

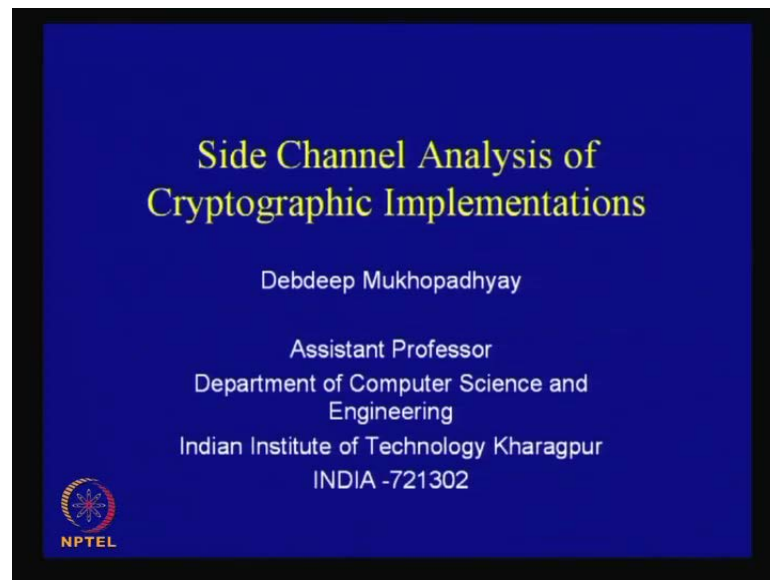
Cryptography and Network Security
Prof. D. Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No. # 01

Lecture No. # 41

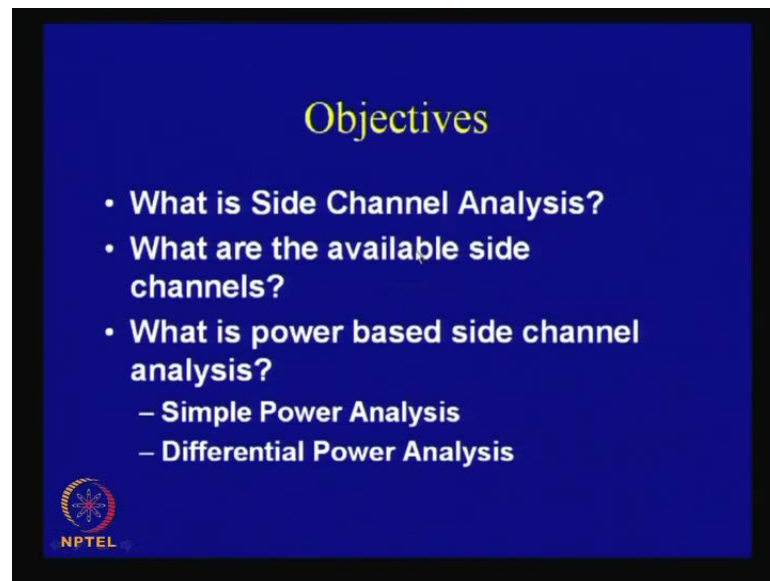
Side Channel Analysis of Cryptographic Implementations

(Refer Slide Time: 00:24)



So, today, we shall discuss about a topic, which is called as side channel analysis of cryptographic implementations. So, as we have seen in the previous discussions **about crypt**, when we discussed about cryptographic algorithms, we have actually **mean bothered** about the mathematical security of the algorithms. But this is the first time, when we would actually see that you cannot not only target the algorithms, but you can also target the implementations. So, which means that your implementations also need to be taken, I mean, has to be done in a proper way.

(Refer Slide Time: 01:03)



The slide has a dark blue background with a black border. The title 'Objectives' is centered at the top in a yellow serif font. Below it is a bulleted list of white text. The first three items are main bullet points, and the last two are sub-bullet points. In the bottom left corner, there is a small circular logo with a red and blue design and the text 'NPTEL' below it.

Objectives

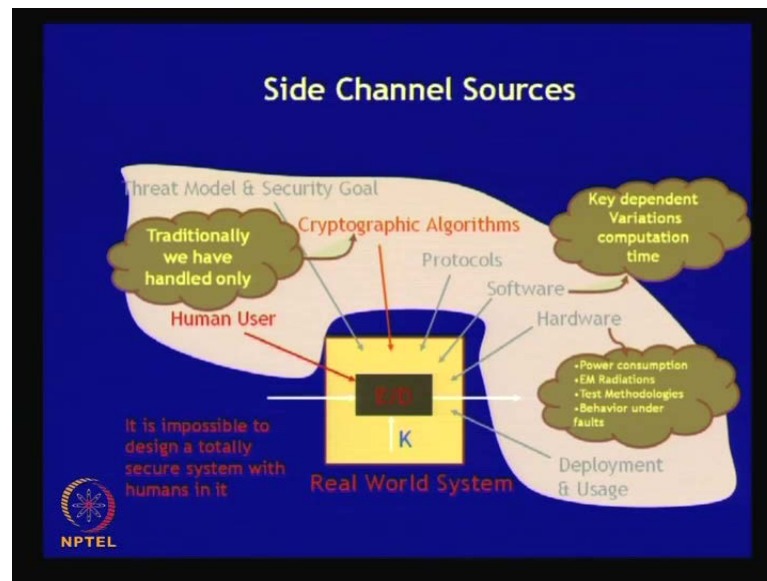
- What is Side Channel Analysis?
- What are the available side channels?
- What is power based side channel analysis?
 - Simple Power Analysis
 - Differential Power Analysis

NPTEL

So, this is the very important field of study right now and without it, discussion on security and cryptographic algorithms is not complete. So, hence, let us see some of the **internal issues** some of the important issues. So, we will first consider and understand what is mean by side channel analysis and **why** how does it kind of, rather where does it stand in the conventional cryptographic model **and...** So, we shall actually discuss about what are the available side channels.

And we would actually focus about something which is known as powered based side channel analysis, which actually targets the power consumption of the device. And I discussed about two important things, which are three different analysis techniques, which is known as simple power analysis and most importantly the differential power analysis. So, this will give us a feel about how to perform side channel analysis of cryptographic algorithms.

(Refer Slide Time: 01:37)



So, to start with, there are several sources of side channels in the real world. So, therefore, as we have seen previously, that is, the cryptographic algorithm; we also have the threat models and the security goal. So, we have the corresponding protocols; we have the corresponding software; we have the hardware; so we have the deployment and the usage; we have got the human user; so there are so many factors which are involved in a real world cryptographic system.

Now, we will say that, **it is** although we know that with humans it is impossible to design totally secured systems, but traditionally we have actually try to handle cryptographic algorithms. So, we have been bothered about what is the security of that cryptographic algorithm and we have tried to develop our mathematical principles and design strategies only concentrating on the cryptographic algorithms.

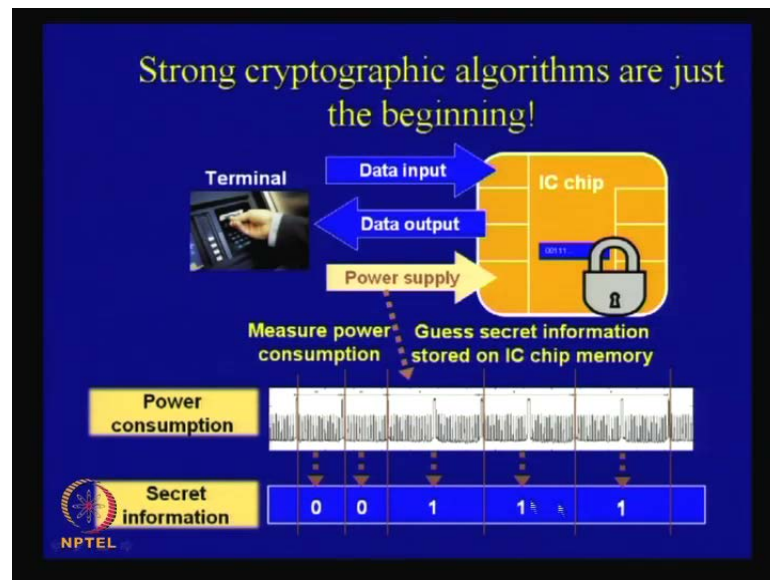
But however, there are certain other factors like the software's **or** in context when you finally take a cryptographic algorithm and you have to finally implement it. So, therefore, you have to implement it either in a piece of software, that means, you have to write a piece of C code or any software code or you have to basically develop corresponding hardware designs; so you have to shift it into a hardware.

So, this actually comes out with these properties like, there can be a key dependent variation in the computational time or **there could be...** and when you are talking about

hardware's the power consumption, the electromagnetic radiations, the test methodologies invert, the behavior of it under faults. So, what we will see is that, in this side channel analysis **or there are** the side channel analysis tries to exploit these features, which are there in the corresponding implementation.

So, you **to may** take a perfectly **secured mathematically** secured cryptographic algorithm, implement it either in as a software or a hardware, but all though they are mathematically secured, these properties of the intimations, that is, the variation in time, the variation in power, the radiations, the testing techniques, the behaviour of the under faults can actually meet the secret.

(Refer Slide Time: 03:51)



So, this is actually a threat, because a very big threat, because finally the n to n security is again compromised. So, there is one more example, motivating example. So, we take a smartcard and we swap that, therefore imagine like maybe an algorithm like RSA which is the very strong and mathematically secured algorithm is actually protecting my inherent key and we take this smartcard and access it in a card accepting device.

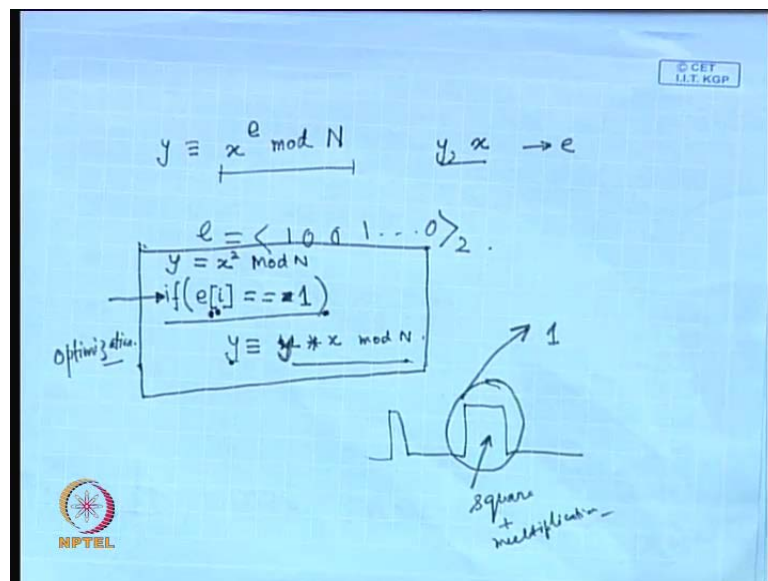
So, while they exchange of the data input and the data output, traditionally as we have considered is the plain text and this is the cipher text; this does not leak any information about the key and therefore we may be very happy, very confident. But on the other

hand, there could be some side channel information's like, may be the power which is been consumed.

So, here is an example like for RSA algorithm and we see that these are the power profiles, which have been consumed by the smartcard, which is actually a piece of hardware.

So, therefore, it could be either a piece of hardware or it could be, maybe a microcontroller, which is again software. But then you have got this kind of power consumptions, which are involved and we can observe that there can be some narrow peaks and there can be some broad peaks, which actually can be exploded for attacks. So, therefore, we will see that.

(Refer Slide Time: 05:03)



So, if you remember the conventional RSA algorithm. So, in the conventional RSA algorithm, we have got a public key e . So, therefore, when we perform this operation, then for example, perform this operation say x power of e mod N or mod of a natural number. So, typically how **do** you do this operation, is by raising this to the power of e you can actually engage an algorithm, which is typically known as the square or multiply algorithm.

So, therefore, this e could be a public quantity; this e could be a secret quantity also or a private quantity depending upon the context; depending upon, whether we are using it

for encryption or whether we are using it for verification and things like that or whether we are using for decryption. But what I am trying to say is that, mathematically we know that, if I give you this information of y and if I give you this information of x and if I ask you like, what is the knowledge of e , rather we know that it could be little bit mathematically challenging to find it.

On the other hand, let us consider as simple algorithm, which we know as the square or multiply algorithm. So, what we do in the square and multiply algorithm is, typically we take e and we actually write them in a binary format. So, therefore, imagine like, there is a binary format like this **mean**; so that is just an example.

So, we know that in this algorithm, the algorithm if you remember, which we previously studied in context to RSA was actually trying to do an alternative operation. So, therefore, the algorithm what it does is that, it checks for the e_i i th bit of a key and checks that, if e_i is equal to 1, then it does some extra operation of multiplying. So, what it does is that, it takes this y and it **multiplies it with...** So, the algorithm is that, the first step is you perform the squaring operation, you perform like $x^2 \bmod N$ and so you are always performing the squaring operation, if your bit is 1, then you not only perform the squaring operation, but you also perform a multiplication operation, that is, you take this y and you multiply it with x .

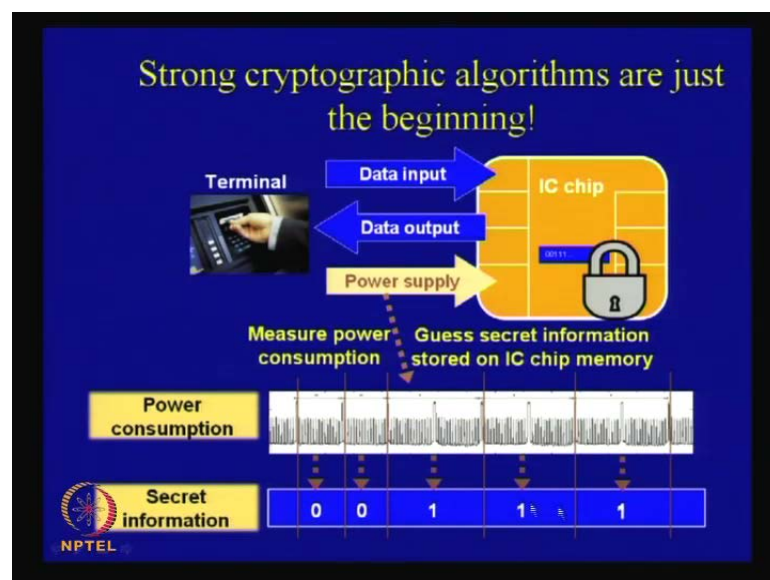
So, remember, whenever you are doing this multiplication, you are always doing, may be this module N operation, but that is not important; we are not actually so much interested about the mathematical prissiness of this algorithm right now, but understand that these are the broad steps and observe that, if there is an if statement provided here, this statement if is provided for optimization purposes. Because we want that, if this is 1 only, then this multiplication operation takes place; otherwise, it is not important; it is not necessary to perform a multiplication operation. So, we want to make our computations fast and more optimized.

But on the other hand, what happens is that, when you are talking about side channels, is that, because if this bit is 1, if i th bit is 1, you are performing an extra operation. So, if you perform the power consumption and every regular intervals of the clock, then you will find that, **whenever there is an extra...** So, if you observe the peaks in the power, if

you find that there is broad peak, then you know that this corresponds to the square plus the multiplication step.

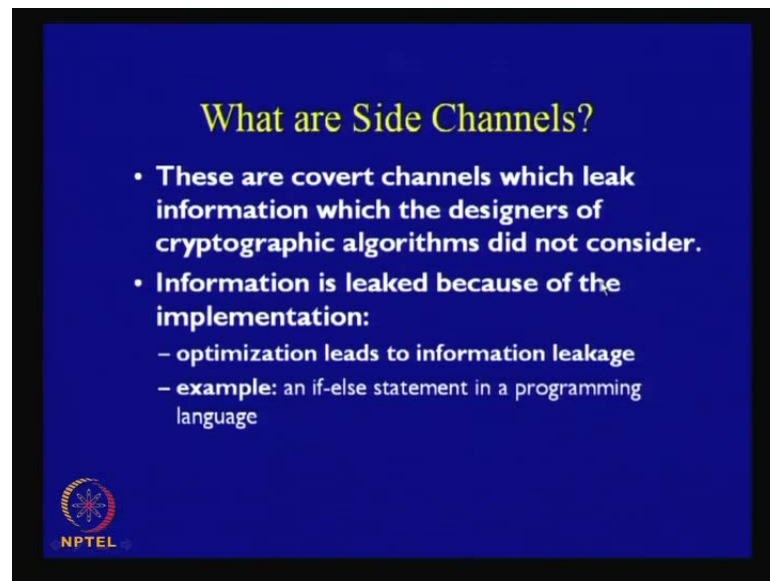
So, this implies that, the bit that it is processing is actually 1. So, we see that this extra power consumption is giving you information about the i th bit of i . Now, without this i th bit of i is a public quantity or a private quantity, I am not really bothered about, but what I am saying is that, ideally this y should not have leaking any information about this value of e mathematically. But on the other hand, here we see that, because of this power consumption, which was nowhere in the picture when we actually thought of cryptographic algorithms is actually leaking the information of the i th bit. So, this is called as side channel analysis. So, this is the basic idea behind side channel analysis.

(Refer Slide Time: 09:25)



So, again coming back to this particular picture you see that, because of this power consumption it may happen that the secret information is leaked. Because if you see that, there are some broad spikes, then you know that, this is 1, your **infra-red** is 1 and if there are some narrow spikes, you **infra-bid** is 0. So, therefore, what may happen is that, although you are using a perfectly strong cryptographic algorithm, it may leak the information of the secret.

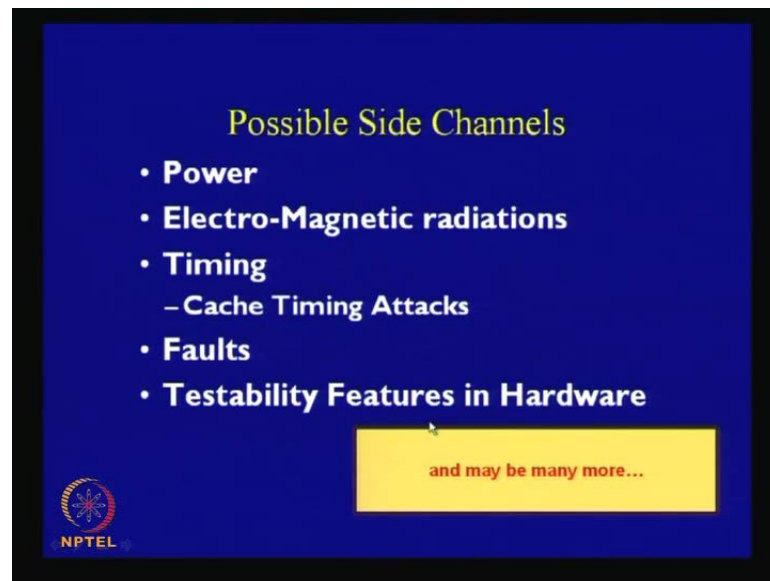
(Refer Slide Time: 09:50)



Now, therefore, side channels are actually covert channels, which leak information, which the designers of the cryptographic algorithm actually did not consider. So, therefore, the idea of the problem is not with the algorithm designer, because the algorithm designer did not actually think about these kinds of leakages,

So, therefore, information is leaked because of the implementation and this you can take as a **thumb rule** that, whenever we perform optimizations, this will actually lead to information leakage. So, optimizations are responsible or needed definitely, because we want to make our cryptographic algorithms efficient; we want to make our cryptographic implementations efficient, but **if** this also comes with the audit cost of information leakage.

(Refer Slide Time: 10:40)



Now, an if-else statement in a programming language therefore, could be a typical example, which can be subjected to side channel analysis. Now, what are the possible side channels? The possible side channels could be power electromagnetic radiations, it could be timing analysis like, cache timing attacks, faults and testability features in hardware and may be many more. So, we are not actually going to discuss about all of these, but we will actually concentrate about the first side channel, which is called as power and is probably one of the most menacing side channel **side channels** which exist in the present day scenario.


(Refer Slide Time: 11:08)

Possible Side Channels (contd.)

- **Power**
- **Electro-Magnetic radiations**
- **Timing**
 - Cache Timing Attacks
- **Faults**
- **Testability Features in Hardwares**

Power Attacks *Our Focus*

Underlying Idea:
Information leaked by these side channels can give useful information about the secret key


 NPTEL

So, you consider on power attacks and we will try to see how essentially power attacks can be performed.

(Refer Slide Time: 11:13)

Power Attacks (PA)

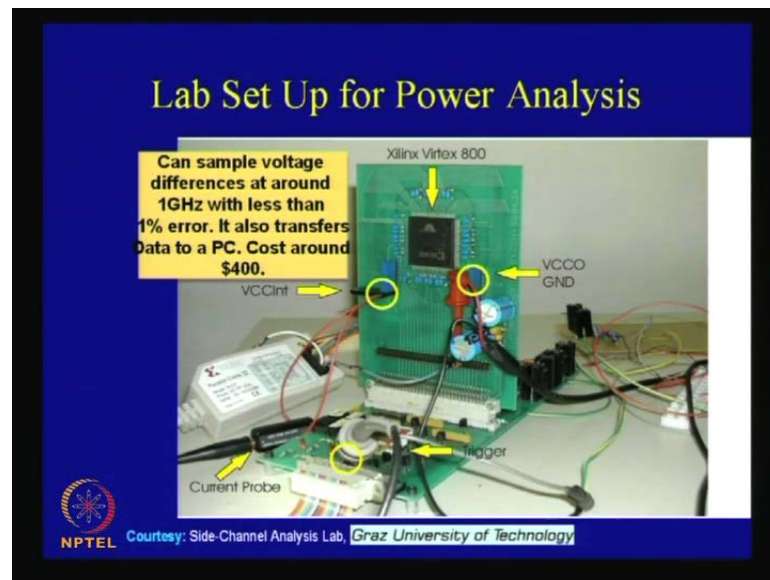
- During the last few years lot of research has been conducted on **Differential Power Attacks (DPA)**
- Exploit the fact that (dynamic) power consumption of chip is correlated to intermediate results of the algorithm
- To measure a ckt's power, a small resistor (50 ohm) is inserted in series with the power or ground input

 NPTEL

Now, during the last years, a lot of research has been conducted on differential power attacks, which exploit the fact that, dynamic power consumption of chip is correlated to the intermediate results of the algorithm, **that is,... and we will see that like...** So, what we do generally is that, **we take** so **the idea of when we do** when we do an experiment is like, **we** you basically take a cryptographic algorithm and implement it in a piece of

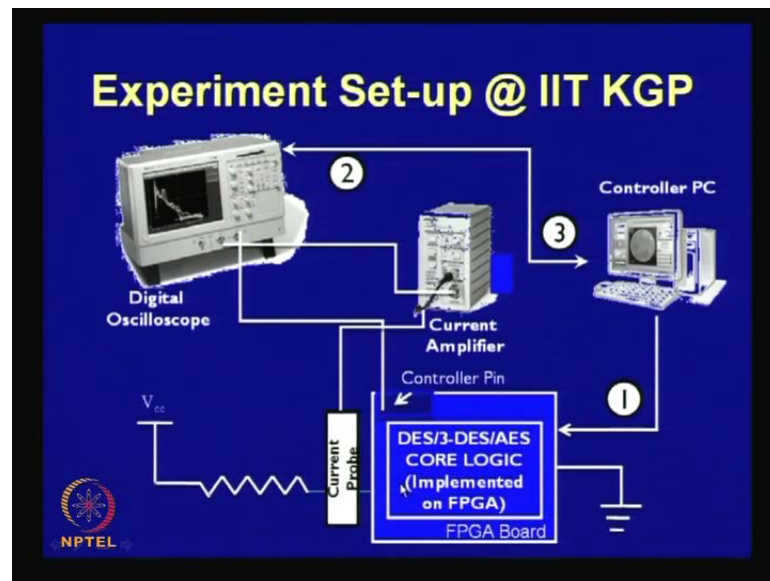
hardware and in order to measure the circuits power, we have a small resistor, which we insert in series with the power of the ground input.

(Refer Slide Time: 11:50)



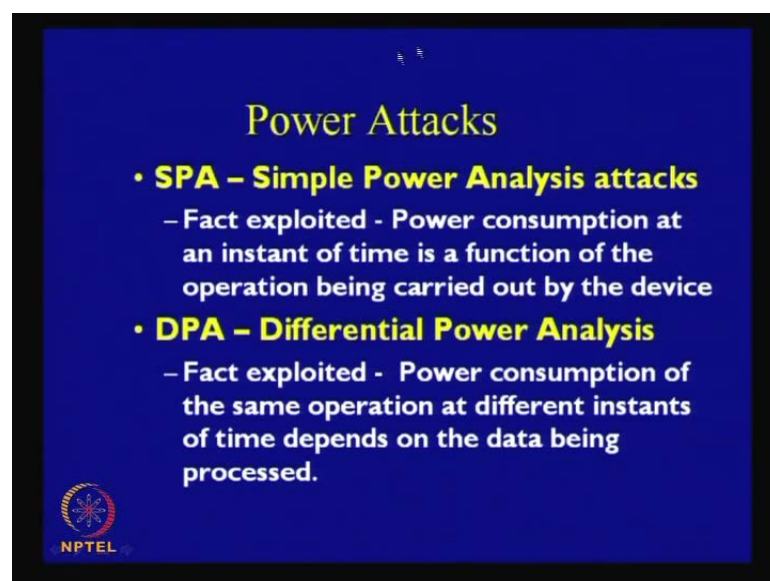
So, here is a diagram of a laboratory set up of **glass** and here, you can see that, what has been done is that, there is a resistor, which has been involved or rather which has been inserted in series with either the VDD line or in the ground line and there is a current probe, which is quite sensitive current probe, which is actually used to monitor the corresponding current, which actually gives the measure of the power, which is being consumed by the device. So, this is a quite low cost set up; we actually also have got a laboratory set up at **all** place.

(Refer Slide Time: 12:20)



You can see that, you have got a **a** may be a FPGA implementation of the inherent algorithms like DES, 3-DES and AES and there is a digital oscilloscope; there is a corresponding current amplifier and there is also a corresponding PC and there is a kind of communication, which goes on among these entities and there is a current probe, which actually monitors the current, which actually is passing through this resistor, which inserted in series with VCC and the FPGA board. So, these are typical laboratory set up.

(Refer Slide Time: 13:00)



Now, we shall discuss about the simple power analysis attacks and the differential power analysis attacks. The first thing is note what are the facts exploited in a simple power analysis. In a simple power analysis, the power consumptions at an instant of time is actually a function of the operation being carried out by the device like, as we have seen in context to this square and multiply algorithm, what we are doing is simple power analysis. Because the power that is being consumed, that is, the extra spike is a function of the operation, which is being carried out, that is, **whether it is the function of** whether you are doing the multiplication or you are not doing the multiplication operation. So that is an example of a simple power analysis. So, typically we will see that in a simple power analysis, the number of power process required are actually quite less or minimal.

But as opposed in a differential power analysis, the power consumption after the same operation at different instants of time depends on the data being processed. So, it is actually like **that** a power which is consumed by the same operation at different instance of time, actually is proportional to the actual data which is being processed.

(Refer Slide Time: 14:05)



Simple Power Analysis (SPA)

- **Directly interprets the power consumption of the device**
- **Looks for the operations taking place and also the key!**
- **Trace: A set of power consumptions across a cryptographic process**
- **1 millisecond operation sampled at 5MHz yield a trace with 5000 points**

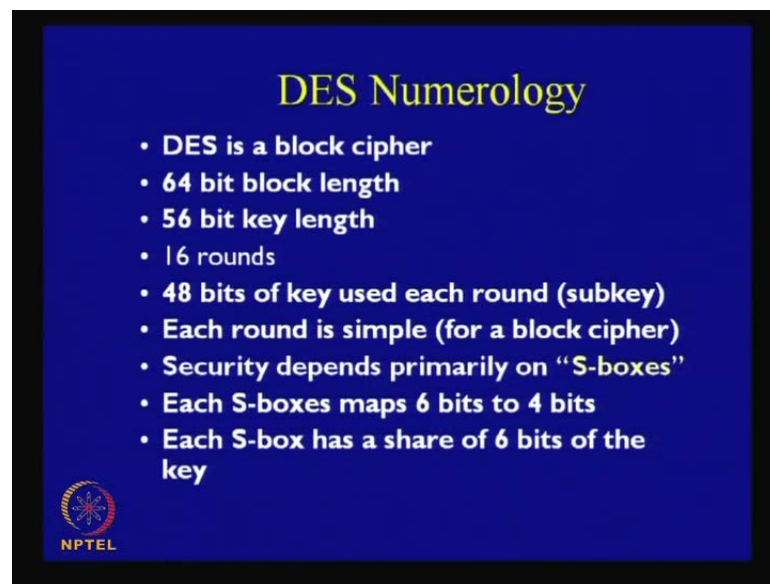
 NPTEL

Now, in a simple power analysis, it directly interprets the power consumption of the device; it looks for the operation taking place and also for the key. So, therefore, often we will find that, if we are able to do a simple power analysis may be a 1 trace or may be a 2 or 3 power traces can actually leak the key. So, therefore, what is the trace? A trace is a set of power consumptions across a cryptographic process. So, therefore, we actually

sample the corresponding power consumption at regular intervals and you obtain a power profile.


So that is the power consumption consumed by a particular device for an input and therefore, often it may happen that **when you are doing** when you are successful with simple power analysis, then a single power analysis or a power profile leaks the key.

(Refer Slide Time: 14:58)



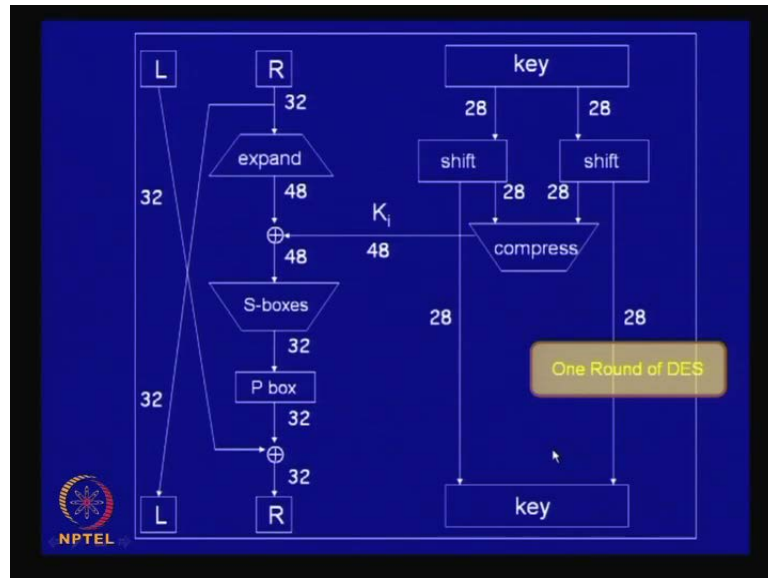
DES Numerology

- DES is a block cipher
- 64 bit block length
- 56 bit key length
- 16 rounds
- 48 bits of key used each round (subkey)
- Each round is simple (for a block cipher)
- Security depends primarily on “S-boxes”
- Each S-boxes maps 6 bits to 4 bits
- Each S-box has a share of 6 bits of the key

 NPTEL

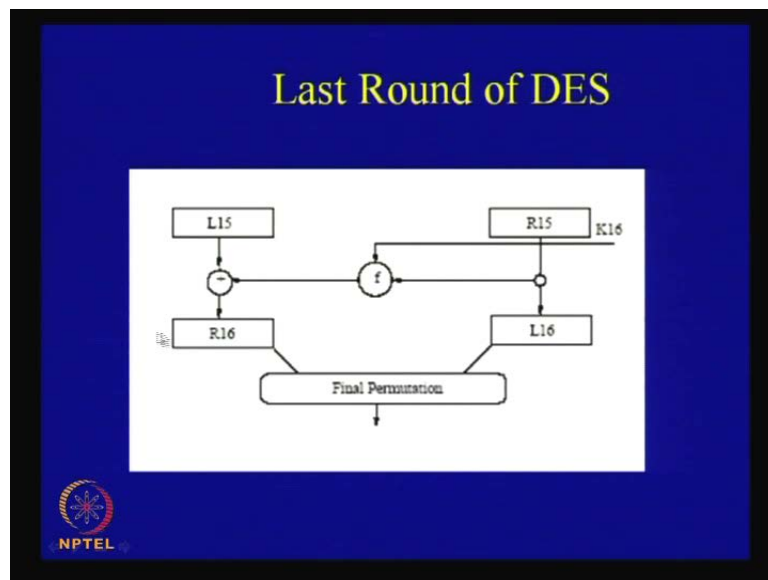
So, let us again remember back the DES; so, this something that we will target to understand the techniques. So, DES as we know is a block cipher; 64 bit block length, it has got a 56 bit key length, 16 rounds, 48 rounds of key are used for each round; each round is simple for a block cipher and this kind are iterated. So, it was a iterated block cipher that we studied and a security primarily depended on the S boxes; an each S box maps the 6 bits to 4 bits and therefore, S box has a share of 6 bits of the key, that is, each S box has got a share of 6 bits and they are in total there are 8 S boxes, 8 into 6 is 48.

(Refer Slide Time: 15:41)



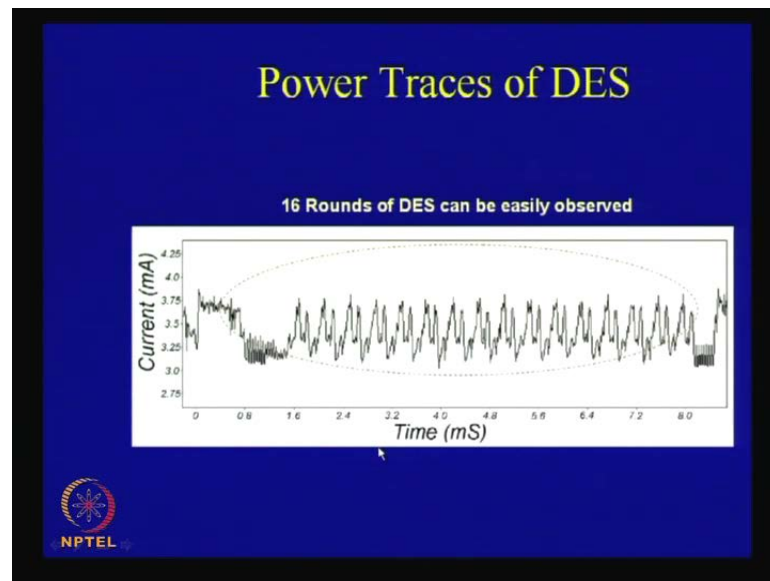
Now, this was the pictographic representation of this thing and therefore, this is the S box that is primarily responsible for the security and we will see that how we can actually target this for developing a power analysis.

(Refer Slide Time: 15:56)



So, this is the last round of the DES S box as well as last round of the DES and you can see that, this is L₁₅ and this is R₁₅, that is, the left part and the right part and this is the corresponding output, which is finally permutative and this the cipher text, which is generated by your encryption device.

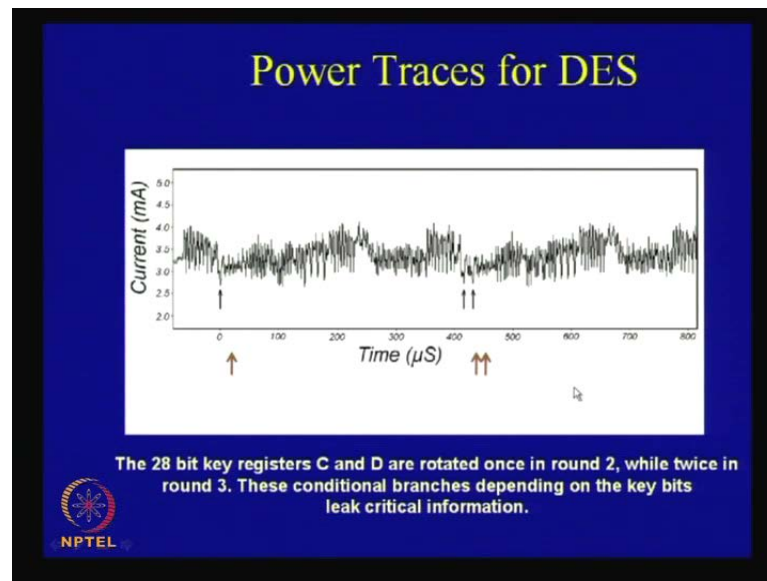
(Refer Slide Time: 16:13)



Now, the objective is like, let us see that first of all a simple power trace of this DES. So, therefore, you can see that the 16 rounds of DES can actually be observed very clearly. So, you can see that, all these **jitters** even you can actually count and find that, they are 16, so which means that you can observe the 16 rounds of DES.

So, you can see one phenomenon that, although you are not able to get the key as such as we have seen in the previous case in context to the square and multiply algorithm, but it still leak some information; it is still leaking that your DES has got 16 rounds; it is still leaking the information that your cryptographic algorithm, the nature of the cryptographic algorithm it is telling you that the nature of the cryptographic algorithm is an iterative in nature, which performs the same kind of operation over and over; it is also use telling you that, there are 16 rounds which makes DES a very potential choice of the cryptographic algorithm.

(Refer Slide Time: 17:07)



So, these are some of the inherent zoom. So, if you zoom in, you can actually obtain by the further information; you can probably obtain the S boxes, the S box operations and the inherent operations.

(Refer Slide Time: 17:18)

DPA Overview

Introduced by P. Kocher and colleagues
More powerful and more difficult to prevent than SPA
Different power consumption for different state (0 or 1)
Data collection phase and data analysis phase
Procedure:

- Gather many power consumption curves
- Assume a key value
- Divide data into two groups (0 and 1 for chosen bit)
- Calculate mean value curve of each group
- Correct key assumption → not negligible difference

The NPTEL logo is visible in the bottom left corner.

Now, we will actually see a very sophisticated technique, which was introduced by Paul Kocher and colleagues and it is more powerful **and more...** So, it was developed around 1995-96 and it is a more powerful and more difficult to prevent than SPA.

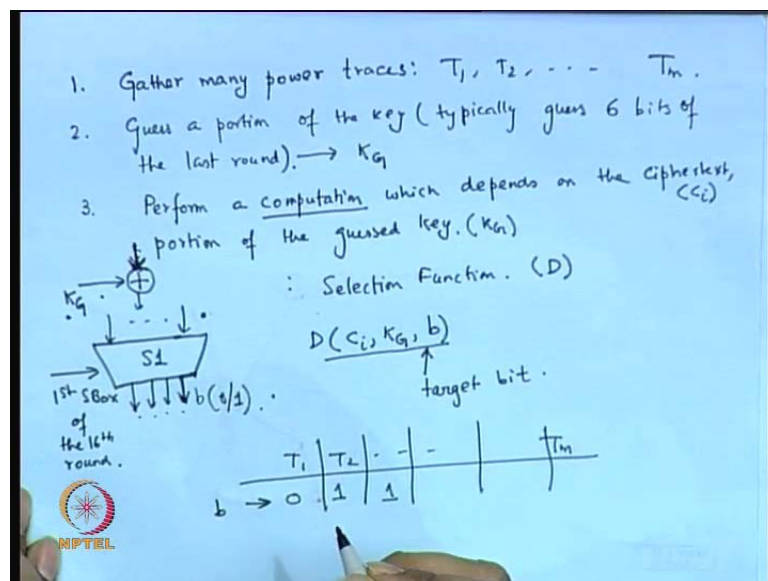
Now, what is differential power analysis based on? So, it is based on different power consumptions for different states. As we have said that, it is based on the fact, the power consumption of the device is actually proportion of to the information, which is being processed to the data **which is** on which the computation has been performed.

So, therefore, it is important to exploit this fact, in this particular statistical analysis, DPA is nothing but a statistical analysis method. So, the process here can actually be summarized as this. So, what we do is that, we gathered many power consumption curves; so this power consumption curves, the numbers could vary may be few 1000 to may be 10000 to 1 lakhs to several lakhs.

So, therefore, it is like many power consumption curves have been accumulated, and then, we actually assume a portion of the key value, and then, we divide the entire power consumptions or the power curves in the two groups, may be like one depending up on 0 and 1 depending up on a particular chosen bit.

Now, what we do here is that, we calculate the mean curve of each group and the correct key assumption, the idea is that, the correct key assumption implies that there is no negligible difference and therefore, **if you are say** that is the basic observation on which **you...** So, you find that, if there is a correct key assumption, then you will find that there is a non-individual difference in the difference of means.

(Refer Slide Time: 19:15)



So, what we do here is as follows, that is... So, if you have got for example large number of power traces, so that is the first, step that is, you gather many power traces; so your power traces, you can actually denote them may be as T_1 , T_2 and so until T_n ; so n could be a large number of power traces.

The second thing what we do is that, you guess a portion of the key. So, therefore, obviously I will not guess, may be the 56 bit key; I will guess a portion of the key. So, when we guess a portion of the key, typically we guess, when we are talking about DES, we are guessing 6 bits of the last round, then we have to do we do a kind of operation and we essentially do some computation so that computation, we perform some computations.

The computation actually depends on the ciphertext, the portion of the key, which has been guessed. the portion of the guessed key So, therefore, this could be like, your guessed key; therefore, I call this as K_G therefore, this is a function of K_G and the corresponding ciphertext which I call as C_i . So, based upon this computation... So, this computation is very technically known as computation of something, which we call as the selection function often denoted by d .

Therefore, D actually targets or rather computes on C_i and the guessed key which is called K_G and computes a target bit called b . So, b is some target bit, which is been computed. So, what is the target bit? So, for example, if you target,... So, what we do in a DPA? We target one of the last round S boxes; so this could be, may be the first S box of the last round; so this is the first S box of the sixteenth round.

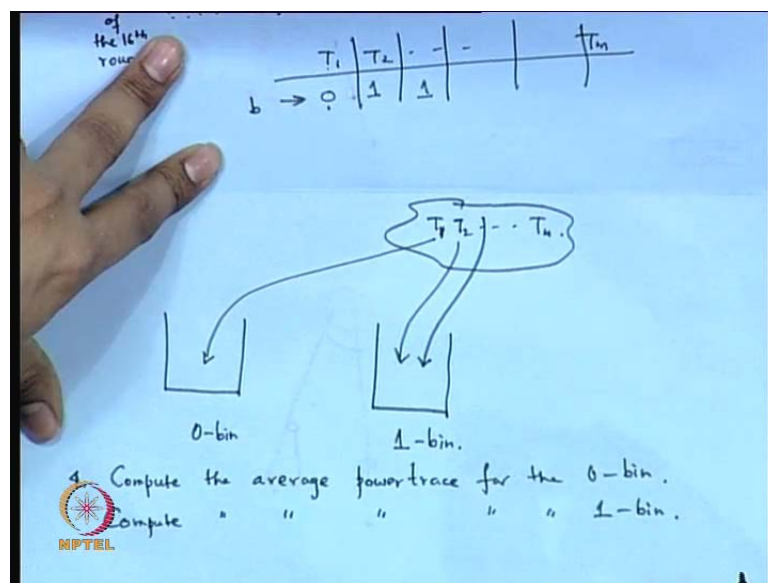
So, what we do typically is that, we can target one of the bits. So, therefore, actually we can target may be one of the out... So, there 4 output bits and there are 6 input bits of this S box. So, what we do is that, if you know that the inputs to this S box would actually be the combination of some information, which is coming from the previous round and the corresponding guessed key K_G .

So, what we do is that, we guess this K_G and from the ciphertext, we actually obtain these particular inputs, and then, we exalt with the K_G , apply the S_1 function and target one of this output bits. So, therefore, let us say that, this particular bit, that is, the $l_s b$ of the output is my target bit.

So, we calculate this particular target bit. So, what are the potential values of the target bit? It can either be 0 or it can either be 1. So, therefore, you mutually understand that all these T_1 , T_2 and till T_n will actually give several values of the bit values; so in all case it will be 0; in one case it will be 1.

Therefore, we can actually have a table like T_1 , T_2 and so until T_n and actually you can have using this particular selection functions, for some cases, you will generate 0s; some cases you will generate 1s for the target bits. So, this is the predicted target bit.

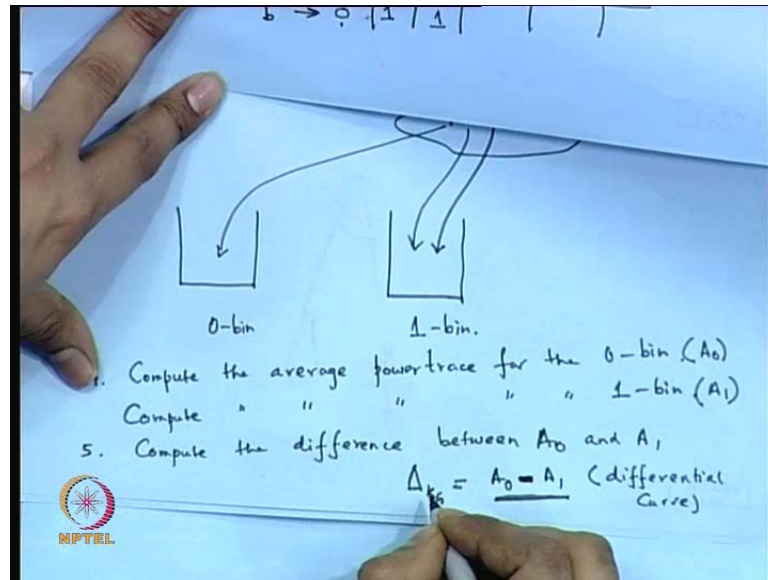
(Refer Slide Time: 23:29)



So, what we do now next is that, we actually maintain two bins: one I call as the 0 bin and the other is the 1 bin. So, what we do is that, we have got a cluster of the cases like, T_1 , T_2 and so on till T_n and we take depending upon, whether the corresponding bit is 0 or 1 like for example, I see that T_1 is 0, so immediately when I see T_1 is 0, I take T_1 and put it into the 0 bin.

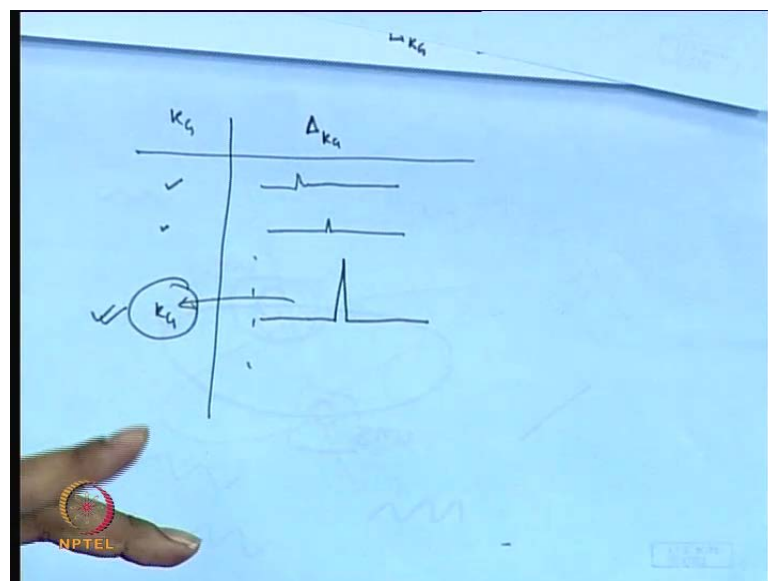
Similarly, I see that T_2 is 1, I put it into the first; so next T_3 this could be 1 and I again place it in the first bin. So, what happens is that, depending upon my guessed key, the corresponding power traces gets split it into the 0 bin and the 1 bin.

(Refer Slide Time: 24:46)



Next, what we do is that, we compute the average power trace for the 0 bin and we compute the average power trace for the 1 bin. The next step is to compute the difference, we compute the difference between these two averages; so I call this average as may be A_0 and I call this as average as A_1 and I calculate the difference between A_0 and A_1 and I say that my difference Δ for the guessed key K_g is equal to A_0 minus A_1 . So, this difference is actually a symbolic difference just to indicate that I am taking a point wise subtraction for the power traces.

(Refer Slide Time: 25:48)



So, therefore, you see that you will actually get a single differential curve; so I call this as a differential curve; so this differential curve is for this guessed key. So, how many potential guess keys can you have? You can have 2 power of 6 potential guess keys and therefore, all these potential curves of or potential values of K_G , you will have defined values of ΔK_G . So, therefore, for one K_G , you will have a differential curve; for another three, you would have another differential curve and you will have got an array of differential curves.

Now, the hypothesis is that, if there is one curve, which is a significant peak in this difference, then that actually tells you that, this K_G is the expected or the actual key which has been used. So, you will find that, there is a peak in the differential curve, then that tells us, that the corresponding key is the actual key.

(Refer Slide Time: 26:32)

1. Gather many power traces: T_1, T_2, \dots, T_m .

2. Guess a portion of the key (typically guess 6 bits of the last round). $\rightarrow K_G$

3. Perform a computation which depends on the ciphertext, (C_i) , portion of the guessed key, (K_G) , and Selection Function, (D) .

$D(C_i, K_G, b)$

b target bit.

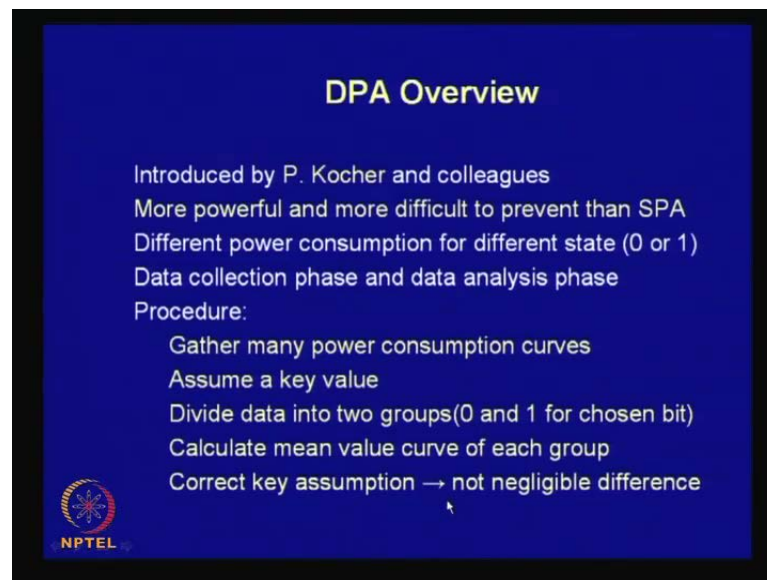
1st SBox of the 16th round.

	T_1	T_2	...	T_m
$b \rightarrow$	0	1	1	...

NPTEL


For each of this 2 power of 6 keys, you have to repeat this process and if you find that one particular key or some of the keys, which actually gives you a significant differential curve, that is, there is a spike in the differential curve, it indicates that the corresponding key is a probable candidate key.

(Refer Slide Time: 27:32)



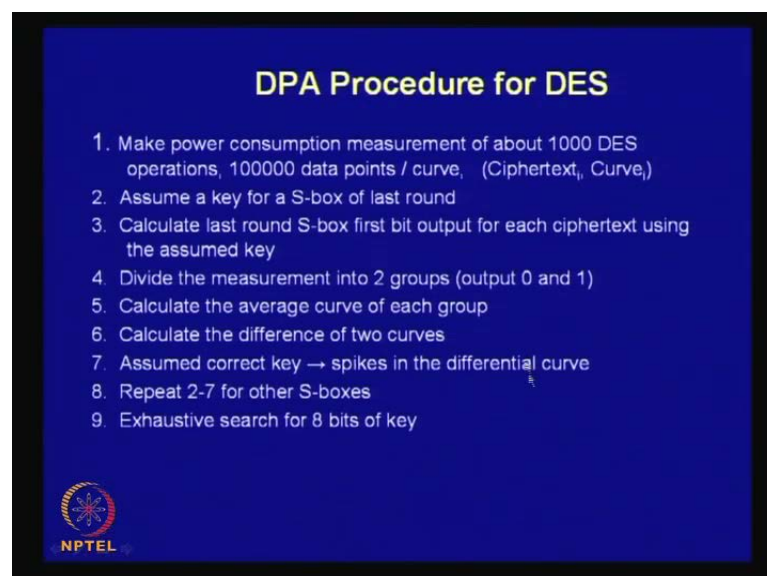
DPA Overview

Introduced by P. Kocher and colleagues
More powerful and more difficult to prevent than SPA
Different power consumption for different state (0 or 1)
Data collection phase and data analysis phase
Procedure:
Gather many power consumption curves
Assume a key value
Divide data into two groups(0 and 1 for chosen bit)
Calculate mean value curve of each group
Correct key assumption → not negligible difference

 NPTEL


So, that is being summarized here, that is, **you come** here, that is, calculate the mean curve of each group, correct key assumption we will actually imply that there is not negligible difference, that is, a non-negligible difference in the differential curve.

(Refer Slide Time: 27:42)



DPA Procedure for DES

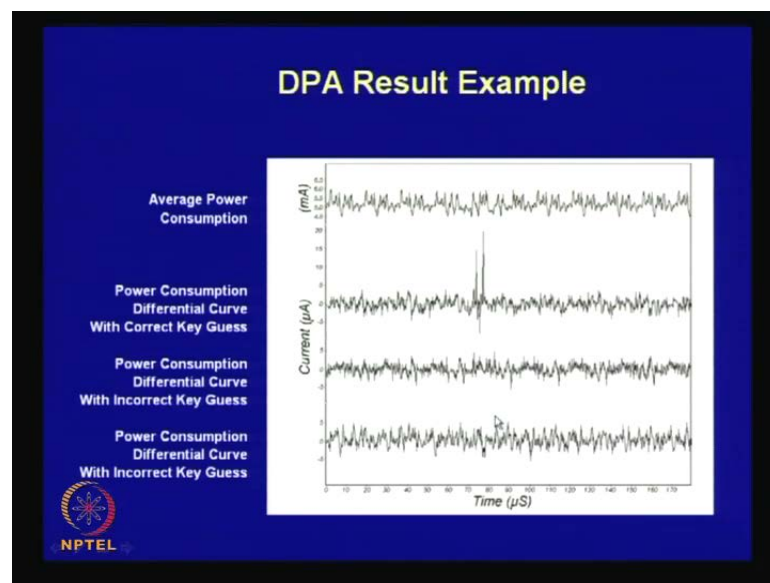
1. Make power consumption measurement of about 1000 DES operations, 100000 data points / curve, (Ciphertext_i, Curve_i)
2. Assume a key for a S-box of last round
3. Calculate last round S-box first bit output for each ciphertext using the assumed key
4. Divide the measurement into 2 groups (output 0 and 1)
5. Calculate the average curve of each group
6. Calculate the difference of two curves
7. Assumed correct key → spikes in the differential curve
8. Repeat 2-7 for other S-boxes
9. Exhaustive search for 8 bits of key

 NPTEL

Therefore, again this is summary of the same thing; you make power consumption measurements like ciphertext i and curve i . So, these are the corresponding things and you assume a key for the S box of the last round, you calculate that last round S box first bit output for each ciphertext using the assumed key, divide the measurements into two groups output of 0 and 1.

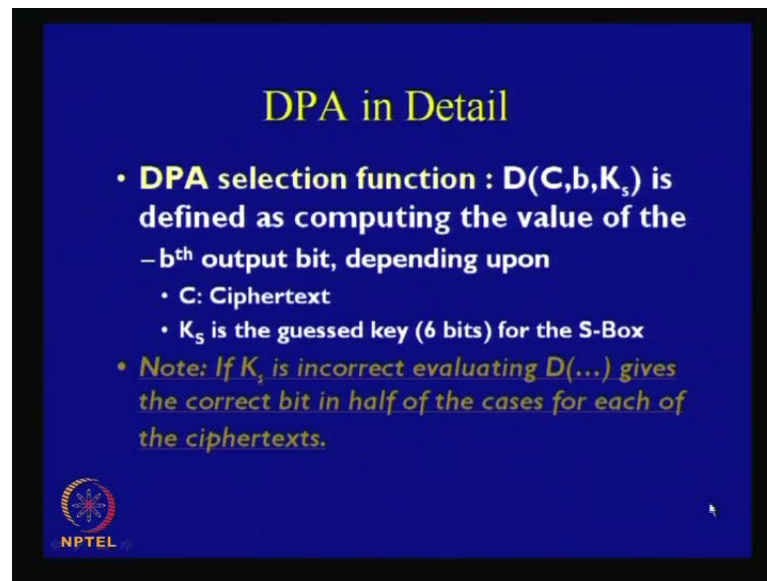
Then you calculate the average curve of each group, you calculate the difference of the two groups, that is, you calculate the differential curve, and then, you assume that if there is a correct key, then it implies that there will be spike in the differential curve. So, you have to repeat this 2 to 7 for the other S boxes. So, you can do an exhaustive search for the 8 bits of the key, that is, the priority bits actually which is there, so for the remaining 8 bits.

(Refer Slide Time: 28:30)




So, here is an example to show you that, some of the differential curves like, you can see that only here, that is, a significant peak in the differential curve well the other curves are actually not showing you the differential curve. So, therefore, these are the power consumption of differential curve with correct key guess. So, therefore, this particular differential curve, if it corresponds to a particular key that is a candidate key; while the others are not candidate keys.

(Refer Slide Time: 28:57)



DPA in Detail

- **DPA selection function : $D(C,b,K_s)$ is defined as computing the value of the b^{th} output bit, depending upon**
 - C: Ciphertext
 - K_s is the guessed key (6 bits) for the S-Box
- *Note: If K_s is incorrect evaluating $D(\dots)$ gives the correct bit in half of the cases for each of the ciphertexts.*

 NPTEL

So, therefore, in detail this is how or the little bit formally is that, this is the selection function D , which operates on the ciphertext C , computes a selection target bit b and there is K_s which is the key which has been guessed and it is defined as computing the value of the b^{th} output bit depending upon the ciphertext and also the guessed key that is the 6 bits of the key.

Now, note this important fact that, if K_s is incorrect, evaluating D will give you the correct bit in half of the cases for each of the ciphertext, that is, if your K is incorrectly guessed, then your selection function will actually function something like a pseudo random function. Because it will generate a b value, which is a 0 value or a 1 value; half of the times, it will be 0 and half of the times it will be 1 for each of the ciphertext. And therefore, you will not get a significant differential curves, that is, you will not get a differential curve with a significant peak, but if your K_s correct **it will** guess, then actually there will be a **box** which is exploited in this attack.

(Refer Slide Time: 30:05)

DPA in Detail

- **Attacker obtains m encryption operations and capture power traces, $T_{1..m}[1..k]$, with k sample points each.**
- **An attacker records the m ciphertexts**
- **No knowledge of the plaintext is required**

NPTEL

Now, in detail what we have done is like this, the attacker obtains m encryption operations and capture power traces, it can be indicated like T 1 to m, where these are like 1 to k means, that these are the sample points of the power traces. So, there are m power traces, each of them has got k sample points; the attacker records the m ciphertexts and you note that no knowledge of the plaintext is required, in this purely ciphertext only attack.

(Refer Slide Time: 30:36)

Attacker's Power Board

Sample Points

C I P H E R T E X T S	$T_{1[1]}$	$T_{1[2]}$		$T_{1[k]}$
	$T_{2[1]}$	$T_{2[2]}$		$T_{2[k]}$
	$T_{m[1]}$	$T_{m[2]}$		$T_{m[k]}$

NPTEL

So, an attacker records the m ciphertexts and then, it creates a power board like this, therefore, you see that each of this power boards says you that, this is the corresponding ciphertext, that is, the first ciphertext and there are k sample points in this power trace. So, this is as a pictographic representation, I call it the attacker's power board and similarly, if the second ciphertext, you have got k points; you can see that they are n sub ciphertext and each of them has got k points.

(Refer Slide Time: 31:00)

The Selection Function D

- Attacker knows L16, hence R15
- Attacker knows R16
- Guess K16 (6 bits)
- Compute output of f
- Compute the b^{th} bit of L15
- If K_{16} is wrongly guessed, then the computed value b matches with the correct result half of the time

$$f(R_{15}, K_{16}) = P(S(E(R_{15} \oplus K_{16})))$$

Now, what we do is that, we consider the selection function. Now, how do you calculate the selection function? You see that, in case of DES, the attacker knows L16 that is an attacker because of final permutation is known to everybody. So, once you have got the ciphertext, you can actually obtain the value of R16 and L16. So, if you know L16 and you know R16, you can if and if you guess K16, then we can compute the output of f and therefore, you can compute the b th bit of L15. So, therefore, what you can do is that, you can compute the b th bit of this particular S box and therefore, we can compute this corresponding output.

So, since you know that, you already have knowledge of R16 and since you have L16, from there you can actually obtain this value of R15 and if you guess this K16, you can actually apply this function f and you can obtain the corresponding b th bit. So, you can either obtain the b th of a particular target S box. So, as I told you that, if I target the first S box, I can obtain the corresponding b th bit.

So, therefore, if K16 is wrongly guessed, then the computed value b matches with the correct result half of the time. That is, as I told you that, if K s is wrongly guessed, it will work like a pseudo random function. So, where is the selection function? R15 exert with K16, then the application of this E and it apply the S box; so E is nothing but the expansion function, which is **the they are** as part of this function, then you apply the f function, **which is the...** you apply the S box, and then, you perform a permutation. So, **this is** this could be a composed selection function.


(Refer Slide Time: 32:52)

DPA in Detail

- **Attacker now computes a k-sample differential trace $\Delta_D[1..k]$ by finding the difference between the average of the traces for which $D(\dots)$ is one and the average for which $D(\dots)$ is zero.**

$$\Delta_D = \frac{\sum_{i=1}^m D(C_i, b, K_z) T_i[J]}{\sum_{i=1}^m D(C_i, b, K_z)} - \frac{\sum_{i=1}^m (1 - D(C_i, b, K_z)) T_i[J]}{\sum_{i=1}^m (1 - D(C_i, b, K_z))}$$

Principle: If K_z is wrongly guessed, D behaves like a random guess. Thus for a large number of sample points, $\Delta_D[1..k]$ tends to zero. But if its correct, the differential will be non-zero and show spikes when D is correlated with the value being processed.



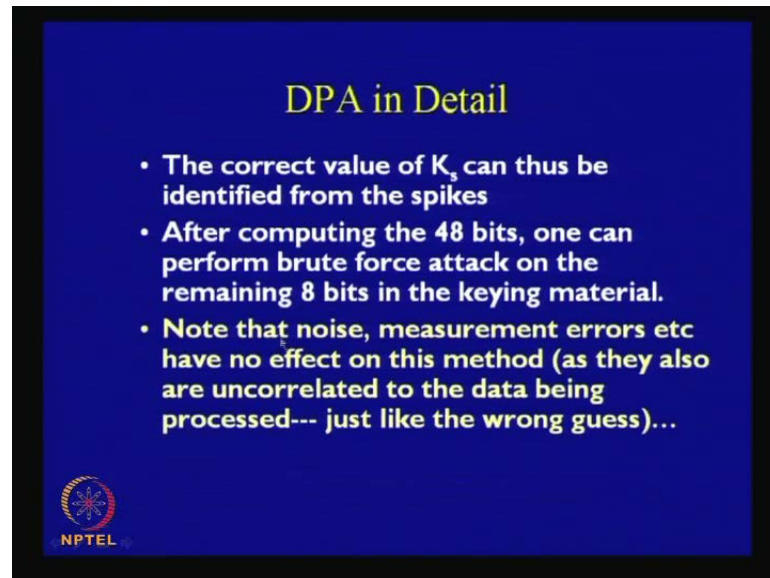
And what we do is that, the attacker now computes a k sample differential trace, that is, each differential trace will actually again of k samples, because there are k samples in your power trace and you will find out the difference between the average of the traces for which D is 1 and the average for those which D is 0.

So, therefore, you see although it is complicated form, actually it is quite simple. Because it takes this average is for when D is 1 and this is when D is 0. So, if D is 0, then this is actually **is** added here in this particular sigma; if D is 1, then it is added here, therefore, this average is for the 1 bin.

For all the power traces, which actually have gone to the 1 bin and these are the averages of those power traces, we have gone into the zeroth bit. So, we take a difference and we compute the delta D and the idea is that, if K s is again wrongly guessed, then D behaves

like a pseudo random guess, that is, for large number of sample points ΔD 1 to k will actually tend to 0; but if it is correct, the differential will be nonzero and show some spikes when D is correlated with the value being processed.

(Refer Slide Time: 34:11)



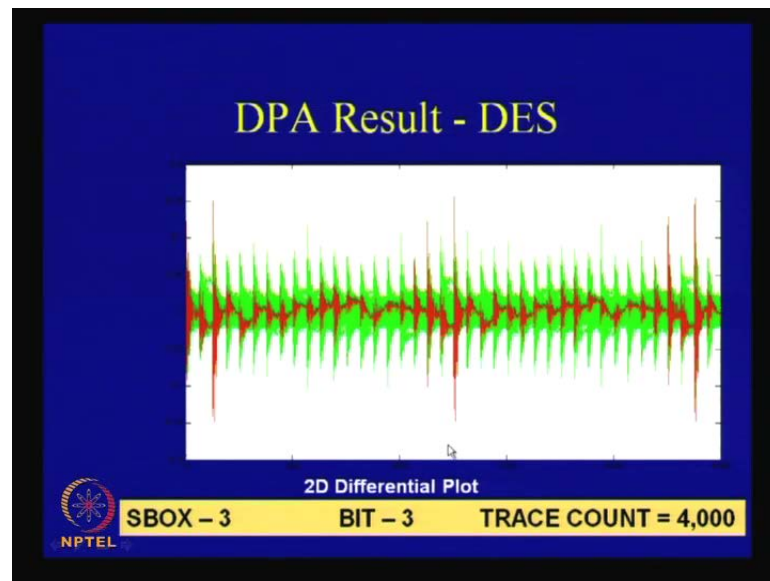
DPA in Detail

- The correct value of K_s can thus be identified from the spikes
- After computing the 48 bits, one can perform brute force attack on the remaining 8 bits in the keying material.
- Note that noise, measurement errors etc have no effect on this method (as they also are uncorrelated to the data being processed--- just like the wrong guess)...

NPTEL

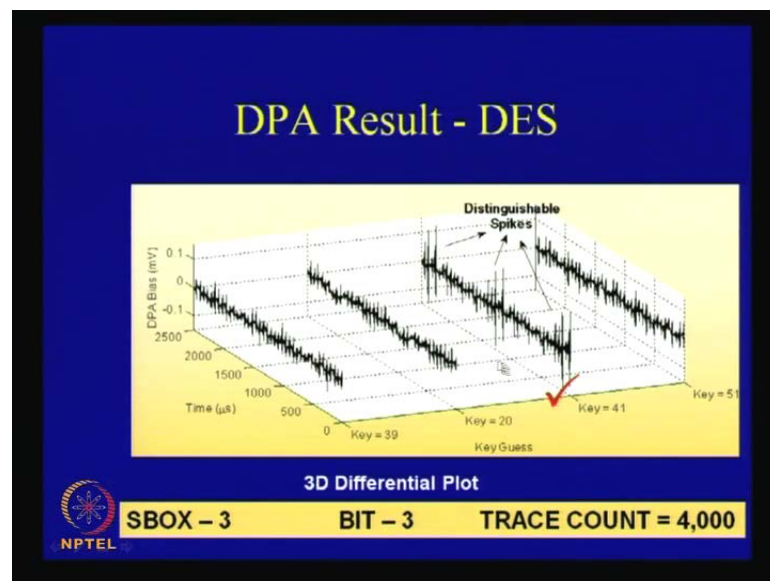
Therefore, that whenever the value is being really processed, then D will be actually correlated and therefore, we will get significant spikes in the power traces. So, here some ideas, I mean, some very important point is this, that is, note that, noise measurement errors, etcetera have got no effect on this method, because if they are present, then **they** like the wrong guess will be uncorrelated to the data, which is actually being processed. So, therefore, may be because of noises, the number of measurements will increase, but they cannot prevent the DPA attack from being successful.

(Refer Slide Time: 34:45)



So, therefore, this is a very important point and therefore, it is really a concern about how to actually implement so that the DPA attacks are prevented. So, here I have some snapshots of DPA results in our laboratory on... they have been used by like or even attack on DES, which show that using 4000 power traces, you can actually attack a cipher that is the DES algorithm.

(Refer Slide Time: 34:58)



And this is 3-dimensional plot, which tells you that, this is the correct key and you can actually see some of the distinct spikes actually.

So, these spikes have been generated and they **essentially** are leaking the information or telling that actually this particular key is the correct key while the others are not. So, this has been generated by our co-researchers in the laboratory. And we can see that, this tells us that there is another point, which also needs to be observed, that the distinct spikes are not continuous that is **that is used on by** the spikes are always present.


(Refer Slide Time: 35:55)

DPA in Detail

- **Attacker now computes a k-sample differential trace $\Delta_D[1..k]$ by finding the difference between the average of the traces for which $D(\dots)$ is one and the average for which $D(\dots)$ is zero.**

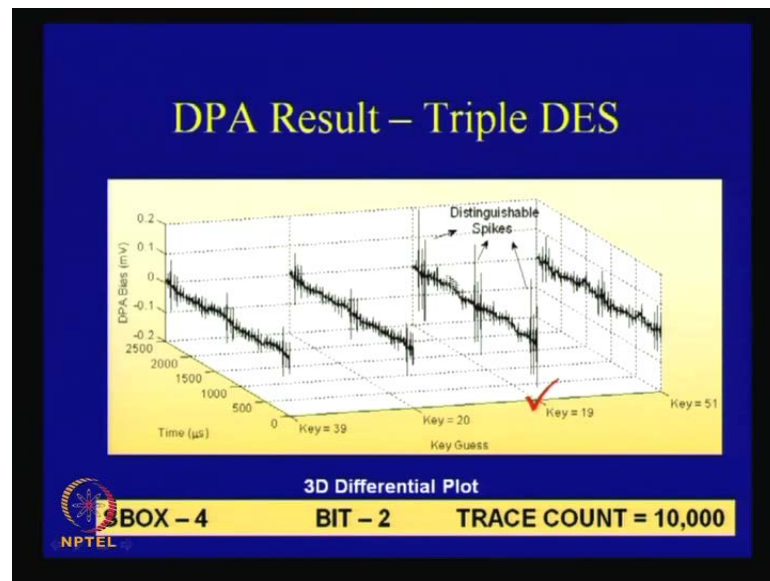
$$\Delta_D = \frac{\sum_{i=1}^m D(C_i, b, K_z) T_i[J]}{\sum_{i=1}^m D(C_i, b, K_z)} - \frac{\sum_{i=1}^m (1 - D(C_i, b, K_z)) T_i[J]}{\sum_{i=1}^m (1 - D(C_i, b, K_z))}$$

Principle: If K_z is wrongly guessed, D behaves like a random guess. Thus for a large number of sample points, $\Delta_D[1..k]$ tends to zero. But if its correct, the differential will be non-zero and show spikes when D is correlated with the value being processed.



The spikes are present only at instants and that means, that your actual computation that is the S box, which we are targeting is that always being computed; it is being computed certain time only then the thing is correlated and therefore, this particular line, that is, the spikes will be shown when D is correlated with the value being processed.

(Refer Slide Time: 36:07)



So, this is a point of time this when that even instant of time when D is correlated, not always and therefore you will find that the spikes will present at times and not at sometimes. So, if you increase traces, here is 10000 power traces on triple DES, if you increase, then it will be become more and more visible, that is, actually this is the correct key; the wrong keys will actually that the corresponding differential spikes will further reduce.

This is a DPA result on AES, which shows that using 15000 power traces targeting the 11 th S box and targeting the 8 bit, it has been compromised. So, you see that well algorithmically, these are actually wonderful algorithms, but these kinds of implementations can be concerned; so can be something like which has to be taken into care of.


(Refer Slide Time: 36:44)

Points to Ponder

Consider a 2-input AND gate, $q=a \& b$. Prepare a truth table with all possible transitions for the inputs and outputs. Note that the transitions can be $0 \rightarrow 0, 0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 1$.

Now compute the average energy for $E(q=0)$ and $E(q=1)$. Note that $E(q=0)$ will comprise of those transitions for which the output q makes a transition to 0. Similarly $E(q=1)$ will comprise of those transitions for which the output q makes a transition to 1.

Is $E(q=0)$ is the same as $E(q \neq 1)$. If yes under what conditions? Convince yourself that if $E(q=0)$ is not same for $E(q \neq 1)$, DPA works.



So, I will stop here, I mean, conclude with actually a point like, which we can think on like, why essentially our DPA attack successful, that is, what is the reason that DPA attacks indeed work. So, what we can do is that, in order to understand that, we can actually develop a 2 input and get truth table.


(Refer Slide Time: 37:15)

$q = a \& b.$

$q=0$			$q=1$		
a	b	q	a	b	q
0→0	0→0	0→0 ✓	1→0	0→0	0→0 ✓
	0→1	0→0 ✓		0→1	0→0 ✓
	1→0	0→0 ✓		1→0	1→0 ✓
	1→1	0→0 ✓		1→1	1→0 ✓
0→1	0→0	0→0 ✓	1→1	0→0	0→0 ✓
	0→1	0→1 ✓		0→1	0→1 ✓
	1→0	0→0 ✓		1→0	1→0 ✓
	1→1	0→1 ✓		1→1	1→1 ✓

$$\left. \begin{aligned} E(q=0) &= (9E(0 \rightarrow 0) + 3E(1 \rightarrow 0)) / 12 \\ E(q=1) &= (3E(0 \rightarrow 1) + E(1 \rightarrow 1)) / 4. \end{aligned} \right\} 4.$$

$E(q=0) \neq E(q \neq 1)$



So, we can consider like, there is a AND gate, which say q is equal to a and b and we can try to consider the corresponding truth tables for this AND gate. So, what we can do? We

know that, we can take the inputs as a and b and the corresponding outputs as q and we can actually consider all possible toggles of the input.

So, for example, the input could toggle therefore, there are four toggles that we will consider that if you consider like 0 to 0 toggle, 0 to 1 toggle, 1 to 0 toggle and 1 to 1 toggle; so there is a 4 toggles that we can concern. We can consider, every input if you consider like, there is a which goes from 0 to 0; b goes from 0 to 0; goes from 0 to 1; goes from 1 to 0 and goes from 1 to 1.

So, depending upon this, you can actually make your corresponding output transitions, because you know initially both of them were 0; so the output will still be 0 here; in all of them, it was 0, so it would be 0 only.

So, **this is the** four transitions are possible here; now what about this on 1? If I take **you that** this is 0 to 1, then 0 to 1 will indicate that again there can be 4 toggles here 0 to 0, 0 to 1, 1 to 0 and 1 to 1. So, the corresponding toggles here will be 0 to 0; this is 0 to 1; this is 0 to 0; this is 0 to 1. Now, what about here? It will be 1 to 0, so this is 0 to 0, 0 to 1, 1 to 0, and 1 to 1. So, again this is 0 to 0; this is 0 to 0; this is 1 to 0; this is 1 to 0.

So, then you have got 1 to 1, so you have got 0 to 0; 0 to 1, 1 to 0, and 1 to 1. So, your toggles are 0 to 0; 0 to 1 and you have got 1 to 0 and you have got 1 to 1. So, this is a state transition diagram, which we can make a state transition, which we can make for the AND gate and we can actually try to compute the value of the corresponding average energies for E_q equal to 0.

So, E_q equal to 0 will actually comprise of those transitions for which the output q makes a transition to 0. So, therefore, in this case, E_q equal to 0 will actually comprise of these transitions. (Refer Slide Time: 40:10) So, we will consider this, this, this, we also comprise of this; this, all these, this and this. So, how many are there from E_0 to 0?

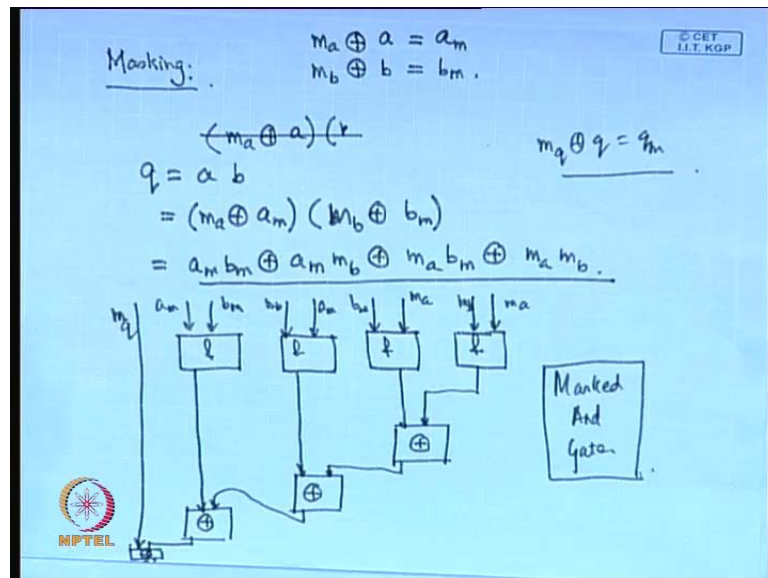
How many are there? **6, 8**, 9 and how many are there from E_1 to 0? **This one is there no, 3**. So, there are three 1 to 0; similarly, you can compute like E_q equal to 1. So, E_q equal to 1 will have again E_1 to and other 0 to 1 plus E_1 to 1; E_1 to 1 is 1 only and this **one** is there 1, 2, 3, so it is 3. So, you can immediately see that, these two energy levels like for

$E_{q=0}$ and $E_{q=1}$ are not the same, because these two average energies are not the same.

So, that is one of the reasons which was actually pin pointed that, this is the reason why the DPA attacks work that is $E_{q=0}$ is not actually equal to $E_{q=1}$. So, they are not the same, because the transitions like, the transitions which are there... So, when I considered, if I find out the average energy, I have to divide this by 12 and in this case, I have to average this by 4 observations.

So, what happens if all these transitions like, all these transitions are same? If all these transitions are same, then both of them will compute to the same value, but in CMOS technology, they are not the same and most of the hardware's that we have considering are actually developed on the CMOS technologies. So, therefore, in case of CMOS technology since these energy levels are not the same therefore, what happens with the average energy? $E_{q=0}$ and $E_{q=1}$ are not exactly equal and that is a design why DPA attacks work.

(Refer Slide Time: 43:05)



Therefore, you can try to think like, how essentially will you make a particular and computation and still ensure that they are prevented against this kind of DPA analysis. So, the idea which is normally used is called as masking.

So, while I will not go much into the masking technique, but one simple technique which is being adopted is like instead of performing a and b , what you do is that, you actually generate two random streams like m_a and m_b and you XOR m_a with a , which we have computed actually and with b and you obtain a_m which is mask and we obtain b_m .

So, now, if I compute m_a XOR a and rather if I want to compute a and b , if I want to compute this a and b , what we do is that, we need to compute q equal to a and b . So, what we do is that, instead of computing a and b directly, we use the masked value, that is, we XOR m_a with a_m and that with b_m or rather m_b XOR with b_m .

So, that is nothing; if you elaborate as a_m XOR m_b XOR with a_m XOR m_b XOR with m_a XOR b_m XOR with m_a XOR m_b , so what is normally done is that, instead of one AND computations, you require to do so many AND computations. So, therefore, typically masked and topology will actually comprise of all this 4 AND gates and we will actually need an idea of XOR gates. So, all of them will be ANDs and all of them will be XORs, so what is normally done is that, you take this m_a and this is m_b , what about this input? This is m_a and b_m ; this a_m and m_b and this is a_m and b_m , then what we do is that, we take this output, we XOR it with this; we take this, we XOR it with this; we take this output and we XOR it with this.

Now, this output is also not given up the straight, but it is actually also XOR with another third mask value which is called as m_q and this is the final output. So, therefore, instead of q being output, what is the output is actually m_q XOR with q , which I denote it as q_m . So, once you have q_m , this is the thing which is actually the output of your masked AND gate. So, this is the typical example of how a masked AND gate works.

Now, the question is that, I want to convince myself that, this is actually protected against DPA analysis. So, therefore you can actually frame a state transition table analogue to what we have seen in the context of the previous venerable AND gate and you can actually convince yourself that in this case, your average computation for E_q equal to 0 and E_q equal to 1 is indeed the same.

(Refer Slide Time: 46:46)

$q = a \& b.$

a	b	q	a	b	q
0→0	0→0	0→0 ✓	1→0	0→0	0→0 ✓
	0→1	0→0 ✓		0→1	0→0 ✓
	1→0	0→0 ✓		1→0	1→0 ✓
	1→1	0→0 ✓		1→1	1→0 ✓
0→1	0→0	0→0 ✓	1→1	0→0	0→0 ✓
	0→1	0→1 ✓		0→1	0→1 ✓
	1→0	0→0 ✓		1→0	1→0 ✓
	1→1	0→1 ✓		1→1	1→1 ✓

$$\begin{cases} E(q=0) = (9E(0 \rightarrow 0) + 3E(1 \rightarrow 0)) / 12 \\ E(q=1) = (3E(0 \rightarrow 1) + E(1 \rightarrow 1)) / 4. \end{cases}$$

$$E(q=0) \neq E(q \neq 1)$$

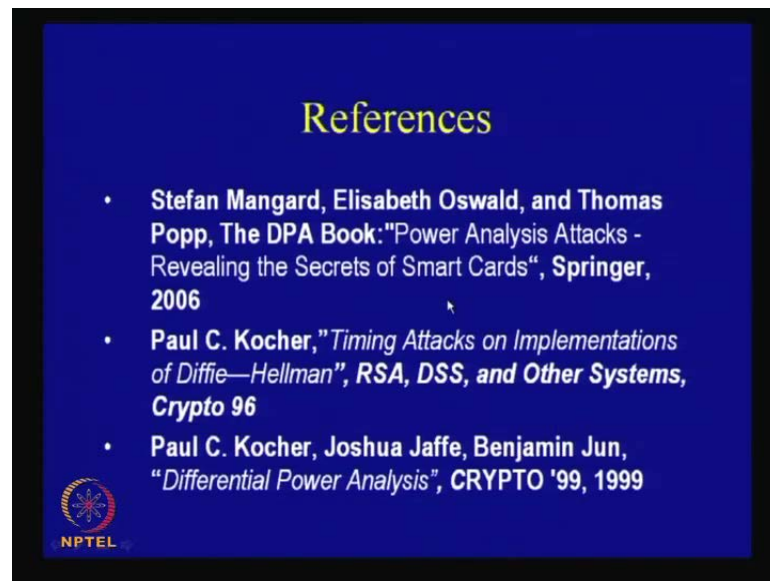
0→0
0→1
1→0
1→1

© CET
I.I.T. KGP

So, this you can take as an exercise and you can try. So, therefore, in this case, you're AND gate would actually not have 2 inputs, but number of inputs as you increased. So, therefore, if you have got a state transition table, then the state transition table in this case had how many rows? It had got the how many rows? 16 rows.

So, in this case, it will be more than that. So, you can actually try to make them and so the idea is that, for each of the inputs, there will 4 transitions possible. So, if there are 5 inputs, there will be 4 to the power of 5 possible input transitions and therefore, the state transition table will definitely get much bigger. And therefore, may be in order to analyze, we need to write a program to check that indeed the average power is same in case of the mass AND gate. So, this we can take as an exercise and try to see the indeed, the masked AND gate will protect against these kind of DPA analysis.

(Refer Slide Time: 47:31)



So, some of the references that are followed is this Stefan Mangard and Elisabeth Oswald and Thomas Popp's book, the DPA book which is power analysis attacks written by Springer. Paul Kocher's phenomenal paper, which was published in crypto 96 and in crypto 99 on timing attacks as well as on differential power analysis. So, anybody who **wants to start reading and** starts to work in this field is advised to go through these important references. So, with this, I conclude this course and thank you very much for your attention.