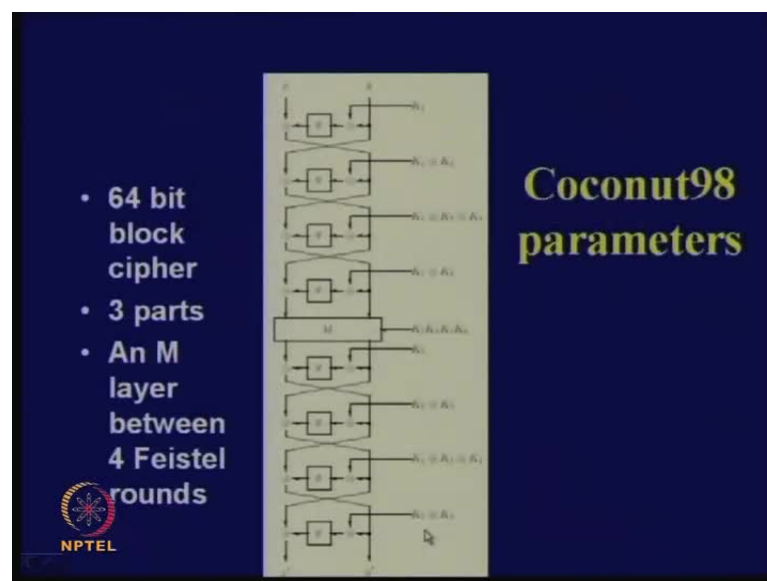
But then people again tried, I mean, found out this boomerang attack and showed that although you are not able to do a differential cryptanalysis, but you can still do an attack because half of the cipher was not so strong. The full cipher was quite strong, but half of the cipher was not really strong. **So, this attack was given by a pro from UC Berkeley; his name is David Wagner.** So, I will give a reference for that also. If you are interested, you can download and study that paper also. Therefore, what is the idea? The idea is that we are essentially using a 256 bit key in this coconut cipher and this is essentially a 64 bit block cipher. It is something like a feistel cipher which we have studied.

So, this is roughly speaking. It is not so important for us, but just observe that there are The idea of a COCONUT 98 cipher was that it had 3 components. It got 2 feistel layers and in between, there was some M layer. I call M layer; actually, it is a decorrelation layer, but for our simplicity, let us call it an M layer. You know that if there are 4 For each feistel layers, there are 4 rounds. So, there are 4 rounds at the beginning and 4 rounds at the end. How many keys are required - round keys? I required 8 round keys.

This is the very simple key scheduling algorithm which says how you are generating those 8 round keys. You take  $k_1$ ; Suppose 256 bit key, you can decompose into 8 components, each of them of size 32 bits, right. Therefore, you take  $k_1$  of 32 bits and that suppose that is that gives you the first round key, for second round key, you can just x-or that with  $k_3$  and you can obtain the second round key, for third round key you x-or that with  $k_4$  and you obtain this and for fourth round key, you x-or these with  $k_3$ . You see that x-oring with  $k_3$  and  $k_4$  alternatively.

Therefore, you can actually implement this quite easily in hardware or any other implementation. So, you take  $k_2$ , you x-or this with  $k_3$ ; you take  $k_2$ , x-or  $k_3$ , x-or with  $k_4$  and again you take  $k_2$ , x-or  $k_3$ , x-or  $k_4$ , you x-or that with  $k_3$  and you obtain the fourth 8 round key.

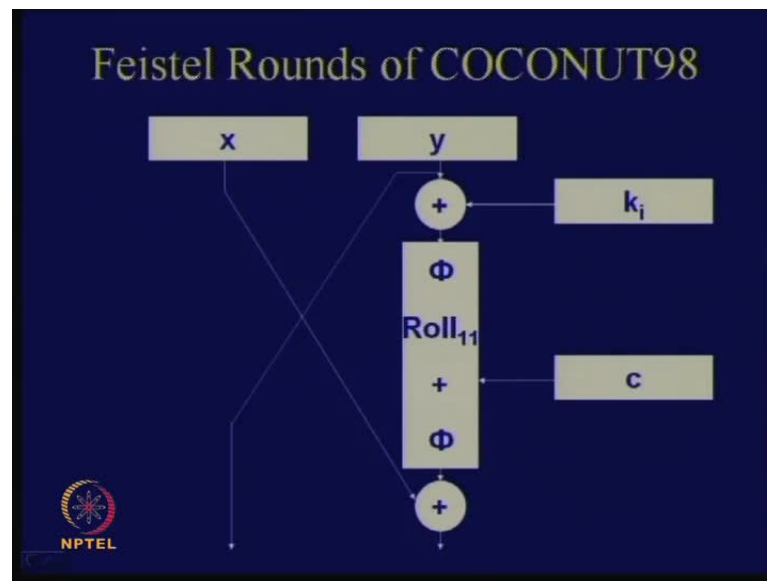
(Refer Slide Time: 19:24)



So, it is a very simple technique to generate all the round keys. This is how the coconut cipher looks like. The details are in this case again elaborate, but broadly this means, it

looks like this. It is a 64 bit block cipher. So, that means that these 2 components are each of 32 bits and you see that this network just looks like a crystal cipher; it looks like DES. Therefore, there are 3 parts. There is a M layer between 4 feistel rounds. There are 4 feistel rounds here, 4 feistel rounds at the end, in-between there is a layer, which we call as M layer.

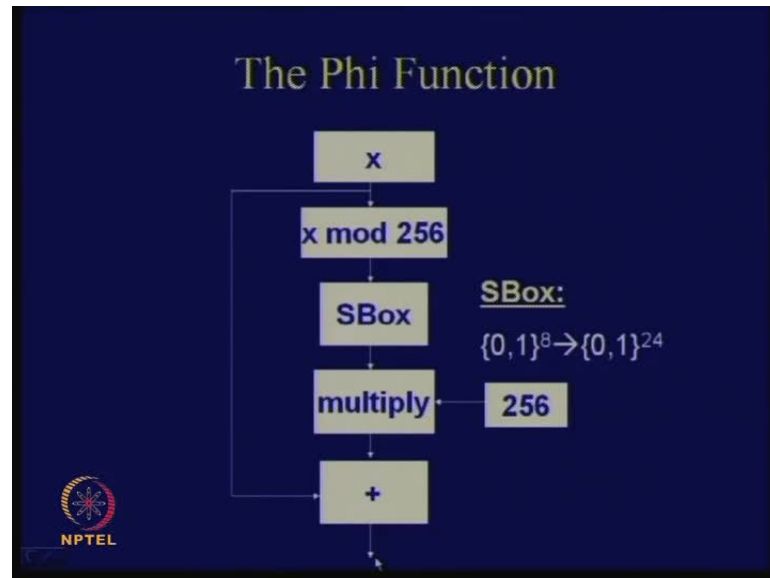
(Refer Slide Time: 19:56)



So, let us look into the feistel rounds. **The feistel round just remembers us of DES.** You see there are 2 parts - left and the right part. You take the right part and bring it to the left. So, that forms your left part and in order to obtain the right part, you take  $y$  and pass that through a function,  $x$ -or that with  $x$  and obtain the corresponding output. Here, you see that you have first of all done a round key and after that you have done a certain sequence of operations and there is something which is a constant  $c$  also.

What is the operation size here? The operation size is 32 bits because this part is only of 32 bits. You see that there are some components here. One is called  $\phi$ , one is called Roll 11 and then there is an integer addition and then, there again is an application of the function  $\phi$ . What is Roll 11? Roll 11 means you are doing a 11 bits circular shift to the left. It just means you take the value, I mean the entire output and you just do the circular shift to the left by 11 steps.

(Refer Slide Time: 21:02)



Let us see, what is the function phi? The phi function is this. That is, you take  $x$ . What is the size of  $x$ ? 32 bits. So, you do an  $x \bmod 256$ . What does it mean? Again  $x \bmod 256$  means you are taking only 8 bits. You are taking all the 8 bits and you are basically neglecting the other part and that you are passing through the SBox which actually takes 8 bits and produces a 24 bit output. So, this is the example of an expansion SBox. You take this S-box and you obtain 24 bits here and then you multiply that with the constant called 256 and then, that you add by the integer addition with  $x$ . What is the size of  $x$ ? 32 bits. You add this and you do modular 2 power 32 addition and you obtain the corresponding output. So, this is the basic idea behind the function phi. Is this clear?

(Refer Slide Time: 22:12)

The M layer


$$M(xy) = (xy \oplus K_5 K_6) \times K_7 K_8 \text{ mod } GF(2^{64})$$

Here,  $p(x) = x^{64} + x^{11} + x^2 + x + 1$

Design is based on decorrelation theory.

If  $K_7 K_8$  are unknown then the probability of a non-zero input differential to produce an output differential is  $1/(2^{64}-1)$ .

But for a fixed key, the output differential does not depend on the input value but depends only on the input differential.

 NPTEL

Then what we do is that we see what M layer is. The M layer is as follows. You take  $xy$ . What is the size of the  $xy$ ? 64 bits. Then you use the components  $K_5$ ,  $K_6$ ,  $K_7$  and  $K_8$  of the key and what you do is that you take  $xy$ , x-or that with  $K_5 K_6$  and multiply that with  $K_7 K_8$ . Note that this multiplication is in the Galay Field  $2$  power  $64$  and this is the reduction polynomial which they have used. Anyway this is just the details. So, this M layer is actually designed based on something, which we call as the decorrelation theory and the idea is as follows. **That you see one particular you can.** If you do a differential analysis of this layer, then what do you observe? If you do a differential analysis of this particular layer then  $K_5 K_6$  gets cancelled. You see that because you are doing an x-or here. All of you see that.

You take  $xy$  and you also take another  $x$  dash  $y$  dash and you know that  $x$  dash  $y$  dash x-ored with  $K_5 K_6$ , multiplied with  $K_7 K_8$  and you take an x-or between them, what you get is this particular  $K_5 K_6$  cancels with other  $K_5 K_6$  and what you obtain is the differential, but the differential is multiplied with  $K_7 K_8$ . Therefore, what does it mean is that if  $K_7$  and  $K_8$  are unknown, then the probability that a non-zero input differential will produce an output differential is actually  $1$  by  $2$  power  $64$  minus  $1$  because if I keep on changing the values of  $K_7 K_8$  and how many choices of  $K_7 K_8$  are there? There are  $2$  power  $64$  choices and I know that the  $0$  value can never occur.



So that excludes that 0 value. There are totally  $2^{64} - 1$  non-zero differential that are expected and out of them, I am interested in only one case. So, the probability is one by  $2^{64} - 1$ , if the values of  $K_7$  and  $K_8$  are unknown, but if the values are  $K_7$  and  $K_8$  are known, then automatically, the differential gets fixed. But for a fixed key, the output differential does not depend on the input value. That is the main idea, but it depends only on the input differential. So, if I fix the key, the output differential actually does not depend upon the input value, but it depends only on the input differential.

(Refer Slide Time: 24:59)

Handwritten mathematical derivation on a blue background:

$$M(xy) = (xy \oplus k_5 k_6) \times k_7 k_8$$

$$M(xy') = (xy' \oplus k_5 k_6) \times k_7 k_8$$

$$M(xy) \oplus M(xy') = (xy \oplus xy') \times k_7 k_8$$

The diagram below shows a box labeled 'M' with an arrow pointing to it from the right labeled  $k_7 k_8$ . Above the box is a double-headed arrow labeled  $\Delta$ . Below the box is a double-headed arrow with a question mark below it.

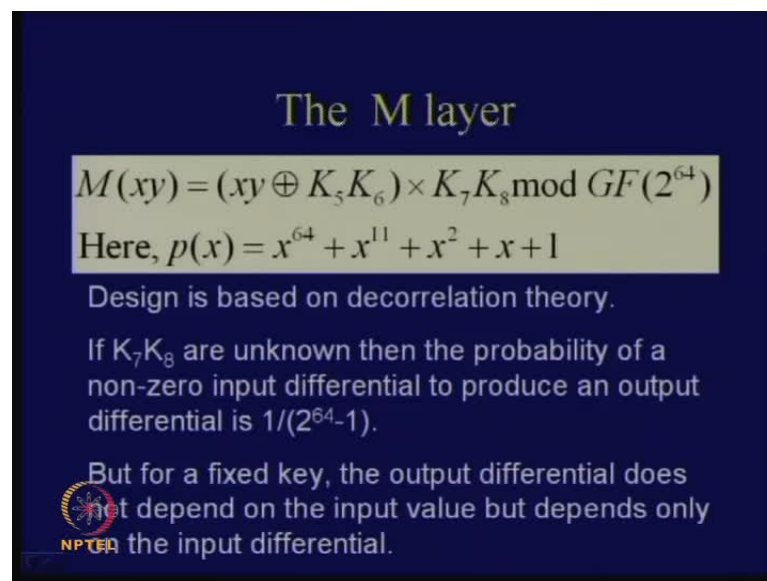
If I do not know the value of  $K_7$ ,  $K_8$  then it is very hard to predict the differential. All the differentials have got uniform probability. Therefore, you see where is the problem? If I do a differential - normal standard differential attack here, analysis here. You see that I have got  $M(xy)$ ; this is equal to  $xy$  x-ored with  $K_5 K_6$  and multiplied with  $K_7 K_8$ ; forget the modulo at this point. So, I take again  $M(x \text{ dash } y \text{ dash})$ . I call that  $x \text{ dash } y \text{ dash}$ ; that is some other text basically.

What you do is that you take  $x \text{ dash } y$ . I call that  $x \text{ dash } y$ . You x-or that with  $K_5 K_6$  and multiply that with  $K_7 K_8$ ; so, this is the other pair. Now, if we do a differential of these 2 things then what is the expected differential at the output? This will depend upon  $xy$  x-ored with  $x \text{ dash } y \text{ dash}$ , but this differential is multiplied with  $K_7 K_8$ . That means that if I am doing a differential analysis then what I am trying to do is that I am trying to force a

differential at the input of your M layer. So, we call that M layer. I am trying to force the differential here and I am trying to compute what is the probability distribution of the differentials here.

So, here you see that if I do not know the value of this corresponding key called K 7 and K 8, then I do not have any preference of any key. I mean for every key, you essentially see that if I change this value then automatically, if I change this delta here then I get a different delta here. I change this K 7 K 8 to another value, I obtain another delta. Therefore, if I do a similar kind of differential distribution table then I will see that the probabilities are called uniform. **For every key, you will get that** I mean for every choice of key, the probability is 1 by 264 minus one because the 0 differential will not occur. That is the idea.

(Refer Slide Time: 27:19)



The M layer


$$M(xy) = (xy \oplus K_5 K_6) \times K_7 K_8 \text{ mod } GF(2^{64})$$

Here,  $p(x) = x^{64} + x^{11} + x^2 + x + 1$

Design is based on decorrelation theory.

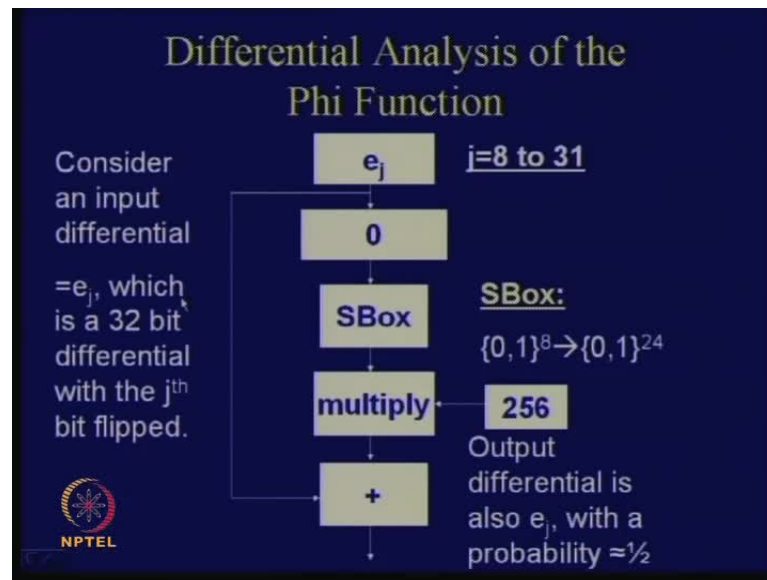
If  $K_7 K_8$  are unknown then the probability of a non-zero input differential to produce an output differential is  $1/(2^{64}-1)$ .

But for a fixed key, the output differential does not depend on the input value but depends only on the input differential.

 NPTEL

So, what you see that using this particular technique, the designers were able to impose a uniform distribution among the differentials. Therefore, doing a standard differential attack at this point is quite hard because of this. Because now, we have been able to develop certain technique through which you can have a uniform distribution of the differentials. So, that is the basic idea of the decorrelation theory.

(Refer Slide Time: 27:21)



Now, we will do a differential analysis of the phi function also. **What it says is that you suppose you know** This is how the phi function was looking like. **Only so** Let us do one thing; let us subject it towards a certain differential. So, we take  $e_j$ . What is  $e_j$ ? The  $e_j$  is an input differential, a 32 bit differential in which  $j^{\text{th}}$  bit is flipped. So, it flips only one particular bit. You see that  $j$  is from 8 to 31. Then what is the differential here? It is actually 0 because here we are taking only the last 8 bits and has not varied those bits.

Therefore, here you are obtaining the same input. S-box also does not compute on the same values and it really does not matter what goes inside. What about this part? Again you see that  $e_j$  is transferred here. Now, you see that you are basically doing an integer modular  $2^{32}$  addition, where one of the components is flipped by 1 bit and the other part is held fixed.

Then the probability that in the output you will find that the  $j^{\text{th}}$  bit is flipped is actually half. Why? Because it depends upon whether your carrying bit at that point is 1 or 0; because here we are doing integer addition. So, integer addition means that there is an input carry also. If your input carry is 0, then automatically it means that the  $j^{\text{th}}$  bit of the output is also flipped. So, therefore, if we do a little bit of analysis, you will see that this probability, if you study the carry bit, it is actually close to half; for this class, you just assume this fact.

**[But where is this carry bit occur]**


Here, you are doing an integer addition, you know, normal addition. So, you are taking this and you are taking this; so, this is a normal integer addition. Although I am saying at this point the differential is 0, but the actual value is not 0. You are processing on 2 inputs, but the idea is that for both the cases - that is, for both the differential pairs, you are essentially processing on the same input at this point, but where is this part of the input. Here only the  $j$ th bit is flipped and what is the probability that in the output also the  $j$ th bit is flipped. That depends upon your carry and how you are able to control the carry bits.

You can actually show that the probability is very much close to half. **so this point** So, actually, the carry bits follow very neat probability distribution. That is, I mean not an objective of our class, but **we can actually see at this probably I mean** you can actually prove theoretically what is the probability that a particular carry bit will be 0 and obviously, when it is 1. It is actually half plus half to the power of  $I$  plus 1 where  $I$  is your bit location; we can actually theoretically prove that also.

(Refer Slide Time: 30:31)

**Differential taking into account**  
**ROL<sub>11</sub>**

- ROL<sub>11</sub> is a circular shift by 11 bits.
- If the entire Feistel function is considered, there are 3 additions.
  - $(x+a \bmod 2^{32})+b \bmod 2^{32}$  is equivalent to  $x+c \bmod 2^{32}$ , where  $c=a+b$
- Thus the output differential is  $e_{j+11}$ . The subscripts are taken modulo 32.
- Similarly,  $e_j \wedge e_k \rightarrow e_{j+11} \wedge e_{k+11}$  with probability  $\approx 1/4$

  
NPTEL

Anyway that is not important. What is important is that you are able to get a differential in your file layer with a probability of half. So, the ROL 11 was the circular shift by 11 bits. Therefore, what it will do is that it will take the  $e_j$  input. (Refer Slide Time: 30:54) You see that here what you have obtained is, if you are given an input differential of  $e_j$ , you are able to obtain an input differential here also of  $e_j$  with a probability of half, but

what are the subsequent operations? (Refer Slide Time: 31:06) Here, the subsequent operations are that after a phi layer, we are doing a Roll 11 and then again do an integer addition and again followed by a phi function.

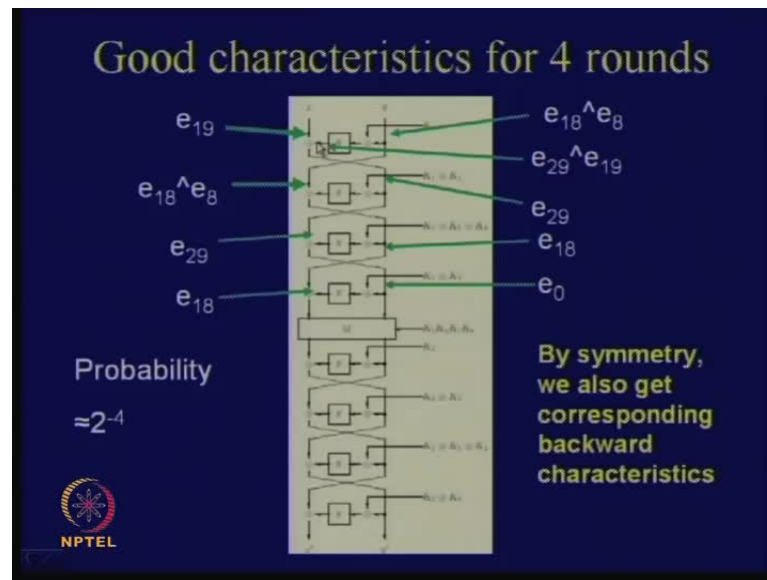
So, you see that in this phi function that is one integer addition. When you are adding with  $c$ , that is one integer addition and when you are doing phi function that is another integer addition. **Therefore, now what is the probability that so** If I do a  $e_j$  here, I should be able to obtain the  $e_j$  here also and the probability of that is half; that we have seen. Therefore, due to this rotate 11, this  $j$  bit gets shifted to  $j + 11$ th location; it gets shifted to the left.

But what about this plus function? If I would still like to maintain the differential at the  $j + 11$ th location, then also I need to control these carry bits and similarly, also for the phi function. Therefore, we may think that what we have obtained for one integer addition, we may have to multiply the probability for the 3 successive integer additions but, actually, it is not so because these 3 integer additions I can squeeze them and write as 1 integer addition.

Therefore, the idea is because of this property that you know that if I do  $x + a \pmod{2^{32}}$  and then add  $b \pmod{2^{32}}$ , this I can write as  $x + c \pmod{2^{32}}$  which means, although you have to control 3 integer carries, 3 integer additions, we actually can squeeze them into one single integer addition. Therefore, the probability that this input differential of  $e_j$  will actually yield an output differential of  $e_{j+11}$  is actually  $2^{-2}$  and not  $2^{-1}$  into  $2^{-1}$  into  $2^{-1}$ .

Similarly, you can obtain other such kind of differential also. For example, you take  $e_j$  and you x-or that with  $e_k$ , the probability that this will yield  $e_{j+11}$  x-ored  $e_{k+11}$  is actually  $2^{-4}$ . Why? Because you take half and you take another half. So, you multiply them because they are independent. So, you get  $2^{-4}$  assume them to be independent.

(Refer Slide Time: 33:14)



We will study one good characteristics propagation for this 4 round. It is quite interesting. You see that suppose I subject this feistel network to  $e_{19}$  and I subject this part to  $e_{18} \oplus e_8$ . **so what is the corresponding** Because of the feistel network, this gets transferred here. So, you obtain  $e_{18} \oplus e_8$  here. Therefore, what is the expected differential of the corresponding feistel function here?

(Refer Slide Time: 33:49) So, you see that we have studied is that if I take  $e_j \oplus e_k$ , it gets transformed to  $e_{j+11} \oplus e_{k+11}$ . So, here we take  $e_{18}$  and  $e_8$  and add 11 to this index and add 11 to this index also and remember that when we are doing this addition on the indices, then we are doing a mod 32. So, what is  $18 + 11$ ? 29. So, you obtain 29 here and what is  $8 + 11$ ? It is 19.

Now, you obtain a differential here, which is equal to  $e_{29} \oplus e_9$ , right. If I take an x-or between these two, you see that  $e_{19}$  gets cancelled; so you have only  $e_{29}$ . So, this  $e_{29}$  now gets to the right side and you obtain  $e_{18} \oplus e_8$  here. What is the expected differential here? Now, at this point, it is  $e_{29} \oplus e_8$ . What is  $29 + 11$ ? 40. So, you take 32, it is 8.

Therefore,  $e_8$  gets cancelled and we have only  $e_{18}$ . **So, you see  $e_{18}$  comes to right and to the left what comes?**  $e_{29}$  comes. So, if you take  $e_{18}$  here, then add 11 to 18, what do you get? You get 29. So, that cancels this one. Actually, you should get 0 here. I have written this wrong; actually, this will be  $e_0$ . So, this is 0 and you have obtained here  $e_0$ .

18. Therefore, if you see that if you throw a differential of  $e_{19}$  and  $e_{18}$  XORed  $e_8$ , then the probability that you will get here  $e_{18}$  and 0 is actually, approximately equal to half into half into half. So, that means, it is actually equal to  $2^{-4}$ .

So, this is a standard differential analysis. Therefore, if I obtain a differential in this nature, by symmetry, we can also do the same thing for the backward thing also. Like, if I throw  $e_{18}$  and  $e_8$  here or rather  $e_{18}$  and 0 here, we should obtain back  $e_{19}$  and  $e_{18}$  XORed  $e_9$  or  $e_8$  with the probability also of  $2^{-4}$  or close to  $2^{-4}$ .

Therefore, I obtain a differential for the forward transformation; I also obtain the differential for the inverse transformation. Now, what is my setting? The setting is that I have obtained differential for half of the cipher - one part of the cipher and similarly, I try to obtain the differential for the we have obtained that for the other half of the cipher also. Now, see that you can relate it to the boomerang attack. What you can do is that you can call this as  $e_0$  and you can call this part as  $e_1$ .

(Refer Slide Time: 36:42)

**Success Probability**

Define the complete cipher,  $E = \varphi_1 \circ M \circ \varphi_0$   
 Here,  $E_0 = \varphi_0, E_1 = \varphi_1 \circ M$   
 It does not matter that  $M^{-1}(\nabla^*)$  is unknown to attacker. What is important is it depends only on the key and not on the values of the ciphertexts.  
 Define,  $p_{\Delta^*} = \Pr[\Delta \xrightarrow{e_0} \Delta^*], q_{\nabla^*} = \Pr[\nabla \xrightarrow{e_1} \nabla^*]$   
 Success Probability  $\approx \sum_{\Delta^*} p_{\Delta^*}^2 \sum_{\nabla^*} q_{\nabla^*}^2$   
 Fact: If,  $\Delta = \nabla = (e_{10}, e_{31})$  provides  $p \approx 1/1900$ .

So, you take or include the M layer into the  $e_1$  cipher. What you can do is that you can write this in this form like you can take the cipher  $e$  and I assume that it is composed of  $\psi_1 M$  and  $\psi_0$ . So,  $\psi_0$  is the first part and the other part is  $\psi_1$  and M. Therefore, you apply M layer and the  $\psi_1$ . What is  $\psi_1$  and  $\psi_0$ ? They are the feistel layers. So, you can basically break your cipher into 2 component ciphers.

(Refer Slide Time: 37:07)

The slide has a dark blue background with yellow text. At the top, the title 'Obtaining full round characteristics' is written in a serif font. Below the title is a bulleted list of four points. At the bottom left is the NPTEL logo, and at the bottom right is a mathematical equation in a light-colored box.

### Obtaining full round characteristics

- Need to find some way to take advantage of these half round characteristics.
- The M layer creates problem for standard DC.
- Boomerang attack helps us to control the effect of the M layer.
- Key idea! M is affine. So, for a fixed key, there is an excellent characteristics with probability 1:

NPTEL

$$\nabla^* \rightarrow M^{-1}(\nabla^*)$$

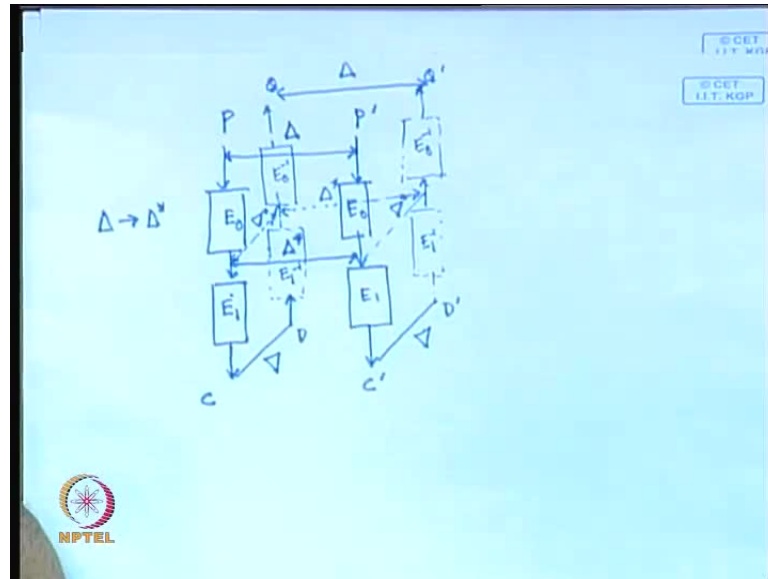
**So now you have obtain** In order to obtain the full round characteristics, you see that you need to find some way to take advantage of these half round characteristics. You have obtained some half round characteristics and I would like to **obtain some way some technique** find out some way through which I can take advantage of these half round characteristics. The M layer creates problem for standard DC; we have discussed why. So, the boomerang attack helps us to control the effect of the M layer. What it does? The key idea is like this: that since M is affine, if I fix the value of the key and if you throw an input differential, then the output differential is actually expected with the probability of 1.

So, that is the basic idea. Therefore, if I throw in a differential of  $\lambda^*$ , then your M inverse, that is, the inverse of your decorrelation of the M box or the M layer, then you are expecting a differential of M inverse of  $\lambda^*$ . **What is the idea therefore,** What is the probability of this? The probability of this is 1, if this key is fixed and the central idea behind boomerang attack is that I am not really interested in the value M inverse  $\lambda^*$ ; in differential attacks, we were interesting in the value of a differential at the output of the R minus 1 th round

But in boomerang attack, I am not interested in the value of the differential. What I am interested is in the fact that this particular differential M inverse  $\lambda^*$  exists. That is, there is a particular differential for which this actually holds.



(Refer Slide Time: 38:51)



So, coming back to this diagram, I mean, you can see this. That is, when we are throwing this differential here - that is delta here, I was not really bothered about what is the value of this delta star. What was important to us was the fact that actually you had some delta star. I am not really bothered about the value of the delta star.

**So then what I do is that I take some differ I mean some** So, I obtain this and I apply some lamda here and some lamda here. I am again not bothering in the value of this lamda star. What I am bothered is that both the ciphers end up in having the same differential. Then again this property holds that I give delta and I get back delta as the differential; that is what I am more interested in. I am not really bothered in the values of the internal differential; so, that is the key. That is another fundamental difference between the idea of boomerang attacks and normal differential attacks. So, in that case, I am not really bothered about whether I throw in a differential of lamda and I obtain back differential of M inverse lamda and what is the value of the M inverse lamda. I know that there is an M inverse lambda which exists.

(Refer Slide Time: 40:02)

**Success Probability**

Define the complete cipher,  $E = \varphi_1 \circ M \circ \varphi_0$   
Here,  $E_0 = \varphi_0, E_1 = \varphi_1 \circ M$   
It does not matter that  $M^{-1}(\nabla^*)$  is unknown to attacker. What is important is it depends only on the key and not on the values of the ciphertexts.  
Define,  $p_{\Delta^*} = \Pr[\Delta \xrightarrow{e_0} \Delta^*]$ ,  $q_{\nabla^*} = \Pr[\nabla \xrightarrow{e_1} \nabla^*]$   
Success Probability  $\approx \sum_{\Delta^*} p_{\Delta^*}^2 \sum_{\nabla^*} q_{\nabla^*}^2$   
Fact: If,  $\Delta = \nabla = (e_{10}, e_{31})$  provides  $p \approx 1/1900$ .

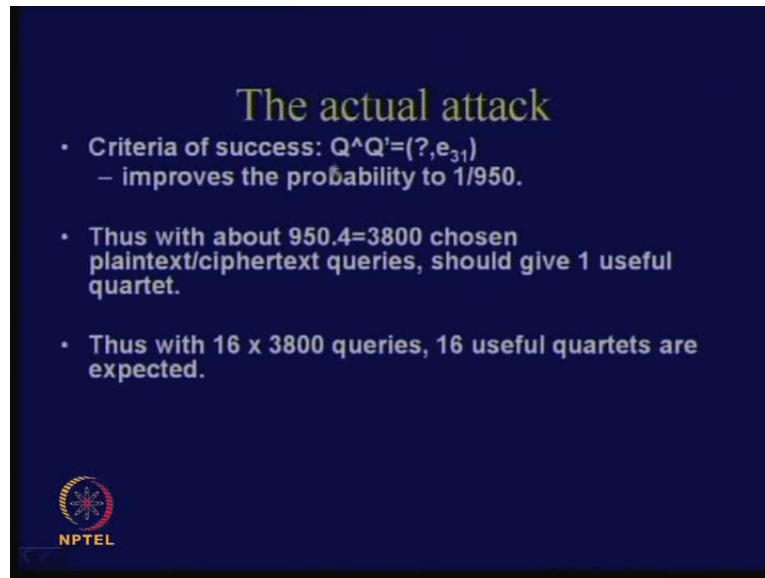
Therefore, what does it mean? It means that it does not matter that  $M$  inverse lambda star is unknown to the attacker. What is important is that it depends on the differential and not on the cipher text values. We define these 2 probabilities. That is the first probability is that on the **psi** 0 layer that is the on the first part, we have a probability and I designate that as if we throw in delta, you get back some delta star. So, I denote that by some  $p$  delta star and similarly, for the inverse transformation, we throw in lamda, you get back lamda star. So, it is the inverse feistel ciphers notation and I denote that probability by say  $Q$  lambda star.

In this case, you can see that your success probability is actually square of this and square of this and a sigma over this. Why? Because there are four - there are 2 differentials of this nature and there are differentials of this nature. I am really not bothered in the value of delta star or in the value of lambda star. I take a sigma over these 2 things and **you can see that these I mean** this was worked out for an example delta lamda equal to  $e_{10}, e_{31}$  and it was found out that this probability was around 1 by 1900. That is some empirical value. You do an analysis and you find out and you can do a similar kind of analysis like we have seen for our half differentials.

Now, you see that what you have done is that we have thrown in a differential of  $e_{10}, e_{31}$  and you are expecting that this differential should come back to your  $Q$  and  $Q$  dash. If I do not expect anything for this  $e_{10}$  **then what is the probability**, and if I guess only on


the half of the stuff then what is my probability? My probability doubles; my probability becomes equal 1 by 950. Therefore, now you should be able to understand the actual attack.

(Refer Slide Time: 42:01)



The actual attack

- Criteria of success:  $Q \oplus Q' = (e_{10}, e_{31})$ 
  - improves the probability to 1/950.
- Thus with about  $950 \cdot 4 = 3800$  chosen plaintext/ciphertext queries, should give 1 useful quartet.
- Thus with  $16 \times 3800$  queries, 16 useful quartets are expected.

 NPTEL

Therefore, the actual attack works like this. **What he does is that he expects that ok, you throw in the differential of as I told you** (Refer Slide Time: 42:10) So, you throw in a differential on  $P$  and  $P$  dash of these values, that is,  $e_{10}$  and  $e_{31}$  and we expect that because of the boomerang attack, you should get back  $Q$  x-ored with  $Q$  dash and that should be also be equal to  $e_{10}$  and  $e_{31}$ , but let us not assume anything on the first part; let us just assume on the second part. Why? It is because the probability of that actually increases and I will have more number of cases, which satisfy this fact. So, my probability is around 1 by 950.

**So you know that** Therefore, how many queries you are doing for one boomerang effort? You are doing actually 4 queries. Therefore, if you do inverse of this, that is, to obtain 950 quartets, then for obtaining these quartets, you have to do 4 encryption or decryption queries. Four, I mean chosen plaintext or cipher text queries. So, totally you do 950 into 4 - that is, around 3800 chosen plaintext or cipher text queries.

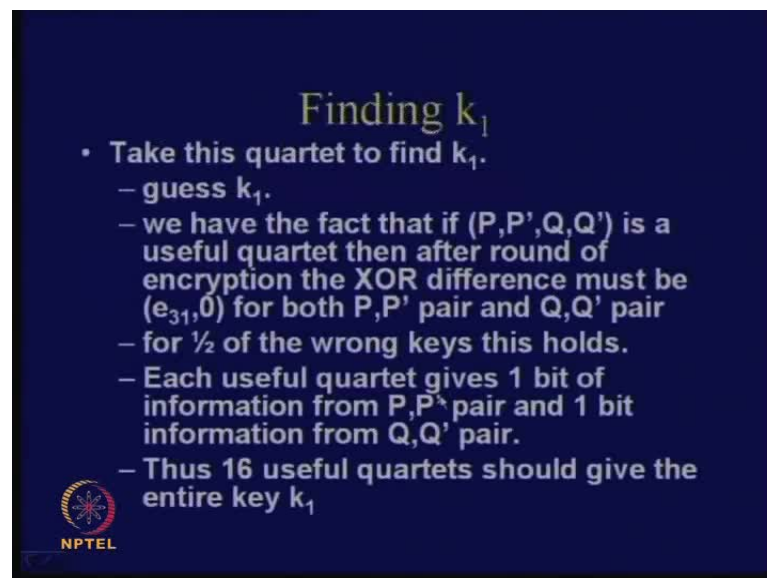
This should give you one useful quartet; that is the idea. This follows from the fact that if there is an experiment whose probability is  $p$  then we can actually prove that you repeat that experiment for say  $1$  by  $p$  number of times, then you obtain success. You can

actually prove this fact. Therefore, there is an experiment whose probability is  $p$  and if you keep on repeating that experiment for say  $1/p$  number of times then you should get at least one success. The expected number of trials to get one success is actually  $1/p$ .

So therefore, here if my probability is  $1/950$ , I should repeat this for how many times - 950 times and in each of those trials, there are actually 4 queries. So, in total there are 950 to 4 queries. If I do so many queries, I should get one useful quartet. Useful quartet means what? I get back the desired differential for  $Q$  and  $Q'$ . It satisfies this relation; that is the success of my experiment.


So, thus with around sixteen into so, if I repeat this I will I mean this is just the empirical value. If I repeat this for 16, I will explain why 16, that is if I repeat this experiment for 16 into 3800 queries then you should get back 16 useful quartets because through this you have obtained one useful quartet. If you repeat this for say, 16 times, you will obtain again 16 such quartets. So, 16 useful quartets are expected and why 16? Any idea, why 16?

(Refer Slide Time: 44:59)



**Finding  $k_1$**

- Take this quartet to find  $k_1$ .
  - guess  $k_1$ .
  - we have the fact that if  $(P, P', Q, Q')$  is a useful quartet then after round of encryption the XOR difference must be  $(e_{31}, 0)$  for both  $P, P'$  pair and  $Q, Q'$  pair
  - for  $1/2$  of the wrong keys this holds.
  - Each useful quartet gives 1 bit of information from  $P, P'$  pair and 1 bit information from  $Q, Q'$  pair.
  - Thus 16 useful quartets should give the entire key  $k_1$

 NPTEL

So, this should be clear, if we observe what I am trying to obtain. First, let us try to obtain the  $k_1$  key. So, what you do is this, that is, you take one useful quartet, concentrate on one useful quartet and you guess the value of  $k_1$ . So, we have the fact that if  $P, P', Q, Q'$  forms a useful quartet, then after one round of encryption, the

exact difference must be  $e_{31} \oplus 0$ . That you can see again using similar techniques as I told you, doing that plus 11, we can actually prove that this will be equal to  $e_{31} \oplus 0$ , for both P, P dash pairs and Q, Q dash pairs.

(Refer Slide Time: 45:41) So, maybe you can observe, if you study this. That is, what you are doing here is that you are taking a differential of  $e_{10}$  and  $e_{31}$ . If you take  $e_{10}$  and  $e_{31}$  here and if you apply  $e_{10}$  and  $e_{31}$  here, what do you expect here?  $31$  plus  $11$ . If you take mod  $31$ , how much did you get? You get back  $10$ . So  $e_{10}$  cancels with  $e_1$ . You obtain here  $0$  and this particular  $e_{31}$  comes to the left. So, you are expecting after one round,  $e_{31} \oplus 0$ , if it is a useful quartet.

So, therefore, that is what we have written here particularly. You are expecting after one round,  $e_{31} \oplus 0$  for both P, P dash and similarly, you can show it for Q, Q dash here as well. For wrong keys, actually half of the points, it will hold and half of the points, it will not hold. **What does it mean but,** For the actual key, large number of cases you will find that actually, I mean it will always get satisfied for the actual key.

What does it mean? It means that you have got something, some event with the probability of half. Which means what? That is, information is equivalent to one bit. I am trying to estimate how many trials I require. You see that if I tell you that there is a particular coin tossing experiment and you know that probability for unbiased coin; the probability of head is half. So, what does it mean? The fact that the head falls, carries one bit of information. Now, what I am saying is that for a random key guess, you have got half probability that this particular event will hold and not hold.

The fact that you are able to obtain rather I tell you that this particular thing holds carries a half probability. That means it carries one bit of information. Therefore, you should be able to get 1 bit of  $k^{-1}$  through this information. That means how many keys do you have? Now, you have P, P dash, Q, Q dash each of them reveals 1 bit of information - equivalent to 1 bit of information; that is, 2 bits of information and how many bits are there in  $k^{-1}$ ? There are 32 bits. Therefore, how many trials do you required to do? You are required to do 16 trials.

(Refer Slide Time: 48:14)

The slide has a dark blue background with yellow and white text. At the top, the title 'Obtaining other keys' is written in yellow. Below it, a bullet point in white says 'Similarly, we obtain'. This is followed by two lines of XOR operations:  $k_1, k_1 \oplus k_3, k_1 \oplus k_3 \oplus k_4, k_1 \oplus k_4,$  and  $k_2, k_2 \oplus k_3, k_2 \oplus k_3 \oplus k_4, k_2 \oplus k_4.$  Below these, white text states 'This helps to obtain the entire 128 bits of the key.' At the bottom, white text says 'Complexity of the attack is around  $2^{16}$ .' In the bottom left corner, there is a small circular logo with a star and the text 'NPTEL' below it.

Therefore, if you do 16 trials, you should be able to obtain the entire 32 bit keys. So, this is the statistical idea, statistical guess. Therefore, the idea is that if you do 16 useful quartets, you should be able to obtain the entire key  $k_1$ ; that is the idea. Similarly, you can obtain the entire key that is the entire 128 bit key and the complexity of this attack. Let us not going into the details, but this is the idea and let us try to at least understand that why boomerang attack works whereas, normal differential attacks fail. The complexity of this attack was around  $2^{16}$  because you are guessing **the sixteen I mean you are guessing the**  $k_1$ . So, it is actually not  $2^{16}$ , it is actually  $2^{32}$  because you are guessing  $k_1$ . So, what is the size of  $k_1$ ? 32, therefore, this should be actually  $2^{32}$ .


(Refer Slide Time: 48:55)

## Square attacks on 4 round AES

- Let  $\Lambda$  be an active set of 256 states, that are all different in some of the state bytes and are all equal in the other state bytes.

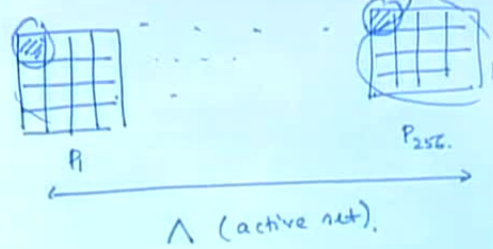
$$\forall x, y \in \begin{cases} x_{i,j} \neq y_{i,j} & \text{if } (i,j) \text{ active} \\ x_{i,j} = y_{i,j} & \end{cases}$$

Since the bytes of a  $\Lambda$  set are either constant or takes all possible values,


$$\bigoplus_{x \in \Lambda} x_{i,j} = 0, \forall i, j$$


(Refer Slide Time: 49:08)

AES



Square Attacks



I will just conclude our talk with another kind of interesting property. I mean we can actually study this. We will again come back to AES. We have talked about feistel ciphers and we will just again conclude with AES. What was the idea of AES? Let us try to see how disturbance works in AES. I think I have already told you once in a class; that is, let us disturb one particular byte of AES. If I disturb one byte of AES, **how many such cipher text can I obtain 256 ciphers** how many plaintexts can I obtain? 256 plaintexts.

**So therefore, if I call this as an sort of** Therefore, I can keep on disturbing these bytes and I can obtain say P 1 to P 256 corresponding values of plaintext. I call this set as

something which I call as an active set. So, I call this as an active set. These are various inputs that you have taken.

(Refer Slide Time: 50:16) Therefore, coming back to the definition, you see that let this be an active set of 256 states that are all different in some of the state bytes and are all equal in the other state bytes. So, I call such kind of set as active sets. It means that certain byte locations are active and certain byte locations are not active. So, active means for all them you are essentially, they are different, but for the other locations they are same. So, again like in our diagram you have seen that for these locations, all of these things have got the same values, but these values are all distinct. So, all of them are distinct values.

So, what is the property of this active set? If you add them, you get 0. You get 0 because these are all same, but what about this? This takes all possible values. If you add them, you will get back 0. You can see that this property in AES remains invariant till 3 rounds. I mean, if I impose or rather give plaintext of this nature, then this property does not get disturbed for 3 rounds of AES.

(Refer Slide Time: 51:43)


**Square attacks on 4 round AES**

- Let  $\Lambda$  be an **active set of 256 states**, that are all different in some of the state bytes and are all equal in the other state bytes.

$$\forall x, y \in \begin{cases} x_{i,j} \neq y_{i,j} & \text{if } (i,j) \text{ active} \\ x_{i,j} = y_{i,j} & \end{cases}$$

Since the bytes of a  $\Lambda$  set are either constant or takes all possible values,

$$\bigoplus_{x \in \Lambda} x_{i,j} = 0, \forall i, j$$

 NPTEL




So, that is the basic idea behind an attack which was called as square attacks. So, you have an invariance that if you add them, you get 0; that does not change for 3 rounds of AES. So, if you take x-or of these things, you would get 0, for all possible values of  $i$  and  $j$ . So, that is the basic idea, which does not change.

(Refer Slide Time: 51:50)

### Invariance of the active set

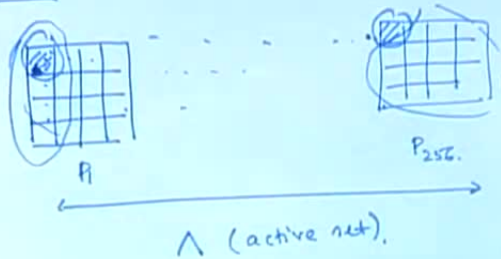
- Consider a  $\Lambda$  set in which only one byte is active.
- Lets observe the propagation of the active set through 3 AES rounds.
- SubBytes, AddRound keys does not alter the property of active set.
- ShiftRow transposes the active byte position.
- The column in which there is one active byte, because of the linear transformations with invertible coefficients, there is one column with 4 active bytes.



Let us see the invariance, why the invariance? I mean, it is quite easy actually. What you can do is that you consider this set in which only one byte is active and then what you do is that you observe the propagation of the active set through 3 AES rounds.


(Refer Slide Time: 52:24)

### AES



$\Lambda$  (active set).

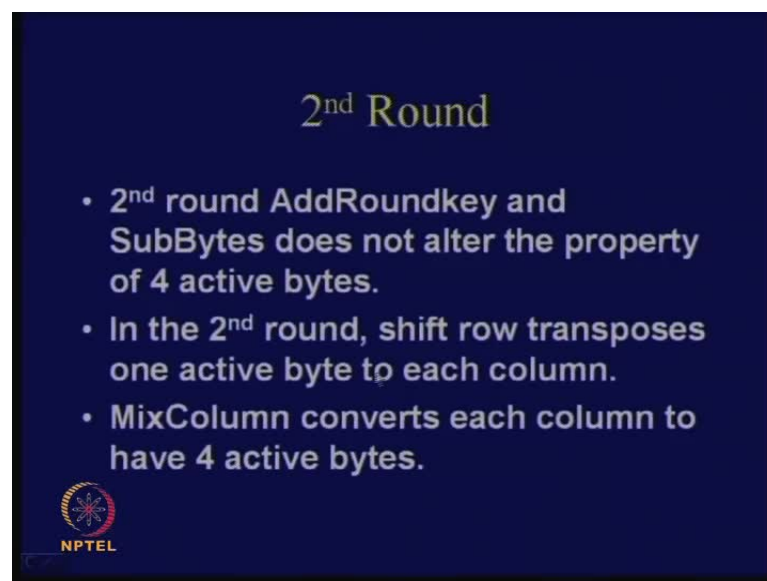
Square Attacks



What is the first layer? The first layer is sub byte. What you do in a sub byte? You are basically considering a one to one map. Therefore, if you observe sub byte, you take one particular byte value here and all of these values are distinct; all these values are distinct. Therefore, if you do a sub byte operation after this or an add round key operation here, then essentially, we will find a one to one map for every byte location. Therefore, in the output of this also, you will get all possible values. So, the property does not change because of the sub byte or the add round key.


What about the shift row? The shift row will just transpose this; it will not change. What about the next column? You see that in the next column, if in one particular column, there is one disturbed byte, then in the output of the next column, all the 4 bytes get disturbed. Therefore, you again obtain an active set, but only thing is that the disturbed byte propagates to all the 4 rows of one particular column.

(Refer Slide Time: 53:26)



**2<sup>nd</sup> Round**

- 2<sup>nd</sup> round AddRoundkey and SubBytes does not alter the property of 4 active bytes.
- In the 2<sup>nd</sup> round, shift row transposes one active byte to each column.
- MixColumn converts each column to have 4 active bytes.


  
NPTEL

So, that is the idea. Therefore, now if you come if you see this if you see that the column in which there is Therefore, this is what would be the justification why it does not change and why it remains in invariance. For the second round, the second round add round key and a sub bytes also does not alter the property of 4 active bytes. In the second round, the shift row transposes one active byte to each column and the mix column converts each column to have 4 active bytes. So, this is if you have solved the example of diffusion, which I told you to study for AES, this should become quite obvious.

(Refer Slide Time: 53:51)

### 3<sup>rd</sup> Round


- 3<sup>rd</sup> round AddRoundkey and SubBytes does not alter the property of 4 active bytes per column.
- ShiftRow merely transposes.



(Refer Slide Time: 54:15)

### 3<sup>rd</sup> Round

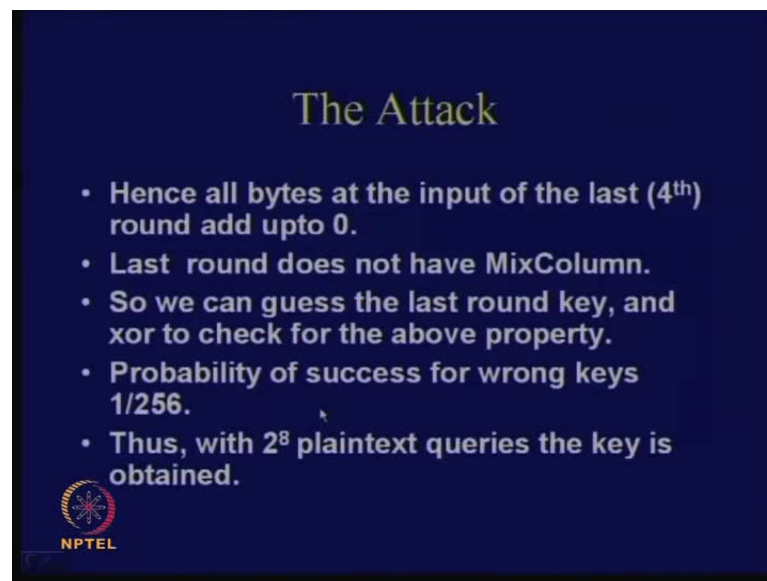
If the input be denoted by  $a$  and the outputs by  $b$ :

$$\begin{aligned}\therefore \oplus b_{i,j} &= \oplus \text{MixColumn}(a_{i,j}) \\ &= \oplus (02.a_{i,j} \oplus 03.a_{i+1,j} \oplus a_{i+2,j} \oplus a_{i+3,j}) \\ &= (02 \oplus a_{i,j}) \oplus (03 \oplus a_{i+1,j}) \oplus a_{i+2,j} \oplus a_{i+3,j} \\ &= 0\end{aligned}$$


What about the third round key? You will see that in the third round, you essentially have the same property which holds, but now you essentially have the entire state matrix which is disturbed. Therefore, all the  $i, j$  locations are active now. Therefore, now I mean, predicting beyond this, that is the third round mix column becomes hard. That is a little bit tricky, but you will see that this also follows, if you observe the matrix of mix column.

So this you can see. Although it does not remain an active set, still the property that if you sum, then you get 0 still holds. That means you do what you are doing that x-oring the column of a  $i j$  and remember that the mix-column was a linear transformation. Therefore, you can now bring out this in this fashion. **and you can show that actually** So, these are all x-ors. This is not 02 x-ored with a  $i j$ , it means 02 multiplied with x-or of all a  $i j$ s. Maybe, I could have written the bracket here, rather than written here. Similarly, for this also, this 03 multiplied with x-or of a  $i$  plus 1  $j$  and here also; this is simple.

(Refer Slide Time: 55:11)



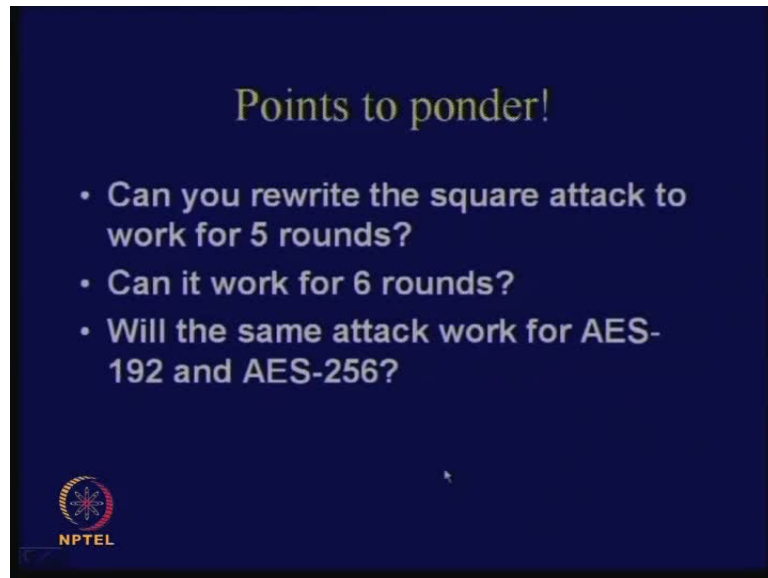
The Attack

- Hence all bytes at the input of the last (4<sup>th</sup>) round add upto 0.
- Last round does not have MixColumn.
- So we can guess the last round key, and xor to check for the above property.
- Probability of success for wrong keys 1/256.
- Thus, with  $2^8$  plaintext queries the key is obtained.

NPTEL


Therefore, you x-or them to get back 0. **Therefore, this property does** I mean if you add all the byte locations, the fact that you get back 0 does not change because of the third round mix column also, but the sub byte will change it actually. You know that in the last round, if you consider a 4 round AES for example, then you know in the last round, there is no mix column. Just consider a very simple, smaller portion or smaller variant of the AES, where the last round does not have the mix column. What you do is that you just guess the key, you go back to the sub byte layer and you check whether the input of the sub byte layer if you add all of them, whether you get 0 for all the  $i j$  locations.

(Refer Slide Time: 55:57)



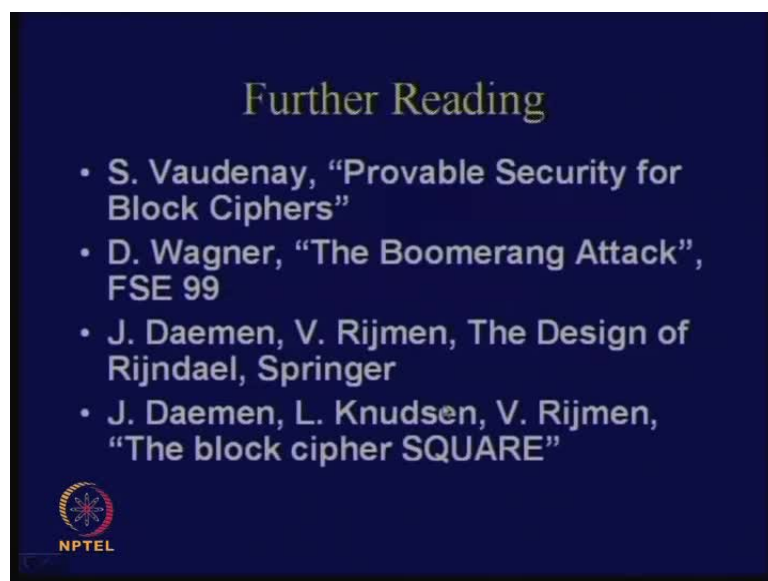
**Points to ponder!**

- Can you rewrite the square attack to work for 5 rounds?
- Can it work for 6 rounds?
- Will the same attack work for AES-192 and AES-256?

 NPTEL


If it is a correct key then it should hold; if it is a wrong key it should not hold. The probability of this fact is 1 by 256. Again, you know that if you repeat this for 256 number of trials, you should be able to identify the key. That is the basic idea behind this attack, but you can just ponder upon certain points. What we have discussed in this basically is a square attack on 4 round of AES. What you can just think is whether the square attack will work for 5 rounds, will it work for 6 rounds of the AES and will the same attack work for AES 192 or an AES 256.

(Refer Slide Time: 56:22)



**Further Reading**

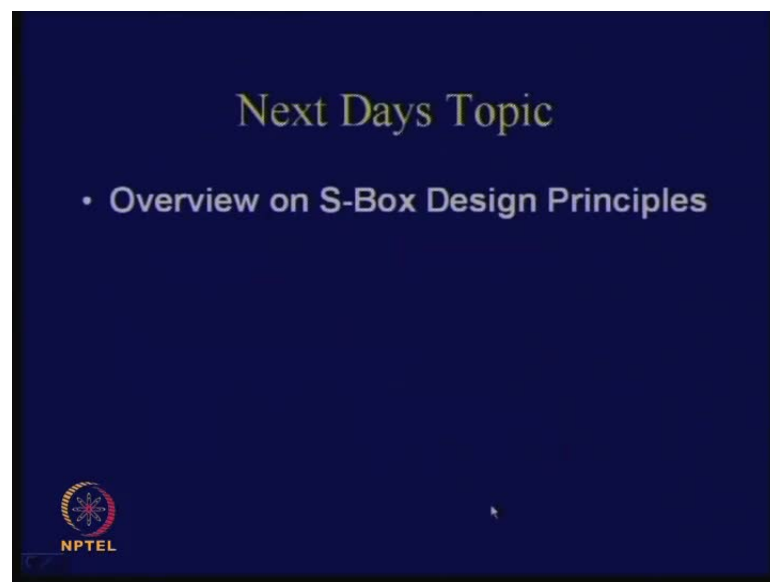
- S. Vaudenay, "Provable Security for Block Ciphers"
- D. Wagner, "The Boomerang Attack", FSE 99
- J. Daemen, V. Rijmen, The Design of Rijndael, Springer
- J. Daemen, L. Knudsen, V. Rijmen, "The block cipher SQUARE"

 NPTEL

So you can just think on these problems. The references that I have followed are “Provable security for block ciphers” by S Vaudenay. These are freely downloadable; you can download them. The boomerang attack by FSE, published by FSE 99 by David Wagner and also, this book is not really available. You can follow this particular work, the block cipher.

So this was written by Daemen Knudsen and Rijmen. This is also freely available. This SQUARE cipher was actually an ancestor of the AES; somewhat similar kind of idea exists there also. The designers themselves gave this attack and therefore, the name square attack actually. This was originally proposed for square, but it works for AES as well.

(Refer Slide Time: 57:11)



So, the next day's topic we will discuss. You see that it is quite difficult to design a block cipher. It is not so easy to make it secure and the center idea is on the S-boxes. So, we will again concentrate on the overview of S-box design principles and discuss on that.