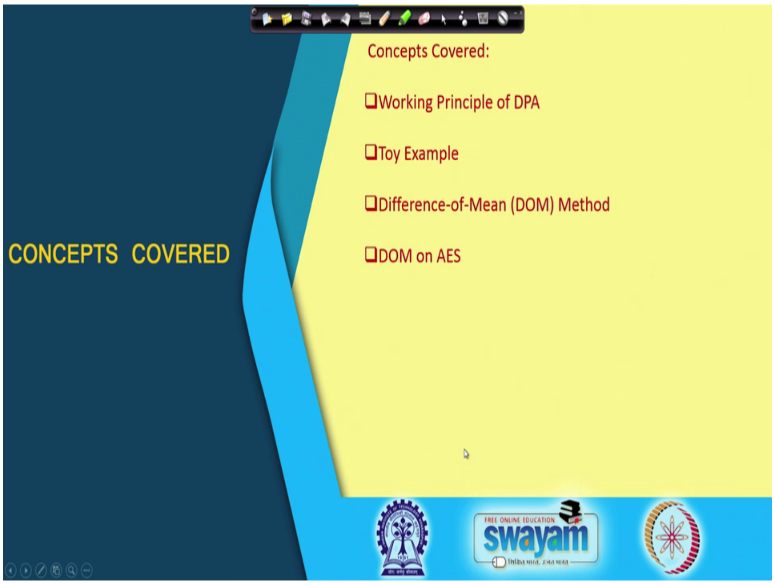
Hardware Security

Prof. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 25
Power Analysis – I

So, welcome to this class on Hardware Security. So, last day we started to discuss about side channel attacks. So, in particular we shall be trying to look into one of the very or most popular forms of side channel analysis which is known as Power Analysis in today's class. So, we shall initiate this topic.

(Refer Slide Time: 00:35)



CONCEPTS COVERED

Concepts Covered:

❑Working Principle of DPA

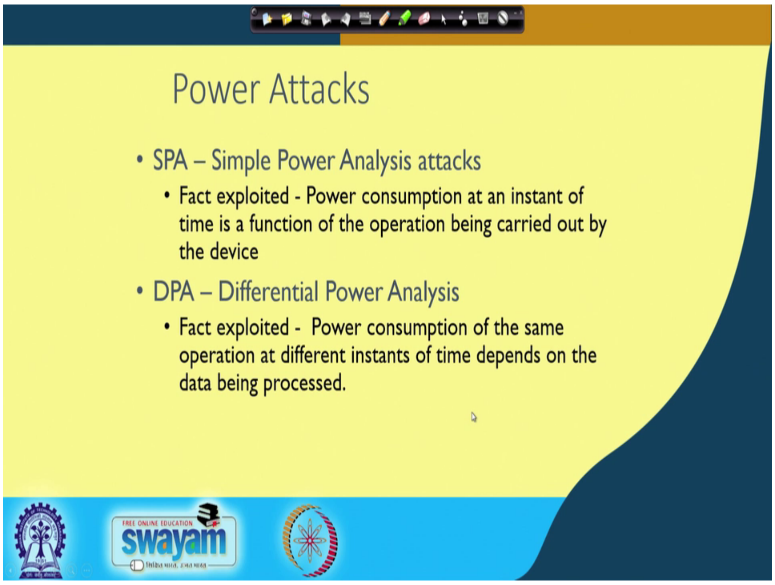
❑Toy Example

❑Difference-of-Mean (DOM) Method

❑DOM on AES

And today we shall be trying to kind of go through the working principles of differential power attacks or DPA as it is called we shall try to given toy example to explain how it works and we would like to talk about difference of mean which is a very popular method and classical method of doing DPA and try to see how DPA can be or DOM can be applied on AES which we have studied previously in the class.

(Refer Slide Time: 00:57)



Power Attacks

- SPA – Simple Power Analysis attacks
  - Fact exploited - Power consumption at an instant of time is a function of the operation being carried out by the device
- DPA – Differential Power Analysis
  - Fact exploited - Power consumption of the same operation at different instants of time depends on the data being processed.

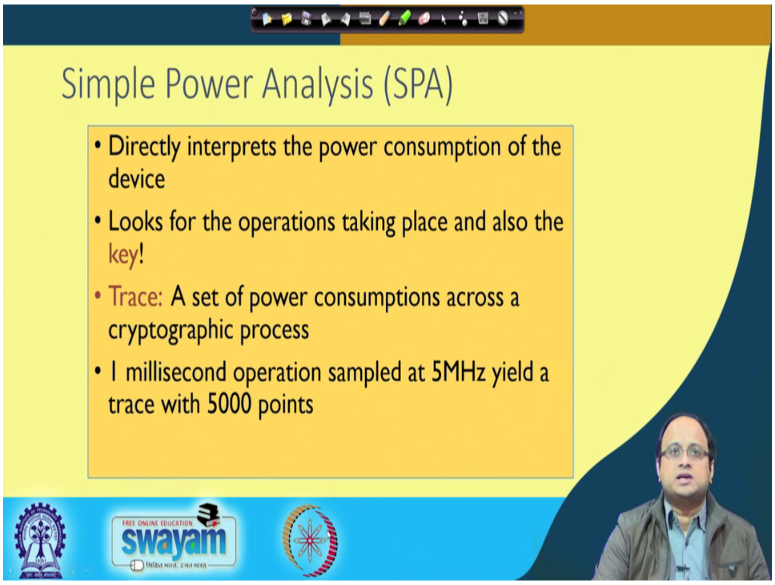
So, as we have already discussed there are two forms of power attacks broad category is the power attacks we have got simple power attacks where the fact which is expert it is at power consumption depends upon the underlying operation ,like whether it is squaring or whether it is multiplication or whether it is addition or doubling. So, the idea is at the power consumed depends upon the operation. So, if the architecture or if the implementation is very (Refer Time: 01:22), then often it may happen that you have different branches in your execution and where you are doing different operations ok.

So, therefore, right if the power consume depends upon the operation then that would imply there if you have a very accurate measurement of power consumption then you would have an idea about which operations are being taking place. And therefore, you know the execution path of your program and if your execution path is dependent upon some secrets then the secret can trivially leak. So, that is the idea of simple power attacks, on the contrary right you have got an even more powerful adversary which is called as differential power attack where it the fact which is exploited is that power consumed depends upon the underlying data.

So, therefore, right even if you protect your designs against Simple Power Attacks or SPA by say you know like doing always the same operations right. Even then DPA can work because DPA would fundamentally depend upon the operations which is being done like the I mean the rather the data on which the operations are being done.

So, therefore, you would know you would probably get an idea about like what your multiply this operating on what you are squaring is operating on or what you are doubling is operating on or what you are addition is operating on. So, and if you have got an idea about the intermediate data then you can do you know like follow it up by script analysis to get the actual secret key. So, DPA is very powerful attack model and what we have seen is that it can lead and as we seen more details in our class is there it can need to very efficient attacks against implementations.

(Refer Slide Time: 02:59)



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

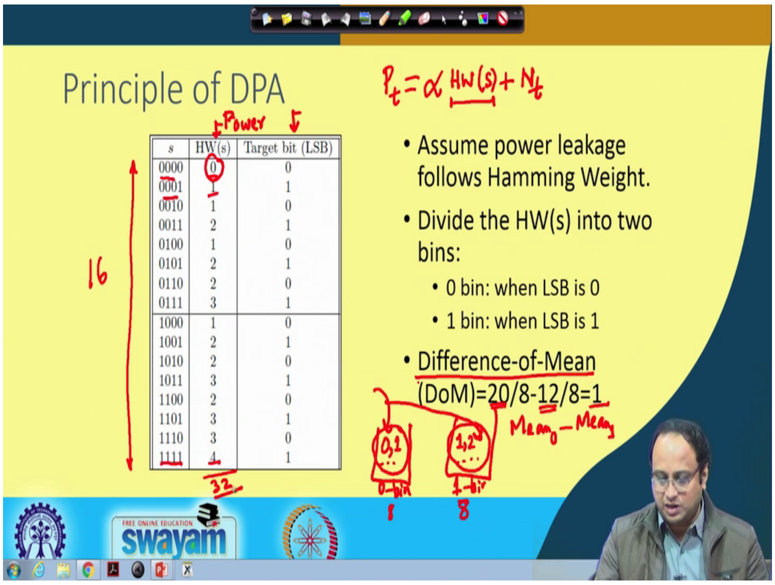
So, therefore, right as I said already that SPA or simple power analysis directly interprets the power consumption of the device and it tries to look for the operation which takes place and also the secret key ok. So, therefore, often the in the trace the secret key is evident. So, what is the trace because we will be using the word trace several times. So, trace is nothing, but a set of power consumptions across a cryptographic process ok.

So, often we basically have something which is called as a triggering signal which is used to kind of indicate the initiation and termination of a specific cryptographic operation and then the set of power consumptions which you essentially kind of accumulate during this period when the actual crypto operation is being done is called as a trace ok.

Note that we are not actually measuring the trace of only the cryptography operation, but rather we are measuring the trace of the total chip of the power consumption of the total chip and then we will try to statistically kind of correlate that power with your underlying data ok. So, for example, lot of taken example like suppose if you have if you have got you know like a 1 millisecond operation; that means, the total duration is 1 millisecond and if you are sampling is at 5 mega hertz rate like if you are oscilloscope is sampling at 5 megahertz rate then you will have a power trace which you have 5000 thousand points ok.

And therefore, this kind of observation or of real data which essentially could be in milli words or micro word depending upon your power consumption right and if you are plotting that across 5000 points of time there essentially this statistics which you collect or accumulate is called as a trace is called as a power trace ok. So, now, our observation our objective will be that we will try to utilise this power trace.

(Refer Slide Time: 04:47)



Principle of DPA

s	HW(s)	Target bit (LSB)
0000	0	0
0001	1	1
0010	1	0
0011	2	1
0100	1	0
0101	2	1
0110	2	0
0111	3	1
1000	1	0
1001	2	1
1010	2	0
1011	3	1
1100	2	0
1101	3	1
1110	3	0
1111	4	1

- Assume power leakage follows Hamming Weight.
- Divide the HW(s) into two bins:
  - 0 bin: when LSB is 0
  - 1 bin: when LSB is 1
- Difference-of-Mean (DoM)=20/8-12/8=1

So, now, therefore, let us. So, as we have kind of have an idea about what and SPA essentially means we shall in particular try to look what DPA means actually ok. So, as I said that in DPA the assumption is that the power consumption depends upon the underlying data. So, here is and very simple example a very simple example to illustrate that this principle of the principle on which DPA is based has got nothing inherently to do about the algorithm. So, rather if you take any binary encoding of information of data you will probably find you know like that why DPA works ok.

So, let me take a very simple example of say a 4 bit state of some arbitrary algorithm and then let me observe the hamming weight of the states. So, as you can see write that I have kind of varied s from for all these values like for example, since as we the 4 bit value there are 16 possible values ok. So, I have taken all these 16 possible values and I have observe the hamming weight in this column. So, the hamming weight means the number of ones which have there in the state.

So, for example, in the first state which is 4 0s there are no 1. So, the hamming weight is 0 in the second state right you have got 0 0 0 1. So, there is there is 1 1. So, the hamming weight is 1, likewise if you go to 1 1 1 1 right hamming weight is 4 because there are 4 1s. So, why do I observe the hamming weight now because we will see that in VLSI there are different power models ok. So, the objective of a power model is try to get a hypothesis of the power of course, power consumption is a very complex manifestation of several things, but what has been found as that a major contribution comes from the inherent encoding of the state ok.

So, the idea is that there different power models like hamming weight, hamming distance, some other you will see as we progress in the class, but a very simple power model is what is called as a hamming weight power model. The idea is that you take the state and try to see the number of ones typically you will see that the power consumed will be proportional be the hamming weight of the s ok. Of course, that other noise terms and there are other aberrations and things like that, but it has got a proportionality so, it has the proportionality is with you know the hamming weight of the s.

So, we can write this also in the form of a formula like the power consumed at say for example, time t you can write that as say you know like alpha which is a constant in to say the hamming weight of your target state plus some noise term ok. So, this noise term is set denoted as N t. So, this could be a Gaussian noise ok. But this is essentially what I am meaning as the power model actually.

So, so you can actually you know like virtually simulate the power model also using this model ok, but in a very simple setting assume that the hamming weight is representing your power model; that means, when the hamming weight is increasing I am expecting in the power will be more when the hamming weight is less then I am expecting the power will be less ok. So, is a very simple setting. So, in this setting right if I observe then; that means, this column of the hamming weight column is kind of indicatives of your power consumption ok.

So, now if you observe the target, bit of the state s ok. So, you can see that this column stores the LSB of state s. So, the LSB of state s is sometimes 0, sometimes 1, sometimes 0, sometimes 1 and so on. So, it kind of alternates right. So, now, I do a very simple exercise, the exercise is as follows I take 2 bins ok. So, I call one of them as say 0 bin and the other one as 1 bin and I take a particular power value say hamming weight of s indicated by hamming weight of s and put that either in the 0 bin or in the 1st bin depending upon whether the LSB is 0 or not ok. So, I put it in the 0 bin in the LSB 0 I put it in the 1 bin if the LSB is 1.

So, therefore, you can easily see that the first is hamming weight s here HW s will go to the 0 bin right. So, 0 goes here next you have got HW s is 1 ok. So, this goes to the 1 bin ok, next you have got another one, but this one now goes to the 0 bin the next two goes to the 1 bin like this right in basically start accumulating and you have got out of 16 the 0 this 0 bin will have 8 corresponding components and the 1 bin will also have 8 data.

And now I basically take an average of all these values and I take the average of all these values. So, this I call as two means say; mean 0 and mean 1 and I take the difference of these things that I take mean 0 minus mean 1 and I this is what I call as the different of mean or DoM.

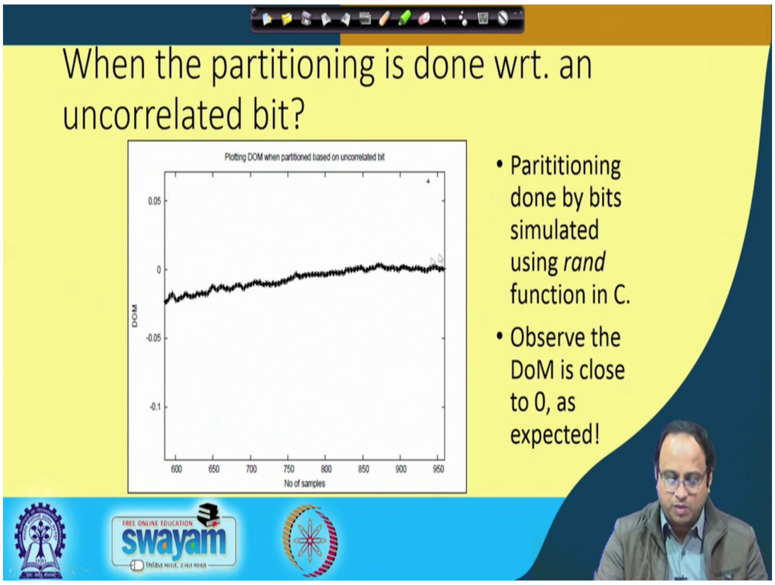
So, it turns out and you can check that later on that the data which is over in the 0 bin and the 1 bin you will see that one of the bins the number of the if you add up those values, they will add up to 20 and therefore, the other one will be 12 because the some will be 32. So, therefore, if I take the difference of mean the difference of mean will turn out to be 1 in this case which is the non 0 value ok, but at the same time observe that the total sum is 32 right. So, if I add up these things I will get 32.

Now, if the split was random was purely random then I would have expected the expected sum of 0 bin and the expected sum of 1 bin both of them would be 16 because they each of them suppose it goes with you know like to the 0 bin with the half probability and to the 1 bin with also half probability then I would have got this 32 speed exactly into 60 then 60 or at least around 60 then 60.

So; that means, the difference of mean would have been 0 in that case so; that means, if you have basically splitting right depending upon the s which is especially the you know like the LSB of s for example, then you find a non zero difference of mean whereas, if you kind of split you know like a randomly or do a random partition then the difference of mean is 0.

So; that means, if the fact that you are splitting the hamming weight of s depending upon or correlated with some kind of state bit of s can easily bit distinguished whether with respect to you know like whether you are doing the splitting in a random fashion and that essentially tells you that there is an avenue of attack ok. For example, like if you want to see that if you does do this splitting in a random fashion then this is how we look like?

(Refer Slide Time: 11:49)



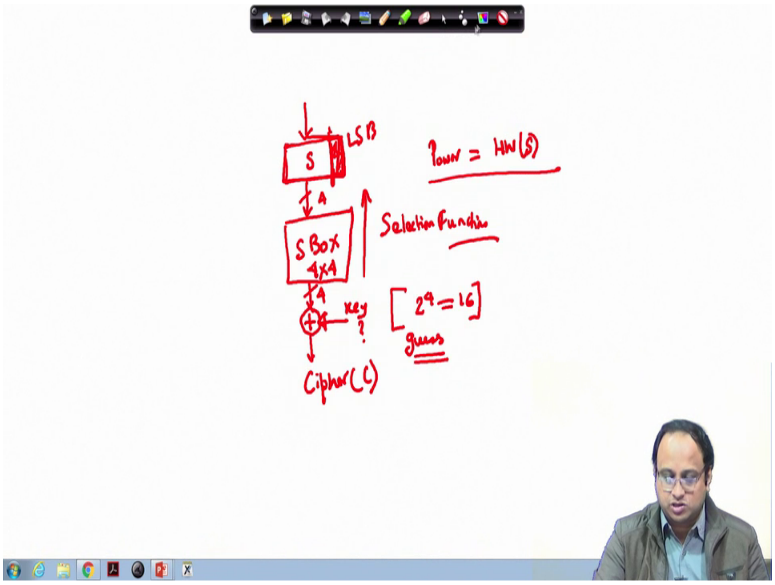
When the partitioning is done wrt. an uncorrelated bit?

- Parititioning done by bits simulated using *rand* function in C.
- Observe the DoM is close to 0, as expected!

So, you know like if I just partition based upon. So, this partition in has been done by rand function which is not necessarily a great random on generation, but suppose we use that as to for the splitting. Then you will find that the observe that the difference of mean right gets closed to 0 as we accumulate number of samples ok.

On the other hand when you are doing the partitioning with respect to a correlated bit; that means, for example, the LSB of s it could be any other bit of s also it could be also the you know like xor or exclusive power of this bits as well you will find that the difference of mean will get two get two some non zero values. So, this you can you know like try to also imagine that this has got the you can basically use it to essentially bound it simple conceptualize the simple difference of mean attack ok.

(Refer Slide Time: 12:43)



For example, suppose you know like you have got an intermediate state S so, this is the state s and this you follow up with some kind of S box ok. So, there is in S box as we have seen already in context of AES to AES for example and then I do an exclusive all with some secret key and I generate the cipher this is my cipher; cipher denoted by C.

So, now what I will do is, I will try to guess this key ok. So, there are you know this just imagine that this is a very small and hypothetical example with say you know like 4 bits of S box. So, there are several 4 cross 4 S boxes suppose I take 1 of this 4 cross 4 S boxes and make a very simple cipher.

So, what is the total key size space therefore, there are 2 to the power of 4; that means, there are 16 possible key values and if I just give you the cipher you cannot tell me you know like that which of them is correct ok, but now I also give you the you know like suppose I give you this the hamming weight of s ok; that means, you get essentially observe the power consumption and imagine that a power consumption is nothing, but the hamming weight of s.

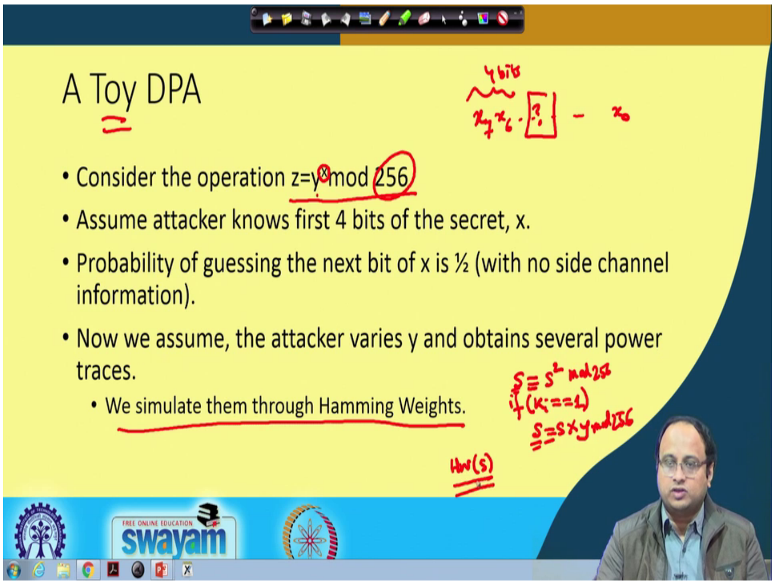
So, now what you will do is you will guess this key ok. So, you will guess you will make a guess for this key because you know that anyone any anyone of there is possible , but observe that if I do not give you this power information you have got no way to understand where are these guess is correct ok. Now what we will see is that, with this power right with this power information you can distinguish between the correct guess with the other wrong ones how can you do that you will guess this you will S or with the cipher you will take the inverse S box ok. So, if you take the inverse S box you will come to the state S ok. So, imagine that I just take the LSB of S I; I basically take the inverse and I just find out what is this LSB ok.

And now, I essentially do the splitting as I told. So, I will take I will do the exercise for all the HW s values and I will split them. If my guess was correct then; that means, I have correctly predicted this state S LSB of S and therefore, my partitioning will essentially take place just as we have seen. On the other hand if my guess was wrong that means for all the 15 wrong guesses I will find out that this will sometimes match will sometimes not match. Because then this function which is also called often as the selection function will essentially operate as a random number generation are will be like a pseudo random function generator more correctly.

And therefore, right you will find that this partition the partitioning of the power traces will be in a random fashion and therefore, the difference of mean will be close to 0. So, therefore, this fact that the for the correct guess you will have a high difference of mean verses for a wrong guess you will have the small difference of mean can be utilized as a distinguishers the distinguish the correct key from the wrong keys ok. So, this is basically in a very simple say terms you know like an exploitation of how it works.

And now, let us see some more concrete examples to understand how it can be applied to ciphers.

(Refer Slide Time: 16:05)



A Toy DPA

- Consider the operation $z=y^x \bmod 256$
- Assume attacker knows first 4 bits of the secret, x.
- Probability of guessing the next bit of x is ½ (with no side channel information).
- Now we assume, the attacker varies y and obtains several power traces.
  - We simulate them through Hamming Weights.

$\overbrace{x_7 x_6 .}^{4 \text{ bits}} \boxed{?} - x_0$

$S = S^2 \bmod 256$

if $(K_i == 1)$

$S = S \times y \bmod 256$

$HW(S)$

So, first let me start with a very toys cipher a toy example. So, this example is of the again a reminder of the RSA algorithm because I am just taking an exponentiation routine ok. So, the exponentiation routine is nothing, but denoted as z equal to y to the power of x mod 256 again this is a toy example.

So, 256 really a small number for practical settings, but, imagine that I want to know this x and assume that x is also in 8 bit value and assume that the attackers can start with x 7 x 6 and whatever write till x 0 and suppose it knows the first 4 bits of x ok. So, attacker knows the first four bits and again like that Kocher like the Kocher's attack where we have seen my objective is to guess the next bit.

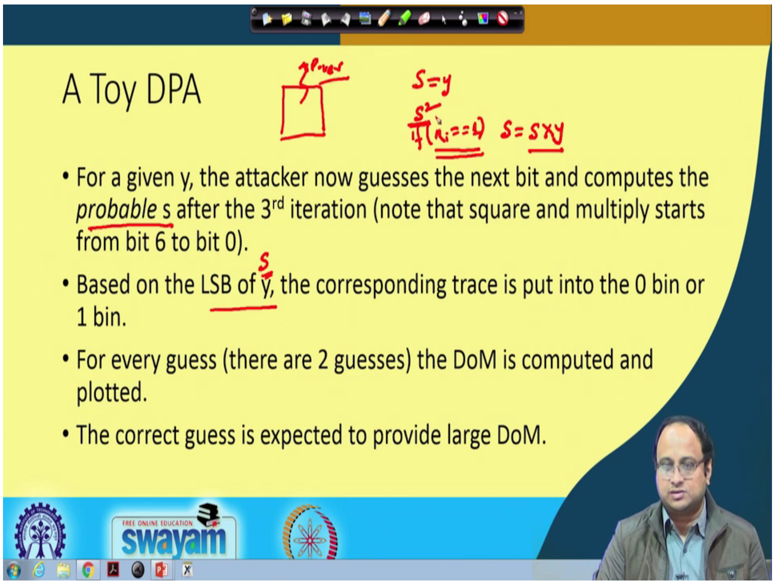
So, the probability of guessing the next bit with no side channel information is half because I have no way to understand the whether it is 0 or whether it is 1. So, now, we assume that the attacker varies y; that means, attacker kind of change is the value of y and obtain several power traces like in Kocher's attack we are observing the timing right here we are observing the power signal in.

So, so we can you know like simulate them in a setting just to understand by hamming weights ok. So, I means I can simulate remember that in the square and multiply algorithm there is a there is a state S right which I multiply which I am either squaring or a multiplying with y ok. So, what I do is that I kind of monitor the value of that state S and I plot the hamming weight of that state s as an indication of the power value ok

So, therefore, I observe the hamming weight of that state S which I have you know like I am. So, in the square multiply algorithm I always do a squaring and if the key bit is 1; that means, if the key i or the ith bit is 1 then I multiply s with y that was essentially my. So, again you know all these equalities are modulo 256 in this case right I am doing mod of n where n is simple 256 so.

So, therefore, right when I am doing this operations right when I am doing operations I suppose monitor the hamming weight of s the objective is that this gives me an indication of the power consumption of this of the algorithm of the circuit. So, therefore, right with this example.

(Refer Slide Time: 18:25)



A Toy DPA

$S = y$

S^2

$\text{if}(k_i == 1)\ S = S \times y$

- For a given y, the attacker now guesses the next bit and computes the *probable* s after the 3rd iteration (note that square and multiply starts from bit 6 to bit 0).
- Based on the LSB of $\overset{S}{y}$, the corresponding trace is put into the 0 bin or 1 bin.
- For every guess (there are 2 guesses) the DoM is computed and plotted.
- The correct guess is expected to provide large DoM.

So, now let us see you know like how you can do the attack for a given y the attacker now guesses the next bit and computes the probable s after the 3rd iteration ok. So, note that the square and multiply starts from bit 6 down to bit 0. So, you are going from bit 6 bit 7 bit 5 and so on till bit 0. So, therefore, right now based on the LSB of y the corresponding trace is put into the 0 bin or the 1 bin ok. So, here what I am saying the you know like the. So, basically the idea. So, so maybe you know like you can say that I am using y also as s, but for clarity probably you can think of this is based on the LSB of the running variable ok. So, that is probably this would be better to say that this is the LSB of S because what I am doing is that I am initialising S to y and I am done squaring every time and if the key i bit is 1 that means, if my secret is 1 that I am multiplying S with y right to get y power of x.

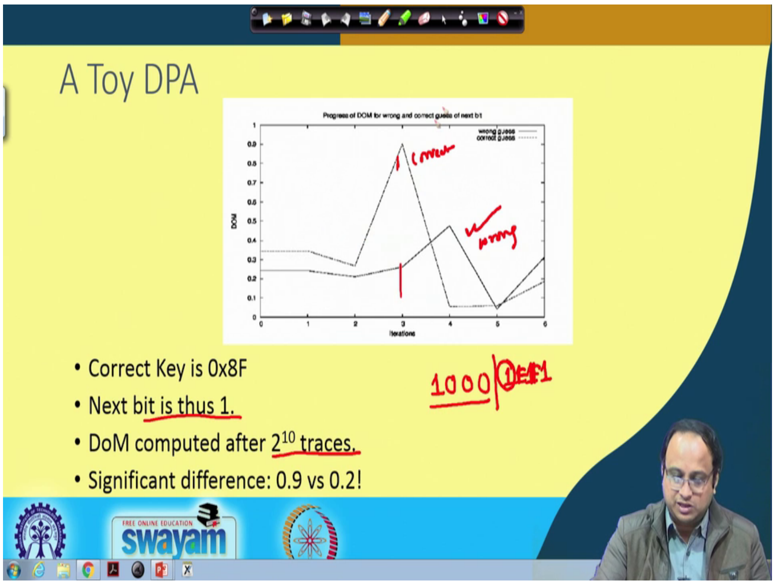
So, therefore, right what I am observe. So, so, when I am you know like I am observing the probable s after this iteration and then I am splitting my or putting that corresponding hamming weight s right which is my power value into the 0 bin or into the 1 bin depending upon what is the LSB of this S, note that I do not know as an attacker whether this bit was 1 or not right. So, the probable s can be after the value of s square or it can be because of s square and then multiplied with y. So, I have got no way of understanding that, but right in a simulation since I am observing the power value the power value is with respect to the actual correct key.

So, therefore, therefore, right the power that I am getting is with respect to either you know like s square or after s square has been multiply with y so, but when I am guessing right then the ambiguity from the attackers point of view is that I have got this power value whether I will put that into the 0 bin or whether I will put that into the 1 bin.

So, that depends upon it guess. So, because the attacker I will guess and there are two choice of the guess right it is either 0 or it is 1. So, if it is 0 then the attacker calculates s square and finds out the LSB of S square and depending upon that it does the splitting and if the, you know like the if the guess is 1 right there will basically calculate s square multipliers with y and again takes the LSB and then splits.

So, now which one is correct that is the question from the attackers point of view. So, therefore, what the attacker does either for every guess there are two guesses the difference of mean is computed whichever is the you know like the higher value of difference of mean is a probable guess of course, this is the statistical attack. So, therefore, what it implies there is that if you observe large number of samples then you are expected to get a better result and or clear separation.

(Refer Slide Time: 21:25)



A Toy DPA

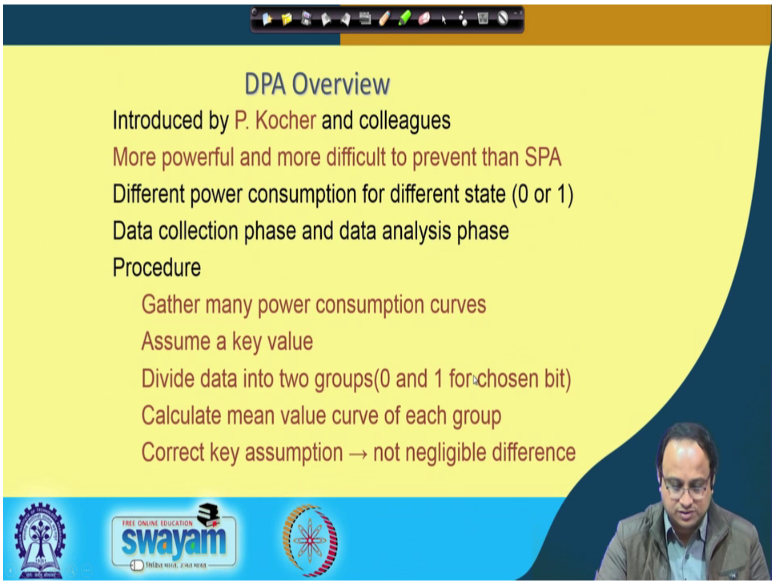
- Correct Key is 0x8F
- Next bit is thus 1.
- DoM computed after 2^{10} traces.
- Significant difference: 0.9 vs 0.2!

So, here is simulation that we run in the in this case the correct key is 0 x 8 F and therefore, the next bit is 1 remember that I know the first 4 bits right. So, therefore, I know one I know the you know the first 4 bits. So, I know that it is 1 1 0 0 0 and then I have got 4 1s in the correct key. So, therefore, right I mean what I am trying to say is that my this is my secret key right and then I have got to F F F. So, F for I mean I have got F. So, so basically I have got 1 1 1. So, I have 4 1s ok.

So, therefore, what I know till this point the attacker does not know this. So, the attacker wants to know this. So, therefore, the correct next bit is 1 ok. So, now, you observe that this is my there is a wrong guess and there is a correct guess. So, this one is for the wrong one. So, this is the wrong guess and this one is the correct guess easily you can see the attacker in correct guess there is a very significant difference of mean peak it is kind 0.9 verses 0.2 which something like 2 to the power of 10 traces you can observe more to get a better separation.

So, therefore, right it indeed kind a tells we that symbol difference of mean attack works in this setting and I should work and likewise right you can also think of extending these attacks.

(Refer Slide Time: 22:47)



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

So, and kind of formalize how it works ok. So, so to the basic idea of difference of mean was introduced by Paulo Kocher and in a in a 99 and 99 paper and his colleagues. So, it is more power full as I said and different and more difficult to prevent than SPA and there are different power consumptions for different states like 0 or 1 ok. So, idea is that it basically kind of reliance upon this assumption that power consumption varies with data whether it is processing 0 or whether it is processing 1 ok; that means, when I am saying it means typically the internal gates of your circuit ok.

So, it could be in a very simple setting and an gate for example, and an gate are all operating upon 2 data say x and y and ending then it could depend upon whether your operating on 0 or 1 if even simplify let us consider an inverter for example, of Simons inverter.

So, whether you are you know like whether you are operating on 0 as an input or whether your operating on 1 as an input, your power consumption with vary ok. And since it vary right you find out a manifestation that the power consumption depends upon the underlying data in our normal technology and that is exactly what is being exploited in these attacks. So, typically these attacks have got data collection phase and data analysis phase initially you should accumulate power traces and for that you need an experimental setup and I will try to give an overview on the experimental setup in the future class.

So, you take basically you take an overview I mean you basically collect your power traces in an accurate setup. So, that the noise is less and the quality of the power traces is high and then you have got a data analysis phase where you basically try to analyse the information which you have collected. So, the procedure is very simple you basically gather many power consumption curves ok.

So, typically 1000s few 1000s for ciphers and then you assume a key value. So, basically you guess the key. So, remember that if you have got a 128 bit key like you have an AES you cannot guess a 128 bit key because if you do that right then your case key complexity can be 2 to the power of 128 and therefore, you do not have any advantage over a brute force attack.

On the contrary right this attack is very powerful because you can do that in a divide and conquer manner. So, you can basically guess parts of the key and then you can combine this information. So, you can recover the key in a byte by byte or dibble by dibble fashion. So, its very powerful in that sense and what we do is that you basically divide into groups. So, is the difference of mean attack which is the most probably classical form of doing an DPA attack, you basically divide the data into two groups a 0 bin and a 1 bin for the chosen bit.

And you calculate the mean value curve for each group and then the idea is that if your assumption or if your guess was correct then you will get a non negligible difference in the differential curve and that essentially with distinguish your correct key vis a vis the wrong key ok. So, that is essentially the idea of DPA and what we will see in the next class is how you can do a DPA attack on AES ok. So, let me stop at this point and we will continue on the DPA on AES in the next class ok.

Thank you.