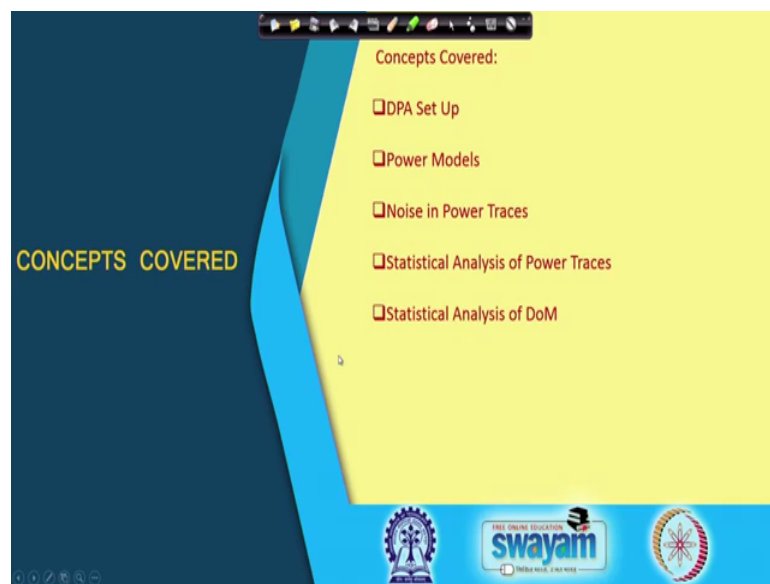


Hardware Security
Dr. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 27
Power Analysis- III

So, welcome to this class on Hardware Security. So, we were studying on Power Analysis; so, we shall be continuing our discussions with that.

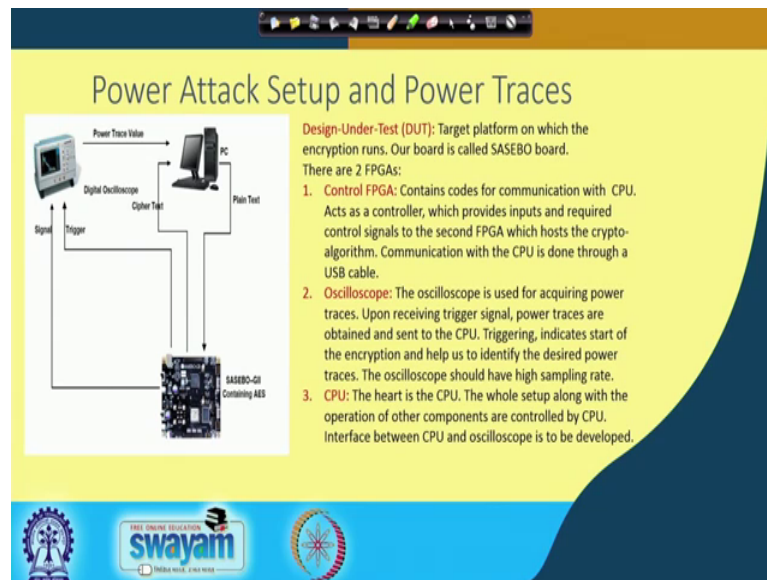
(Refer Slide Time: 00:22)



So, in particular we shall be trying to continue our discussions on difference of mean based DPA attacks. I will start with how do you develop a setup which is very crucial for proper experimentation on DPA. We shall be discussing about power models the various kind of useful power models that we generally encounter.

So, I will be trying to discuss about 2 usual power models like which are called as hamming weight and hamming distances. And then I will be discussing about another very important concept or component in power which is essentially the intrinsic noise, which is there and we shall be discussing about statistical analysis of power traces. And we shall be trying to interpret difference of mean in a statistical setting or using statistical tools.

(Refer Slide Time: 01:06)



So, let us start with this background. So, this is an overview about how a power attack setup typically looks like? So, if you observe there are 3 important components here.

So, you have got the design under test which is essentially the you know where you are basically putting your encryption algorithm. So, essentially it could be an FPGA it could be a smart card or a microcontroller, which is essentially the target of your attack. On the other hand so, this FPGA so, this is I mean there are different platforms that you can of course, target, but typically the platform that we will be considering are essentially called as SASEBO boards or SAKUDA boards these are very standard boards which are available internationally, where we can do our experiments on power attacks and the other kinds of side channels.

So, this board is called essentially it has got 2 important parts and out of them the most important there are 2 there are 2 FPGAs in the boards. So, one FPGA is the control FPGA, which contains the codes for communication with the CPU and it acts as a controller, which provides inputs and required control signals to the second FPGA which houses the crypto algorithm.

So, the crypto algorithm is kind of housed in the second FPGA and the first FPGA or the control FPGA essentially does communications. So, it basically communicates with the CPU and it acts as a central controller. So, therefore, if the objective of the of it right is I mean the controller or the control FPGA essentially has got the key role of performing

communication with the CPU and it is typically done through an USB cable. The other important component in this setup is the oscilloscope.

So, the oscilloscope is typically used for acquiring power traces or electromagnet traces or any other side channel traces that we essentially are interested in. And upon receiving so, there is essentially a triggered signal which the design generates say upon you know like when every encryption is done. So, after every encryption there is a trigger which is generated. So, upon receiving this triggered signal power traces are obtained and they are sent to the CPU.

So, the triggering essentially indicates kind of the start of the encryption and therefore, it helps us to align the power traces. So, that every power trace corresponds to the same set of operations ok. If there is a misalignment then that would lead to lesser success rate I mean it can essentially lead to the fact that the attack will not work also. So, it is very important that we properly align the traces and triggering helps us to do so.

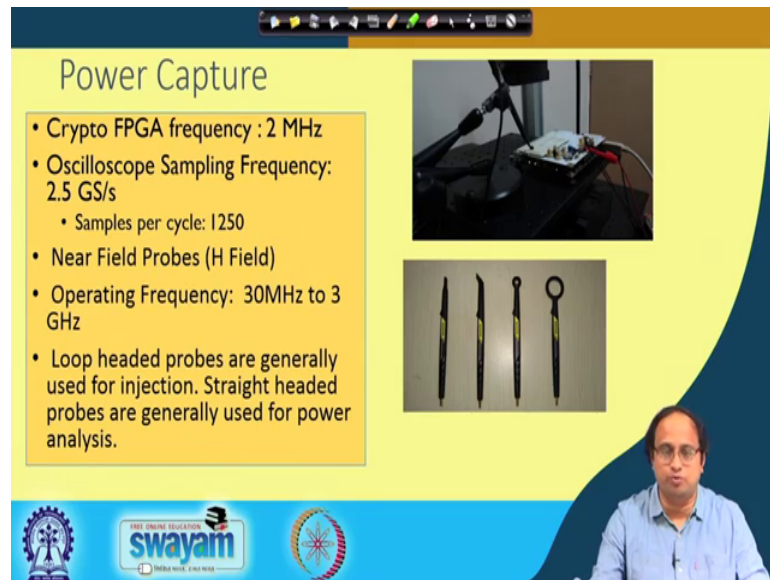
Now, the oscilloscope also right needs to have a high sampling rate, because as we know that if the sampling rate is low then it will miss information and that essentially could be crucial for the attack. So, essentially it has got a high typically it has got a high sampling rate and that is an important criteria for you know like getting the suitable getting a suitable oscilloscope. Then the heart of this entire operation is the CPU ok. The CPU or the on your computer essentially is where the whole set up along with the other you know the other components are essentially synchronized.

So, therefore, right the CPU essential the CPU right as you can see over here is the PC which provides the plaintext to your target board. And it receives the corresponding cipher text and at the same time right it also receives the corresponding power trace. So, in a typical attack like when you are not considering side channels the attacker would probably have access to the plaintext and definitely the cipher text, but here you also observe the side channel information which is the power trace value. So, essentially it is the power consumption trace.

Now, the power consumption is obtained by essentially by the oscilloscope, when the oscilloscope receives a triggered signal from this device. And then it starts kind of observing the corresponding signals and essentially in this case the power signals and these are transported back to the PC. So, to the PC kind of maintains a repository of the

plaintext the cipher text and also the corresponding power trace ok and it is nicely synchronized by the PC. So, the PC essentially plays a very pivotal role in doing this organization or the synchronization.

(Refer Slide Time: 05:22)



The slide is titled "Power Capture" and contains the following text:

- Crypto FPGA frequency : 2 MHz
- Oscilloscope Sampling Frequency: 2.5 GS/s
 - Samples per cycle: 1250
- Near Field Probes (H Field)
- Operating Frequency: 30MHz to 3 GHz
- Loop headed probes are generally used for injection. Straight headed probes are generally used for power analysis.

The slide also features two images: one showing a probe connected to a circuit board, and another showing four different types of probes (two straight-headed and two loop-headed). At the bottom of the slide, there are logos for "swayam" and other educational institutions.

So, the power capture essentially will require you to do have some very vital component of your setup which is that the probes. So, there can be different kind of probes for example, you can see here some pictures of probes. So, you can see that there are some probes which has got straight heads and there are some probes which has got circular heads. Typically these probes are used for injections and for analysis; that means, for example, when you are doing say power analysis, you will be using these probes which has got straight heads ok.

And the operating frequency is also very vital for this probes; that means, the probes typically should have a high frequency allowance; that means, it should be able to pass the high frequency components. Typically your power traces for these circuits can have high frequency components and you do not want to miss those information, because if you miss it right you know that you are not getting the complete information about the power trace.

So, therefore, the power trace typically has an has got in our case an operating frequency of around 30 megahertz to 3 gigahertz and these are all based on near field probes. So, these are near field probes means they are based on magnetic fields. So, therefore, I

mean in another setting right you can actually leap up the line between the voltage or the VCC and your set up and you can implant or put in a resistor and you can observe the current through the resistor. Alternatively right you can actually use these kind of probes through which you can using the magnetic flux right you can observe the corresponding power consumptions.

So, the other as I said the other important component or aspect is the sampling frequency of your oscilloscope. So, typically we said the oscilloscope sampling frequency as high as 2.5 giga samples per second, which you can see that if your crypto FPGA frequency for the SASEBO or the SAKUDA board typically occur operates at 2 megahertz clock. So, if you see 2.5 giga samples per second 2 megahertz that implies that the samples per cycle is around 1024 exactly 1250. So, this you can figure out from these 2 data right that the sampling the samples the number of samples per cycle is 1250.

So, typically right I would like that this value should be high. So, that you know like we get large we do not miss any vital information about the power consumption in the trace. So, therefore, we require a high sampling an oscilloscope with a high sampling frequency.

(Refer Slide Time: 07:50)

The Design Under Test

Rolled AES architecture
AES-128, though can be extended to any parameter. AES-128 implemented on Virtex-5 FPGA (xc5v1k50), has a **trigger signal** which aligns the power traces, so that each trace profile corresponds to the same sequence of operations in an encryption run. The signal is tapped from the controller pin of the FPGA board and is fed to the second channel of the oscilloscope. The plaintext input is fed from the PC and the ciphertext is fed back to the PC for correctness check. The power trace from the board is acquired by the oscilloscope and the power trace values are stored in the PC.

The diagram shows a data flow from a 256-bit Plaintext and 256-bit AddRoundKey input through a MIX block to SubBytes, then through a 3-stage pipeline (BEFFER1, BEFFER2, BEFFER3) with MixColumns and AddRoundKey blocks, resulting in a 256-bit Ciphertext. A trigger signal is shown tapping off from the BEFFER1 stage.

Logos for Swamyam and other educational institutions are visible at the bottom of the slide.

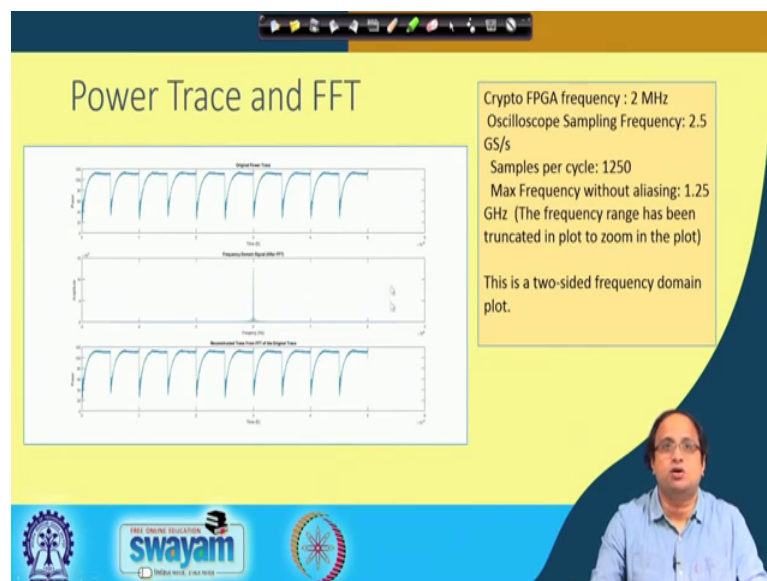
So, here is my design under test. So, of course, you can you can change the target, but let us consider this as our target. So, this is our rolled AES architecture which we already discussed in our previous classes. So, the AES right essentially is also in this case only

the AES 128 version, but you can easily extend it to other dimensions of the AES algorithm. And the AES in this case is implemented on Virtex 5 FPGA and this is the device family of the FPGA. And it has got a triggered signal as I said that the triggered signal will basically indicate that the operation has been done after every encryption.

So, it so, this triggered signals objective is to align the power traces. So, that each trace profile corresponds to the same sequence of operations in an encryption run ok. The signal is tapped from the controller p pin of the FPGA board and is fed to the second channel of the oscilloscope ok. And it kind of gives you a reference. So, that you can basically align the all the traces, because when you are doing a power attack typically you can have like 1000s of traces or 1000s to millions of traces. So, you need to properly align these traces and the trigger essentially is handy in that context.

So, the plaintext input is fed from the PC and the cipher text is fed back to the PC for correctness check. Of course, you need to ensure that your design is working properly as expected by the in the AES algorithm. And the power trace from the board is acquired by the oscilloscope and the power trace values are stored in the PC. As, I said there are power trace values are also even recorded in the PC because there we will do our statistical processing or DP attack.

(Refer Slide Time: 09:24)

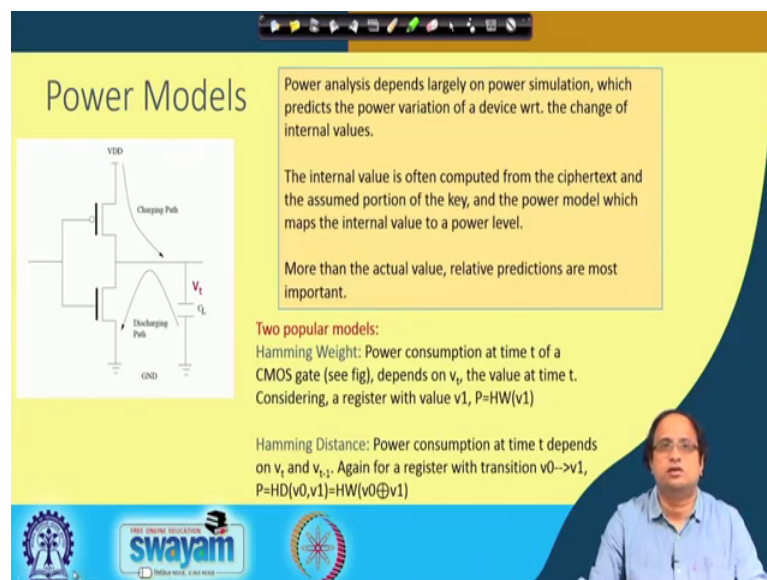


So, here is how the power trace of an AES algorithm looks like as I said that crypto FPGA frequency is 2 megahertz the and you are basically collecting like 1250 samples

per cycle. So, we need an FFT plot of this the objective was to show like ah. So, basically when we are doing this FFT analysis right we is ensure that a maximum frequency without aliasing is kept around 1.25 gigahertz.

So; that means, like if this trace right has got some frequency component which is more than 1.25 gigahertz, then that is lost ok. And when we do an inverse Fourier transformation, then we get back the original power trace kind of showing that you do not have any component which is more than 1.25 gigahertz in this trace. So, this is a 2 sided frequency domain plot and you can observe that essentially there is a central you know like the DC bias of this power trace ok. And that is essentially shown by this 0 frequency component of the FFT curve or the FFT plot.

(Refer Slide Time: 10:27)



Power Models

Power analysis depends largely on power simulation, which predicts the power variation of a device wrt. the change of internal values.

The internal value is often computed from the ciphertext and the assumed portion of the key, and the power model which maps the internal value to a power level.

More than the actual value, relative predictions are most important.

Two popular models:

Hamming Weight: Power consumption at time t of a CMOS gate (see fig), depends on v_t , the value at time t . Considering, a register with value v_1 , $P=HW(v_1)$

Hamming Distance: Power consumption at time t depends on v_t and v_{t-1} . Again for a register with transition $v_0 \rightarrow v_1$, $P=HD(v_0, v_1)=HW(v_0 \oplus v_1)$

The slide features a diagram of a CMOS inverter with labels for VDD, GND, Charging Path, Discharging Path, v_t , and C_L . It also includes logos for IIT Bombay and Swayam.

So, let us before I go into the other aspects that we essentially I will be discussing about the high frequency component of the power signals and how they should be communicated to the oscilloscope? But, before that let us discuss about the power models which is a very vital aspect of doing your differential power attacks. So, power model typically means it is basically a simplistic model through which you basically kind of estimate or I would say rather guesstimate the corresponding power consumption.

So, you see that you know like you have got in this case I will just to explain that I show a picture of an simple CMOS inverter and you can see that there is essentially a PMOS

and NMOS basically stacked to each other the objective is to charge and discharge the capacitor.

So, the capacitor essentially stores the value of the output of this gate and when it is charged right it is charged to the PMOS and when it gets discharged it gets discharged through the NMOS. So, there is a distinct charging path and the discharging path ok, that implies that the power consumption should be different probably for the charging as well as the discharging ok. But the but before I even get into that right I mean we should try to understand why do we need the power model in our attack?

So, if you remember when we were doing the difference of mean based attack the objective was that, you had the access to the cipher text as an attacker you were guessing a portion of the key. And based upon that you were trying to kind of you know like go back to an intermediate state to go into intermediate state. And there you are basically applying a power model ok and the objective of that power model would be to get some kind of estimate about the actual power consumption.

So, power analysis depends largely on this simulation of power ok, because you have got an actual power and you try to simulate the power and then you try to correlate these 2 things ok. And the idea is that if your key guessing was correct or at least the guess for the portion of the key was correct, then the simulation would match with your expected or with your observed power consumption that is the basic idea.

So, power analysis depends largely on powered simulation, which predicts the power variation of a device with respect to the change of internal values. The internal value is often computed from the cipher text and the assumed portion of the key and the power model which maps the internal power to a powered level. So, this could be for example, a hamming weight or hamming distance model ok, there could be other models also.

So, more than the actual value what is more important is the relative predictions? Like whether the power you know like increases or the power decreases, because I am not really bothered about the actual power often ok. I am more bothered about the relative ordering of the power consumptions.

So, therefore, by the unit is not so, important what is important is the relative ordering ok, whether the power consumption is increasing because of a different data or whether

the power consumption is decreasing because of another data. So, very two popular models which is often will very used very usually used ok, very commonly used are the hamming weight and hamming distance models.

So, if you see the figure the hamming weight essentially means that the power consumption depends at a time t of a CMOS gate in this case and inverter depends upon V_t ; that means, the voltage with this capacitor is holding ok. It does not nearly bother in this case in this model about what was the previous state of the what previous charge of this capacitor. So, therefore, I can kind of simulate the power by taking the hamming weight of say v_1 .

So, if we consider that if you stack these inverter in say you know like n number of times and if you get a register for example, then there you know like if you have got say the register is typically you know like a value which has got an n components ok. And each of these components can be say either a 1 or a 0 ok. So, you are basically an n bit register. So, in this case in this model basically the idea is that the power consumption is dependent upon the corresponding hamming weight; that means, the number of ones in that n bit register.

On the other hand if you have a hamming distance model then; that means, that the power consumption depends upon the hamming distance between this current state and the previous state. So, that would imply that if you have got a register which makes a transition from say v_0 to v_1 , then the power is modeled by the hamming distance of v_0 and v_1 , which is nothing, but the hamming weight of v_0 exhort with v_1 ok, that basically counts the number of switches ok.

So; that means, like in this case what I am you know the other, but the power consumption at time t depends upon not only in this current time voltage, but also on the previous value of the voltage ok. So; that means, like it is more probably more realistic because you know like when you are considering dynamic power consumption, then you would expect that hamming distance model probably would be more accurately estimating the power consumption.

At the same time right also keep in mind that when you are doing a hamming weight or I mean if you are comparing the hamming weight power model with a hamming distance power model. In the hamming distance power model, you also need to know what was

the previous state of the resistor ok? Which means that you need more information. And sometimes right that may not be you know like I mean it may not be so, easy to work with the hamming distance model.

But at the same time we can try to if you observe right you will see that the hamming weight model also does not do so, bad ok. One of the reasons like in certain category of implementations, as we know that in CMOS right in particularly when we talk about dynamic CMOS, we have got certain category of circuits which gets occasionally charged I means like pre charged. So, there is a pre charge stage followed by evaluate state.

So, in the pre charge state typically you basic you know like you can either charge the state to 0 or you can charge the state to 1. So, let us consider suppose you charge it to 0 or 1, it does not matter, that implies that in the previous cycle you know like you have a constant ok. And the in that case right a hamming weight model will work as good as a hamming distance model until you will see that these kind of circuits are not. So, you know they are they are in quite abundance actually like, typically microcontroller circuits right not smart card circuits may have a pre charge logic. And therefore, in such context hamming weight model may work pretty good pretty well.

Even in other context where you are not doing a pre charge ok. Suppose, you are considering a cryptographic circuit for example, in a cryptographic circuit often the previous state is randomized right. You basically have a random data and in that in this said the t th clock cycle you are doing a computation, which is probably dependent upon the key and you want to kind of sneak out that dependence by using a power attack.

So, in that context you can also keep in mind that as I said that right I mean the power consumption is not really you know like. So, in that case right if the power consumption was uniform; that means, if the power consumption right was whether you are going from a 0 to 1 state or you are you know like going from maybe you know like a 1 to 0 state. If they were roughly equal, then you probably could not have worked with the hamming the hamming weight model ok. But on the other hand right if you have a hamming if you have this skewness in the power consumption; that means, when you are going from, when we are charging to a 0 state or when you are charging to a 1 state, if

they are dependent like if there is a bias right. Then that would imply that it still correlates with the corresponding hamming weight at the time t ok.

And; that means, right even in that context right a hamming weight model may work pretty well, you know even if you are not applying the hamming distance model ok. But all set and done definitely the hamming distance model is probably more accurate, but a hamming weight model also would probably work quite well and it is therefore, that we often use both in over attacks.

(Refer Slide Time: 18:14)

The slide, titled "An Example Model", is divided into several sections. On the left, a circuit diagram shows an LFSR with 8 bits and a feedback function block. Below it, a text box states: "A Linear Feedback Shift Register (LFSR), and a nonlinear function is implemented on Xilinx FPGA platform XC3S400-5PQ208 device". To the right of the diagram is a large graph showing "Actual power traces taken as voltage drop. The power traces for 80 consecutive clock cycles after de-asserting reset". Below this are two smaller graphs: (a) "Hamming Distance plot of the implemented stream cipher" and (b) "Hamming Weight plot of the implemented stream cipher". A text box on the right says: "Power Simulation using Hamming Weight and Hamming Distance Models". At the bottom left are logos for "swayam" and "MHRD". At the bottom right is a small video inset of a man speaking.

So, here is an example modeling of an LFSR or a Linear Feedback Shift Register, which is implemented along with the non-linear function. So, this is typically how a stream cipher is implemented you have an LFSR block you have a non-linear feedback shift Boolean function to basically which works as the Boolean combiner. And then you have got the plaintext it basically generates a key stream you XOR them and you get the cipher text.

So, this is typically how a stream cipher would probably look like. So, what we do here is that we basically implement on a Xilinx FPGA device and try to observe the actual power taken as voltage drop the power traces for 80 consecutive clock cycles after you de asserting are doing the reset. So, basically initially reset and then allow the stream cipher to generate key streams and you also try to capture the corresponding power consumption.

So, if you see right what we do is we basically observe we basically make what we do here is we just give the transition set 0 to 0 in a and consider all the 4 transitions in b; like it can make 0 to 0, it can make a transition from 0 to 1, it can make a transition from 1 to 0 and it can make a transition from 1 to 1. So, you can observe that if this makes a transition from 0 to 0. So, basically the idea is very simple like, what am trying to do here is that I am trying to basically take a gate. So, here you have got an AND gate ok. And the AND gate has got 2 inputs say a b and the output is y.

So, what we are doing is we are basically considering all possible transitions in a. So, the a for example, can make so, I also write a 0 to 0 transition as 0 to 0 or it can go from 0 to 1 or it can go from 1 to 0 or it can go from 1 to 1.

So, there are 4 possible transitions for a ok. Likewise there are 4 possible transitions for b ok. So, you basically combine all of these possibilities and therefore, right we have got 16 possibilities of 16 combinations right. So, imagine that a is making a transition from 0 to 0. So, if a is making a transition from 0 to 0 just to explain the first line ok, so, a is making a transition from 0 to 0 and b is making a transition from 0 to 0.

So, therefore, it immediately implies that the initially write your a was at the output was at 0 and it remains at 0. So, the y remains at 0 likewise if you consider another case like suppose let us consider at this case ok.

So, in this case we basically observe that suppose a makes a transition from one 1 to 1; that means, it was 1 and it remains at 1 state and b makes a transition from 0 to 1; that means, initially the and right was at logic 0 and after this clock cycle right it becomes logic 1. So, therefore, it goes from 0 to 1.

So, like this right we are basically kind of observed all the 16 possible transitions and then we basically allow write the corresponding energy for these transitions like suppose this gate is making a transition from 0 to 0. So, we basically kind of write or tabulate the energy consumption as $E_{0 \text{ to } 0}$, likewise here we write $E_{0 \text{ to } 1}$.

Similarly, right in some cases we will have $E_{1 \text{ to } 0}$ and we also have $E_{1 \text{ to } 1}$. So, these are the 4 energy you know like energy levels that we are considering. So, now, what we try to do is we basically try to calculate the average energy, when the output is 0 and the average energy, when the output is 1. So, if you remember in the difference of mean we

exactly try to do the same right. We basically try to find out the average energy consumption in both the cases.

So, we basically try to find out what is the average energy when the output gets converted into 0; that means, get gets computed into 0 and the average energy when the output gets calculated as 1. So, therefore, we find out $E_{q \text{ equal to } 0}$ and $E_{q \text{ equal to } 1}$. So, $E_{q \text{ equal to } 0}$ are those cases where you have got the output which results in 0.

So, you can observe that this is such a case likewise this is a case where the output becomes 0, because likewise for all these cases we have got the output which results in 0. Likewise here also the black case also you have got a 0, which is the output results in 0, but this one right is the $E_{0 \text{ to } 1}$ transition; that means, the gate becomes one after the transition.

So, therefore, we will not put this in this bin and this is another case to be considered here likewise these 2 are also important and we also have $E_{1 \text{ to } 0}$ which is essentially shown here. So, all these are you know like cases which we are considering here because the output gets transformed into 0 ok. So, here you can observe that there are 9 $E_{0 \text{ to } 0}$ transitions ok, like 1 2 3 4 5 6 7 8 and 9. So, there are 9 $E_{0 \text{ to } 0}$ transitions and there are 3 $E_{1 \text{ to } 0}$ transitions like these blues are the $E_{1 \text{ to } 0}$ transitions ok.

So, therefore, if you average you will get divided by 12, likewise right for $E_{q \text{ equal to } 1}$, there are 3 $E_{0 \text{ to } 1}$ transitions and there is 1 $E_{1 \text{ to } 1}$ transition. So, the average is essentially shown here. And you can observe that if the transitions like $E_{0 \text{ to } 0}$ 1 to 0 1 to 1 and 0 to 1 ok, they are not identical, then $E_{q \text{ equal to } 0}$ and $E_{q \text{ equal to } 1}$ they are not same ok. And that implies that you do not have the same level or average power and the power essentially correlates with whether the output is 0 or whether the output is 1 ok. And therefore, right this is an important or you know I would say like a simple experiment, but it kind of shows that you know like that there is a correlation.

So, therefore, this correlation is essentially one of the reasons why you have the gates leaking in context to power attacks.

(Refer Slide Time: 25:52)

The slide is titled "Quality of Measurement and Noise" and features a yellow background. At the top, it states: "High quality of power traces captured is central to the accuracy of DPA." Below this, there are two text boxes. The first box explains: "AES-128 Encryption with same plaintext and key, but resulting in different power traces." The second box details: "These fluctuations are due to electrical noise, caused by noise due to power supply, clock generator, conduction and radiation emissions from the components connected to the device under attack." To the right of these boxes is a graph showing multiple overlapping power traces over time, illustrating the variability caused by noise. At the bottom of the slide, there are logos for "swayam" and "INDIA RISE, EDUCATION RISE" along with a small video feed of a man in a light blue shirt.

So, likewise right I mean the other important thing is you know the quality of measurement and noise and it you can observe here that, you know like and you can easily point out that the high quality of power traces is very important and central to the accuracy of differential power attack. So, here in this case we do a very simple experiment we basically run AES encryption with the same plaintext and the key, but as you can see that the power traces are varying ok.

And it results is resulting in different power traces, but if you want write that that your you know like that your power capture is accurate, then you would like to have this variation as small as possible. And this variation essentially takes place because of the noise which is essentially there in your in your implementation.

So, these fluctuations are typically caused due to electrical noise, which are caused by noise due to power supply clock generator conduction and radiation emissions from the components connected to the device under attack ok. And therefore, this is an important aspect that we need to consider when we discuss about power attacks. So, I will be discussing more about you know like about noise and also about how to kind of analyze noise using statistical tools, but that will take up in the next class.

Thank you.