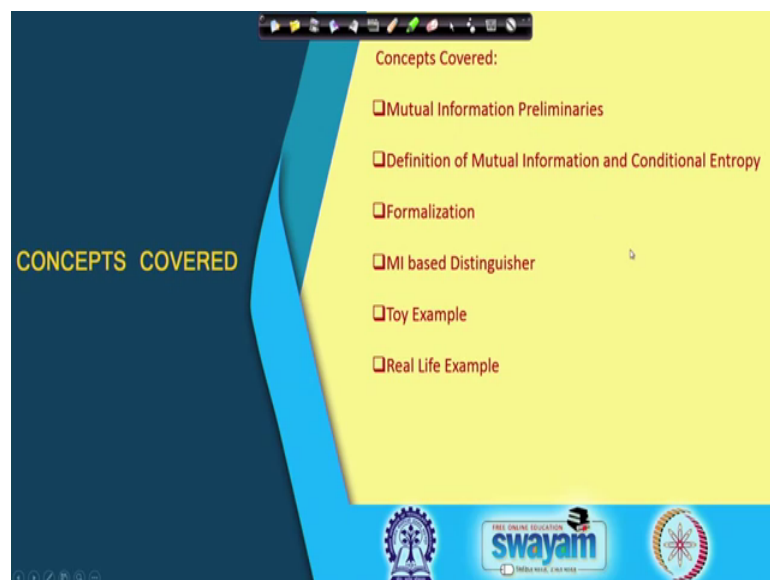


Hardware Security
Prof. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 39
Power Analysis – XV

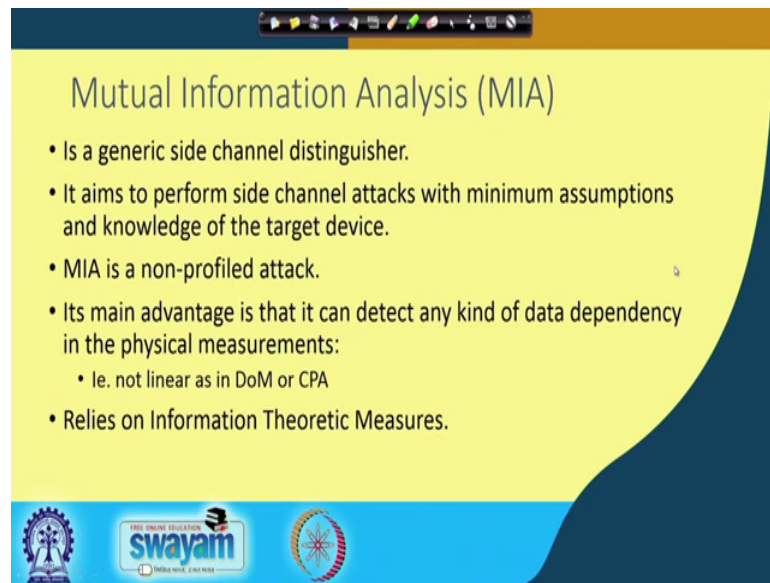
So, welcome back to this class on Hardware Security. So, we shall be counting our discussions on power attacks and in particular today will shall be discussing about the topic which is called as mutual information analysis.

(Refer Slide Time: 00:31)



So, we shall be trying to see how mutual information analysis can be developed. So, we will basically start with the preliminaries behind mutual information or MI as it is abbreviated. We shall try to define mutual information and also conditional entropy, which is fundamental to the understanding of this technique. We shall formalize this with respect to side channel analysis and side channel decay and we shall define what is the MI based distinguisher. And then I will be discussing about couple of toy examples and real life examples to illustrate how it works in practice.

(Refer Slide Time: 00:59)



The slide is titled "Mutual Information Analysis (MIA)" and features a yellow background with a dark blue curved border on the right side. At the top, there is a navigation bar with various icons. The main content consists of a bulleted list of characteristics of MIA. At the bottom of the slide, there are three logos: the Swamiji logo on the left, the "swayam" logo in the center, and a circular logo on the right.

Mutual Information Analysis (MIA)

- Is a generic side channel distinguisher.
- It aims to perform side channel attacks with minimum assumptions and knowledge of the target device.
- MIA is a non-profiled attack.
- Its main advantage is that it can detect any kind of data dependency in the physical measurements:
 - I.e. not linear as in DoM or CPA
- Relies on Information Theoretic Measures.

Logos at the bottom: Swamiji, swayam (Free Online Education), and a circular logo.

So, to start with mutual information analysis or MIA as it is abbreviated is a general or generic side channel analysis distinguisher. So, like we have seen CPA and difference of mean which we essentially like side channel distinguishers which are build around the Pearson's correlation coefficient. So, here we try to apply the idea or theory of information theory and try to developed a statistical tool for doing power attacks.

So, the objective is to perform the side channel attacks with minimum assumptions and knowledge of the target device. So, for example, like when we were applying the CPA or difference of mean then as I said that one of the reasons why the attack works well is because we basically make tacit assumption that the leakage is underline the underlined leakage is linear ok.

So, likewise right I mean and it is more like the correlation coefficient is more appropriate when the leakage is linear ok. But in this case what the objective is basically since it is more fundamental in its way the MI tool is developed. We basically can work with minimum assumptions and knowledge about the target device. So, MIA is also like the DP and the DoM techniques right or the CPA technique is basically a known profile attack; that means, it does not have any profiling phase as we have seen in the context of template attacks ok.

And its main advantage is that it can detect any kind of data dependency in the physical measurements of k . So, that is not necessarily linear as we have seen in the context of DoM or CPA and it relies on very strong fundamentals on information theory ok.

(Refer Slide Time: 02:33)

Preliminaries on Mutual Information

- Let X be a random variable on a discrete space \mathcal{X} , and x is an element from \mathcal{X} .
- The Shannon entropy of a random variable X on a discrete space \mathcal{X} is a measure of its uncertainty during an experiment:

$$H[X] = -\sum_{x \in \mathcal{X}} \Pr[X = x] \cdot \log(\Pr[X = x])$$
- The joint entropy of a pair of random variables (X, Y) expresses the uncertainty one has about the combination of these variables:

$$H[X, Y] = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[X = x, Y = y] \log(\Pr[X = x, Y = y])$$

So, it is important to make a quick recapitulation about how or what are the tools that we have in account in information theory. So, for example, let X be a random variable which is defined on a discrete space denoted as \mathcal{X} and x is an element from this space. Then we can define the Shannon entropy of a random variable on a discrete space essentially in this way.

So, it basically captures the amount of uncertainty which is essentially basically it is a measure of uncertainty during an experiment uncertainty in this random variable ok. So, for as I know that if I have got n bit of data for example, then there are n bits of uncertainty or n bits of unknown the there for the entropy right. In the if I said you that the random variable can take any possible n bit value; that means, there are 2 to the power of n possible values, then the entropy of that variable is n So, n bits in general.

So, here we basically try to give a general formulation of how the entropy can be calculated. So, essentially its probability of X equal to x ; that means, the probability that the random variable X takes the value of small x multiplied with the logarithm with respect to base 2 of the probability of X equal to x . And note that since it is a logarithm right and therefore, the values being lesser than 1. The probability being less than 1; the

logarithm is negative. So, we kind of cancel that by a negative sign and that is essentially the formula of how to calculate right HX.

So, likewise we can define the joint entropy, I mean essentially when we have got a pair of random variables like X and Y. And therefore, this expresses the uncertainty one has about the combination of this variables ok. So, for example, it may happen that there are two jointly distributed random variables and some of them never occurs actually ok. So, therefore, that reduces the entropy of their joint or rather it reduces the joint entropy. So, what we basically say or the way to measure H X, Y is a natural extension of this formula is that you replace this probabilities by their joint probabilities.

So, you have got the same way we can formulate now you have got probability of X equal to small x comma Y equal to small y that is the joint probability distribution. Likewise in the long also we have got probability of X equal to small x, Y equal to small y, but now the sigma is taken over both X and Y. So, there are is basically a double sigma written as single sigma. So, this is essentially how we can calculate the joint entropy of a pair of random variables n.

(Refer Slide Time: 05:03)

Conditional Entropy and Mutual Information

- $H[X|Y] = \sum_{y \in Y} \Pr[Y = y] H[X|Y = y]$
- $H[X|Y] = -\sum_{y \in Y} \Pr[Y = y] \sum_{x \in X} \Pr[X = x|Y = y] \log(\Pr[X = x|Y = y])$
- $H[X|Y] = -\sum_{x \in X, y \in Y} \Pr[X = x, Y = y] \log(\Pr[X = x|Y = y])$

Diagram illustrating Conditional Entropy and Mutual Information:

The diagram shows two overlapping circles representing sets X and Y. The regions are labeled as follows:

- $H[X]$: Entropy of X (left circle)
- $H[Y]$: Entropy of Y (right circle)
- $H[X|Y]$: Conditional entropy of X given Y (left part of X)
- $H[Y|X]$: Conditional entropy of Y given X (right part of Y)
- $H[X,Y]$: Joint entropy of X and Y (intersection of X and Y)
- $I[X;Y]$: Mutual information (intersection of X and Y)

Handwritten note: $H[X] = H[X|Y] + I[X;Y]$

So, now we would like to define a very important definition of what is called as a conditional entropy ok. So, how can we measure conditional entropy? So, the easiest way probably you know like to formulate conditional entropy is probably to think of in this way like let us take assume that this is Y that is the condition Y essentially takes a

specific value say y ok. So, we get the Y equal to small y and then we basically multiply the entropy of X given Y equal to small y with the probability that Y takes y . So, basically like this from you can say that we are applied the kind of the law of total probability and we are measuring this entropy ok.

And now what we do is so, it is not specifically total probability as such, but we are basically trying to find out the total entropy by you know like varying Y over all possible values of Y . So, now, the we would like to calculate H of X given Y equal to small y ok. So, this H of X given small y can be easily calculated therefore, now note that Y equal to small y right.

So, therefore, this is basically I am kind of fixing the value of the condition. The moment I fix the condition right essentially I can write the corresponding description of this entropy in exactly the same way as we wrote the entropy in the previous case. So, now, we will write probability of X equal to small x because this X can take some value say small x condition on Y equal to small y multiplied with the log of probability of X equal to small x given Y equal to small y .

Now, note that these two things can be multiplied and you know like from the definition of conditional probability. We can write as a product of X equal probability of X equal to small x given Y equal to small y multiplied with the probability of Y equal to small y is nothing, but the joint probability distribution of probability of X equal to small x and Y equal to small y ok. And therefore, right this is the corresponding if you note again that the sigma is a double sigma which is a varying over both X and Y . And a very easy way or I would say like a good way of remembering how the entropies are in relation to them is in the form of this Venn diagram ok.

So, if you see this Venn diagram right this is your H_X ok, this is your H_Y and note that this region is $H_{X|Y}$. That means, you know like that if I tell you that this is your H_X space and I tell you that the amount of. So, suppose I want to calculate H of X given Y ; that means, since you know that you know like that Y . Suppose you have the information of Y so, the amount of remaining information which is there is given by this portion that is why this stands for H of X given Y likewise this portion stands for H of Y given X . And you can easily note that we can write you know like if I want that H of X, Y , then that is given by this total region.

So, you can know that if I add $H(X)$ and add $H(Y)$ then it is more than $H(X, Y)$. So, therefore, these kind of relationships becomes quite evident if you think of this Venn diagram. In particular we shall be interested about this region and this region is called as the mutual information of X and Y .

So, therefore, you can easily see that $H(X)$ right essentially which is this total entropy is nothing, but so, we can write that $H(X)$ is equal to $H(X|Y)$ plus the mutual information of X and Y ok. In other words the mutual information of X and Y is equal to nothing, but $H(X)$ minus $H(X|Y)$ similarly you can write with respect to Y as well. So, therefore, this gives an kind of an easy way to kind of remember or understand how the relationships between these you know like this parameters are.

(Refer Slide Time: 08:49)

Mutual Information

$P_r[Y=y|X=x], P_r[X \neq x]$

$$I(X:Y) = \sum_{x \in X, y \in Y} P[X=x, Y=y] \log \left(\frac{P[X=x, Y=y]}{P[X \neq x] \cdot P[Y=y]} \right)$$

$$= \sum_{x \in X} P[X=x] \sum_{y \in Y} P[Y=y|X=x] \log \left(\frac{P[Y=y|X=x]}{P[Y=y]} \right)$$

- Mutual Information is a general measure of the dependence between two random variables.
- It expresses the quantity of information one can obtain about X by observing Y .

$I(X:Y) = H(X) - H(X|Y)$
 Using, $H(X,Y) = H(Y) + H(X|Y)$,
 $I(X:Y) = H(X) + H(Y) - H(X,Y)$

$H(X)$ (left circle)
 $H(Y)$ (right circle)
 $H(X,Y)$ (top label)
 $H(X|Y)$ (left part of intersection)
 $I(X:Y)$ (intersection)
 $H(Y|X)$ (right part of intersection)
 $H(Y)$ (bottom label)

So, therefore, mutual information is a general measure of the dependence between two random variables. It expresses the quantity of information one can obtain about X after observing say Y . So, if I measuring say $I(X:Y)$, it is a amount of information right which you are essentially measuring about say the random variable X by observing Y . Since $I(X:Y)$ is equal to $I(Y:X)$, you can actually define in a commutative way also.

So, you can say this is the amount of information about Y which you are observing by say X . So, therefore, right we can write $I(X:Y)$ is equal to $H(X)$ minus $H(X|Y)$.

given y that is what we have seen and therefore, right using the result that $H(X : Y) = H(Y) + H(X|Y)$.

So, you can note that $H(X)$ you know like $H(X, Y)$ you can write as if like it is I am adding $H(Y)$ with $H(X|Y)$ right. So, therefore, you can also write that $H(X : Y) = H(X) + H(Y) - H(X, Y)$. So, now, I would like to measure these mutual information in terms of the various probabilities actually ok.

So, now, let us see how we can do that. So, $H(X : Y)$ you can also define it in this way. So, you can write this as nothing, but the this again is a sigma which is taken over both X and Y and you are basically multiplying you know essentially you have got $P(X=x)$ that is x small x comma $Y=y$ equal to small y .

The logarithm of the probability of $X=x$ comma $Y=y$ divided by the probability of $X=x$ and probability of $Y=y$. And this essentially is nothing, but the sigma of probability of $X=x$ and you know you can split this probability of $X=x$ comma $Y=y$ as probability of $Y=y$ equal to small y .

Given $X=x$ and then multiply the probability of $X=x$ and this part you can also write in this way by you know like elaborating this probability of $X=x$ comma $Y=y$ in terms of this. That means, you know like you can write this in this way because if you if you want to a kind of elaborate this part, then you can write this part as probability of $Y=y$ given $X=x$ and this is nothing, but and this you are basically multiplying with probability of $X=x$. So, this essentially is equivalent and therefore, this part will cancel with this part and therefore, you will have probability of $Y=y$ given $X=x$ divided by probability of $Y=y$.

So, this is the way in which you can directly calculate the conditional probabilities and from that you can estimate the value of this mutual information ok. You can calculate the value of mutual information.

(Refer Slide Time: 11:43)

MIA for Side Channel Key Recovery

Consider a device performing several cryptographic computations being denoted as $E_k(p)$, where p denotes the plaintexts for a fixed key k .

The device under attack manipulates a target computation and updates a sensitive variable denoted as $V_{s,p}$, for a fixed input P .

Note that the actual leakage because of this sensitive computation is $Y_{k,p}$, where k is a secret key which includes the subkey s .

Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, Nicolas Veyrat-Charvillon: Mutual Information Analysis: a Comprehensive Study. J. Cryptology 24(2): 269-291 (2011)

So, now, with this background let us see and try to see how we can apply MIA for side channel key recovery. So, here is one very fundamental paper from published in journal of cryptology in 2011. So, this diagram is taken from this paper. So, you can observe that what we try to do here is there are certain random variables which are being defined here.

Ah One way to kind of conceptualize this is by considering that there is a device which is performing several cryptographic computations and you denote that by say $E_k(p)$ ok. So, $E_k(p)$ stands for your encryption implemented and it is processing on p which is the plaintext and there is a fixed k key which is k , which you want to determine by your attack. Now the device under attack manipulates a target computation like we have seen in context to the other $d(p)$ attacks, there is a target computation. So, it could be the output of an s box it could be the input of an s box and so on and so forth. So, you there is a the device under attack manipulates a target computation and updates a sensitive variable.

This sensitive variable is denoted as a $V_{s,p}$. So, in this case you know like this is $V_{j,p}$, but you can think of that as some $V_{s,p}$ value ok. The $V_{s,p}$ for a fixed input P ok. So, note that the actual leakage because so, the actual leakage right. So, this is the, you know like the actual computation. So, this is $V_{s,p}$; so, so $V_{s,p}$ is the. So, there are two parts in the device. So, you see like when we are doing the thinking of the mutual information attack, then the attack essentially has got

two parts like as we have seen in other part attacks also. There is one part where you are basically having the device are you know like the device means the actual device which is processing the actual secret key ok .

The secret key is essentially denoted as say small key. This is the correct key this is the actual and when you are computing this, then there is an intermediate target variable denoted as V_s, P . Now this V_s, P although this is the target variable. So, this V_s, P denotes like essentially it is a specific target which we are choosing, but the leakage essentially is denoted as Y_k, P this is an observable. So, this we called as the observable ok . So, this is the observable leakage often we denote it as O . So, note that although you are probably processing on a part of the key like s probably or part of the state.

But when I am trying to I am writing the leakage then the leakage essentially is you know like is essentially k denoted by k where k is actually a secret key which includes the sub key s ok . So, therefore, it is including the sub key s . So, therefore, the total leakage will actually not happen only because of the part of the key, it will happen because of the part of the key along with the remaining parts of the key also. So, that total key is denoted as small k out of which I am interested in a part of the key because I am doing a divide and counter based attack.

On the other hand when you are talking about the adversary that means when you are doing the attack basically you do not know what is the key. So, you try to guess the key right. So, therefore, what you do in the attack you basically you try to guess the key and you try to do a prediction ok . So, you basically try to make a prediction and you derive something which is called as the leakage ok .

So, you basically find out a leakage vector and this is essentially denoted as L . Now this leakage again depends upon the guess which you are doing. So, there is a guess of the key that you are doing and this guess essentially is kind of mapped by some function into a leakage. So, this guess let me denote it as say k^* for example. So, therefore, the leakage will parameterize by k^* . So, therefore, I call that as L, k^* .

And then my objective will be to define a distinguisher which will tell me whether the guess is correct or not ok . So, the guess right k^* could be say j and I want to find out whether these j matches with this is d or not. So, s is the part of the key and the attack is

successful if j is equal to s ; that means, if you are basically guessing correctly a part of the key again remember that the leakage right does not depend upon the part of the key, but the leakage depends upon the total. So, therefore, right let us see how we can do this and we will try to kind of you know like understand the working principle of the attack by essentially defining the following steps.

(Refer Slide Time: 16:03)

The slide is titled "MIA for Side Channel Key Recovery". It features a flowchart on the left and explanatory text on the right. The flowchart shows two paths starting from a common point P . The top path, labeled "Adversary", starts with a guess j (in a circle) which is "predicted" to a hypothetical sensitive state $V_{j,P}$ (in a circle). This state is then processed by a "model" to produce an estimated leakage $X_{j,P}$ (in a circle). The bottom path, labeled "Device", starts with a key k (in a circle) which is "computed" to a sensitive state $V_{k,P}$ (in a circle). This state is then processed by a "link" to produce an actual leakage $Y_{k,P}$ (in a circle). Both $X_{j,P}$ and $Y_{k,P}$ are fed into a distinguisher D (in a circle), which outputs a decision $j \neq s?$ (in a box). The text on the right explains these steps: "In the guess phase, the attacker guesses the part of the key j ." "The attacker computes the hypothetical sensitive state $V_{j,P}$. Then the attacker applies the leakage model to estimate, $X_{j,P}$, corresponding to the hypothesis for subkey, j ." "Then he applies a distinguisher D to compare the different models $X_{j,P}$ and the actual leakage $Y_{k,P}$." "For a successful attack, the best comparison results occur when $j = s$." At the bottom of the slide, there is a citation: "Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, Nicolas Veyrat-Charvillon: Mutual Information Analysis: a Comprehensive Study. J. Cryptology 24(2): 269-291 (2011)". There are also logos for "swayam" and "Maha Vidya" at the bottom.

So, therefore, here is the remaining part of the attack. So, in the guess phase so, therefore, the previous one was the you know like when you are profiling and you are getting the observable O and now you are trying to estimate the leakage s I mean the leakage L . So, this is a guess phase and in the guess phase the attacker guesses the part of the key say j and the attacker computes the hypothetical sensitive state $V_{j,P}$.

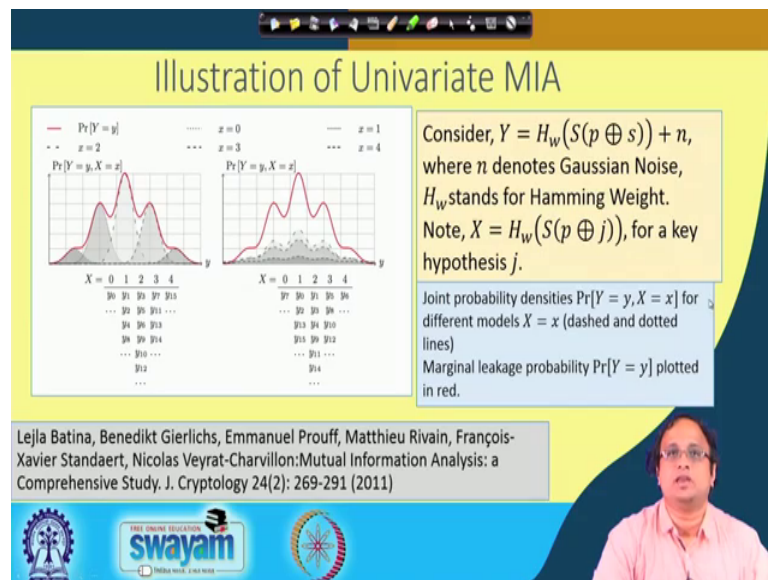
So, this is hypothetical sensitive state which your targeting so ok. So, therefore, I denote that as $V_{j,P}$ and the attacker applies the leakage model. So, this is the leakage model that I will soon elaborate more. So, basically you make a leakage model and you can so, the leakage model could be something like the hamming weight leakage model.

And then you know like the attacker applies the leakage model to estimate $X_{j,P}$ ok. So, this is the $X_{j,P}$. So, therefore, $Y_{k,P}$ is the observable something which you are observing and $X_{j,P}$ is what you are estimating by based on your guess and therefore, now for the successful attack you will basically apply

you know like a distinguisher d to compare these two different you know the different models X_j comma P and the actual leakage.

Note that this why I said a different mode X_j comma P because X_j comma P will depend upon the guess which you make. So, you will basically make j_1, j_2, j_3 and so on and therefore, you will guess a part of the key and you will try to see that whether X_j comma P matches with your actual leakage something similar what we have done in the case of correlation attacks where we are doing a correlation. But now we will be calculating this information or mutual information and we will basically try to find out what is the best comparison results and the idea is that the best comparison essentially will give you the correct subkey.

(Refer Slide Time: 17:41)



So, therefore, right here is an illustration of a univariate MIA. So, univariate mas stands for the fact that you are basically guessing at a at a distinct time instant ok. So, consider that you know like Y equal to so these are just an example. So, therefore, suppose Y equal to $H_w(S(p \oplus s)) + n$ this denoted as so, this is nothing, but s of P XORed s . So, therefore, P stands for the plaintext s for the part of the key and S is the S box which you are targeting and then I measure the hamming weight of the corresponding computation and then I add a Gaussian noise to that ok. So, therefore, right I mean note that the target right when you are basically guessing essentially.

You essentially have basically you are taking a key hypothesis. The key hypothesis is denoted as j and you again measure H_w of S of $P \oplus j$ and you measure the hamming weight of that and you denote that as X for the corresponding hypothesis. So, now, what we plot is basically the so, in this case this diagram shows the process by the red line, it shows the probability of Y equal to small y ok. So, the probability of Y equal to small y stands for the probability distribution of this Y and we also plot by this other graphs. We plot the joint probabilities or probability of Y equal to small y and X equal to small x .

So, note that we basically plot the joint probability densities for probability that is probability of Y equal to small y comma X equal to small x , but different models of X equal to small x . So, Y there are again different models. So, I said this will depend upon the corresponding guess that you do ok. So, the guess essentially in this case right. It is a suppose you are making a guess of a portion. So, the I guess right essentially can be say 0, 1, 2, 3 and 4 ok. So, note that you are making this guess based upon the corresponding you know the base on the corresponding hamming weights.

The idea is that you can reflect on this that if your X takes different values like X equal to 0, 1, 2, 3, 4, you will observe here that this you know like what we what we what we what we note here by the corresponding Y values are those cases right. So, where the hamming weight we basically plot the you know like we basically observe the corresponding values of Y for the corresponding value of X .

Note that in this case right this is for the correct key guess ok. So, for the correct key guess you note that if X equal to 0 then; that means, right I am corresponding to those cases where the hamming weight id 0 ok. And we know that Y equal to 0 or Y 0 essentially is the corresponding case where the hamming weight is 0 and there is only one such candidate.

Likewise when you go to X equal to 1, there are 4 cases like you have got 1, 2, 3 and 8 all of them have got hamming weight of 1. If you consider 2, there are more cases ok. Again if you consider 3, there are 4 cases. Again if you consider 4 hamming weight, there is only one case 1 1 1 1 and that is Y 15. So, now, when you are making so, therefore, right I mean you can observe that in this particular case the marginal. So, the if you observe right; if you observed that the marginal probability the marginal probability

is or the marginal leakage probability is denoted as probability of Y equal to small y and that is plotted by these redline ok.

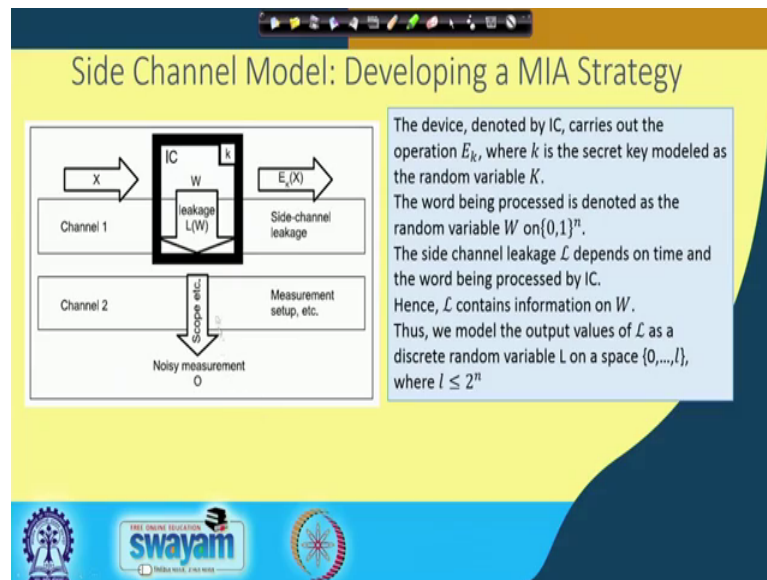
So, you will observe that this marginal probability matches very nicely with your joint probability distributions ok. So, there is a nice overlap between them where is when you are guessing it wrongly then its slightly differs. For example, the reason why it differs is because when you are guessing the key wrongly then; that means, there is a kind of mismatch and the miss match you can observe again by observing this table. So, for example, again X equal to 0, 1, 2, 3, 4; these are all the hamming weights of your, you know like the guessed state and these are the corresponding Y values. For example, note that suppose you know like so, suppose you know so note that for example, for 2 right.

If you consider 2 now like the where X equal to 2 in the previous case all these Y values had hamming weight of 2. But now this number will remain the same that is the number of Y values will fall into the pocket will again remain the same because you are just doing an XOR. So, therefore, you know like its basically a permutation of those values. So, therefore, you will still have the same number of corresponding Y values which we fall into the pocket of X , but the interesting thing is that out of here every Y value does not have a hamming weight of 2 ok.

For example you can see that like y_3 will have hamming weight of you know like of 2 you know like y_9 has a hamming weight of 2 and so, so y_3 has hamming weight of 2 and y_9 has hamming weight of 2 and there are no there are no more. So, therefore, you see that the you like that the joint probability we observe a probability of Y equal to 2 and X equal to 2 we will not now nicely match with probability of Y equal to 2 and this will fall much lesser. And therefore, right I mean what will happen is therefore, you we will find that this does not match very nicely with their. So, therefore, in this way you can actually develop a distinguisher this is essentially the basis why you can develop a distinguisher.

Because you know like that depending upon whether your guess is correct or whether your guess is wrong, you see that the marginal probability essentially has got a nice correlation with the joint probability. And if your guess is correct whereas, if your guess is wrong then that you know like that correlation is lost or that I would say that match is gone basically.

(Refer Slide Time: 23:33)



So, let us not use the term correlation here, but let us say that the match is gone. So, therefore, right this is the basic basis behind the attack behind the how we can develop our attack strategy. And therefore, what we do here is that we so, this is how we are basically you know like we can conceptualize the attack.

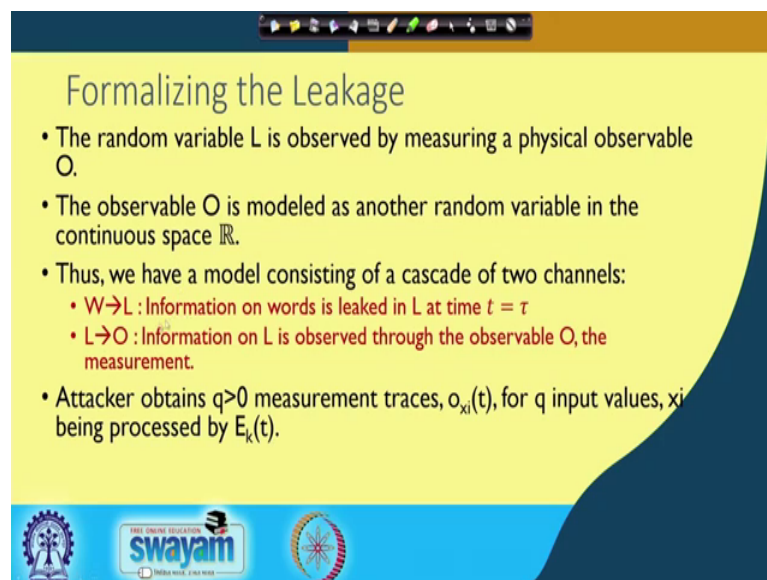
So, we have got a target chip or target design IC for example, and in that right. Basically it is processing some intermediate value say W and there are 2 channels here. In one channel the device correspondingly, you know you have processed the secret key is embedded for example, here as k and the leakage essentially is a side channel leakage ok. And there are 2 leakage directions you can see one. So, there are 2 channels basically in one channel we basically observe this $E_k X$ and the other one we basically do use a measurement. So, the measurement gives me an O value. So, therefore, right this E_k . So, therefore, so the idea is that this leakage ok.

So, basically it processes all the intermediate value say W and then gives the estimate which is essentially denotes as LW . So, this LW essentially stands for your leakage whereas, when you are applying or you are taking your scopes and your probes right, you get an actual power estimate that is your observable power that is denoted by O . So; that means, the device which is denoted by IC carries out the operation E_k where k is the secret key which is modelled as a random variables and the intermediate word which is being processed is suppose denoted by a random variable W on $0, 1, n$.

Now, the side channel leakage which is say denoted as L or math cal L which will depend upon time and also on the word which is being processed by the IC ok. So, hence L contains information about the intermediate value W and we model the output values or output values of L as a discrete random variable L on the space of 0 to L from like 0 1 2 3 4 L and so on where L is much lesser than or equal to 2 is lesser than or equal to 2 to the power of n . So, it can be you know like depending upon the corresponding state value like suppose there is an intermediate value say W which you are estimating.

Depending upon the value of this W you try to estimate L you try to estimate your leakage as L , L and you also have observed measurement which you are essentially measuring.

(Refer Slide Time: 25:51)



Formalizing the Leakage

- The random variable L is observed by measuring a physical observable O .
- The observable O is modeled as another random variable in the continuous space \mathbb{R} .
- Thus, we have a model consisting of a cascade of two channels:
 - $W \rightarrow L$: Information on words is leaked in L at time $t = \tau$
 - $L \rightarrow O$: Information on L is observed through the observable O , the measurement.
- Attacker obtains $q > 0$ measurement traces, $o_{x_i}(t)$, for q input values, x_i being processed by $E_k(t)$.

Logos at the bottom: swayam, THE OPEN EDUCATION, and other institutional logos.

So, therefore, right in the you can formalize this these two channels in this way. So, the random variable L is observed by measuring a physical observable O . So, therefore, the idea is that there is a random variable L which essentially the actual leakage, but you as an attacker are not able to observe that actual leakage, but you are getting a you know modified version of that. So, you are basically getting something like O . So, this is denoted by this channel where rather than observing L , you are observing O as an attacker.

But as an attacker also you can you if you can guess W or you can make a guess on W you can estimate the leakage L at some specific chosen time say τ . So, therefore, now

the attacker what it does is that basically does several measurements by varying the plaintext and it observe different values of the observable.

So, the observable are denoted as O by t which means you are it is nothing, but the measurement traces which you are observing ok. So, you are basically getting different power traces and you are giving inputs which are denoted as X i s and you are observing the corresponding power values at different instances of time. So, you are basically observing the power consumption across time instances.

(Refer Slide Time: 27:05)

Goal of an MI Adversary

- Let x_1, x_2, \dots, x_q be the known inputs being processed by the device.
- The corresponding measurements or observed side channels are the multi-set, $M = \{o_{x_1}, o_{x_2}, \dots, o_{x_q}\}$.
- The side channel experiment is defined as follows:

Experiment Exp_L^{SC}

$K \in_R \{0,1\}^m$

$x_1, \dots, x_q \in_R \{0,1\}^m, o_{x_1}, o_{x_2}, \dots, o_{x_q} \in O$

$k^* \leftarrow D(o_{x_1}, o_{x_2}, \dots, o_{x_q}; x_1, \dots, x_q)$

$Adv(o_{x_1}, o_{x_2}, \dots, o_{x_q}; x_1, \dots, x_q) = \text{Prob}[k^* = k]$

So, you basically get lot of values of O and now what you what you basically try to do is you try to develop a goal for the mutual information adversary. And the idea is that the mutual information adversary which basically what it does is it takes different input values say x_1, x_2 and so on till x_q ok. And then the corresponding measurements are observed side channels are you know like denoted in this multi set which is $O \times 1, x_2$ and so on till $O \times q$.

So, this is your observed power traces and then the side dchannel adversary experiment is defined as follows you guess a part of the key. So, therefore, this is denoted as K for example, and now you basically also using this guess you basically calculate the values of the intermediate target register which is a W which I denoted as remember W right here in this diagram

So, this W is estimated. So, the W is estimated based upon the input X and also the guess which you do on the key. So, therefore, now what I do is, I mean guess means guess on the part of the key. So, now, I guess this part of the key and depending on all my imports right I basically you know like observe the corresponding leakages and I basically try to find out their you know like try to find out apply a distinguisher on this guessed or estimated leakages with my observed output. And the idea is that it returns k^* and if my k^* matches with k , then I say that my experiment was successful and that gives me a measure of the advantage of my experiment.

(Refer Slide Time: 28:39)

Construction of an Information Based Distinguisher

- To each possible key $k' \in \{0,1\}^m$, we associate a partition:

$$H_{k'} = \{H_0^{k'}, \dots, H_l^{k'}\} \text{ on } \{0,1\}^m$$

$$H_i^{k'} = \{x \in \{0,1\}^m \mid L(f_{k'}(x)) = i\} \text{ for } i = 0, \dots, l$$
- The partition $H_{k'}$ induces a subdivision $G_{k'} = \{G_0^{k'}, \dots, G_l^{k'}\}$ of the measurement space O :

$$G_i^{k'} = \{o_x \in O \mid x \in H_i^{k'}\}$$

So, now what we try to do is we basically try to construct an information based distinguisher. So, to each possible key say k^* which I you know like choose from $0, 1$ to n I basically create this partitions of my input space ok. So, basically what it means is I now choose my input space and I divide my input space into certain partitions ok.. So, this partitions are remember that I have got input X I guess a key. So, this k or key part say suppose this is k^* and apply say the box or some kind of function which is denoted as f .

So, I get a value of f_{k^*} applied on x or f_{k^*} applied on x . Now I applied power model. So, this power model this could be you now like the hamming weight power model for example, gives me an estimate of its leakage this leakage is suppose some value i . So, now, what I do is I basically in one partition suppose there is a partition for i , I basically

put in all those x values which is essentially falls into that partition. So, these are if there are two x values which is these to the same I value then they belong to the same partition. So, this way I create this partition and that is essentially denoted as H_i likewise

You know like I have got leakage partitions for 0 for leakage one two 3 and so on. So, all the possible leakage values all these values together gives me this partitioning denoted as H_k . Now H_k also induces a subdivision which is denoted as G_k which is essentially nothing, but. So, if you consider you know like one of these leakages say for example, you know like let me write as you know like H_i . So, what is H_i ? H_i means stand for all those you know like input values which leads to the leakage i .

So, now what I do is in this? So, therefore, imagine that there is a bucket corresponding for L_i or leakage equal to i ; i basically observe all the observables which essentially goes into here ok. So, remember that I can have some observation O_1 I can have so, observation O_2 likewise I can have several observations which can go over here and then they are not same. The observed power values are not same, but the estimated leakage from where you are basically you know like getting the I mean basically write I mean the leakage right is getting modified to 0. So, although the leakage is remaining same, but this can vary because of noise and several other parameters.

So, all those things right which essentially goes into this bucket essentially are denoted O the various O values and this gives me this partitioning which is denoted as G_k ok. So, G_k therefore, has got G_0 and likewise till G_L indicating right that in one of these the if I write G_i then G_i would mean that you know like for all those input x values which are essentially has gone to this leakage or which has lead to this leakage i ok, what are the corresponding values of the observables.

So, all those observables which are essentially you know are corresponding to those x values which are lead to i are essentially observed in this partition they are kept in this partition. Therefore, right I am kind of finding out how many observables are there or what are the observables we are there ok.

(Refer Slide Time: 32:07)

The MI based formalism

- Adversary guesses a key value, k and computes $f_k(x)$, which is an intermediate result like $Sbox(x \oplus k)$
- Usually, W is some bits of $f_k(x)$, say 3 MSB.
 - thus for each guesses key k , there are 8 bins: $L_k = \{L_0, \dots, L_7\}_k$
 - Divide the observables for each inputs among these bins.

• Compute the distributions:

$$P_{0|L_i} = \frac{|\{o_{x_j} = 0, st. x_j \rightarrow L_i\}|}{|L_i|}$$
$$P_0 = \frac{|\{o_{x_j} = 0\}|}{q}$$

Handwritten red annotations on the slide include a diagram of a function f with inputs x and k , and a diagram showing a distribution over bins L_0, L_1, \dots, L_7 .

So, now, so, now with this definition right we can essentially go to calculate or define what is essentially the we can go to and define the actual attack. So, therefore, now what the attacker does is that therefore, the attacker guesses the key value k and computes $f_k(x)$ which is an intermediate result like S box of XOR k . And now I basically you know like this W is say some bits of a $f_k(x)$ say it could be the it could be the hamming weight also, but suppose it is some bits, bits of $f_k(x)$ say it is the 3 MSB bits. So, therefore, you I take the 3 MSB bits like what I am trying to say is that ah

If right when you are doing this right x XORed with k and you are applying the S box say denoted as f and you are targeting one of these outputs. Suppose right my W is given by the first 3 bits over here and these bits or the value of this 3 bits essentially will give me the corresponding leakage class. So, how many values can this 3 bits it can take? It can take 8 values right. So, therefore, all of these 8 values are denoted by this leakage classes from L_0 s you can imagine that there are buckets like L_0 L_1 . So, on till L_7 , there are 8 buckets over there.

So, now what I do is basically I know that suppose I you know like fix L_0 , I want to find out all the O values which are gone over here. So, therefore, it could be that O_1 is you know like there O_1 is one value O_2 is the other observables O_3 is the observables and so on. So, using that right I basically estimate this P_0 given L_i . So, P_0 over given L_i is nothing, but the number of values which are there in L_i ; that means, this stands for the

cardinality of this L_i set. And the numerator is nothing, but all those x_j values which was led to L_i , I have found out the corresponding O values and I have.

I want I am basically observing that whether this $O \times j$ is equal to O or not and that gives me an measurement of this probability distribution of probability O_o given L_i . Likewise I can calculate probability of O as nothing, but total q and you know like the total set of how many $O \times j$ s is equal to O ok. So, this is the cardinality of this set. So, once you have developed these probability notions ok. So, what we can do is we can now measure the corresponding entropy.

(Refer Slide Time: 34:21)

Entropy and Mutual Information

- These distributions lead to the following entropies:
 $H(O|L_i)$ and $H(O)$
- Remember $H(O|L) = -\sum_{o_j \in O, L_i \in L} Pr[o_j, L = L_i] \log(Pr[o_j | L = L_i])$
- Since, this leakage is estimated based on a guessed key, we can also call it L_k
- Thus our required Mutual Information,
 $I(L_k; O) = H(O) - H(O|L_k)$
- Note that these computations are under assumption of a portion of the key.
- Thus, $k^* = \operatorname{argmax}_{k \in K} I(L_k; O)$
- Since, $H(O)$ is fixed for all key hypothesis, so equivalently:
 $k^* = \operatorname{argmin}_{k \in K} H(O|L_k)$

So, the corresponding entropy is there for now given by this formalism. So, therefore, suppose I want to calculate. So, we are all said to calculate $H(O|L_i)$ given L_i and also $H(O)$. And therefore, we know that $H(O)$ given L is nothing, but this probability estimation. So, there it is probability of O equal to O_j comma L equal to L_i multiplied with log of probability of O equal to O_j given L equal to L_i .

So, therefore, right this leakage is estimated based on a guessed key and we call it as L_k and then what we do is that we in our mutual information estimation. We can either maximise this $I(L_k; O)$. So, we basically written that k for which this is maximized or as we can observe the $H(O)$ is constant, we can just minimise this parameter and that is this denoted as this. So, we can basically minimise the conditional entropy $H(O|L_k)$.

(Refer Slide Time: 35:09)

A Toy Example

Diagram: $x \oplus k \rightarrow \text{S-Box} \rightarrow \text{Output}$

000	001	010	011	100	101	110	111
111	110	101	100	011	010	001	000

3	2	2	1	2	1	1	0
---	---	---	---	---	---	---	---

Power Simulation using Hamming Weights

swayam

So, now let us try to apply this on a toy example. So, this is a toy example of an S box which is nothing, but you know like this is a mapping which is it is just a symbolic box. So, what we do is that we basically model the power based on the hamming weights. So, you can see here the hamming weight 3 means it is written the observable is 3. This is the hamming weight is 2 so, the observable is two the hamming weight is 2 the observable is two and so on.

(Refer Slide Time: 35:33)

The Leakage function should not be Bijective

- Suppose leakage is denoted by all the 3 bits: thus the leakage classes are $\{0, 1, \dots, 7\}$
- Suppose, consider an attacker who guesses the key to be k .
- Then he determines the leakage L_k by calculation $S(x \oplus k)$, for different x values.
- Then it is evident that because of this leakage function, $\forall i, \exists j, \Pr[O = j | L = i] = 1$, and for all other j 's, the probability is 0.
- This implies $H(O|L) = 0$, and does not differ with k .

swayam

And now what we do is we basically use it to do the attack. There is one further important point which should be mentioned here is that the leakage function which I mentioned should not be bijective because if the leakage function is bijective. Suppose in this particular example, I choose say all the 3 bits as my leakage value; that means, if I choose all the 3 bits of my output right of my output S box, then there are eight possible classes and you can if you just think a while right you will understand that if it is. So, then it is evident that because of this leakage function for all i , there will exist one j for which these condition probability will be one because the mapping will be bijective.

So, there will be one case we always fall into that case whereas, for all other j s the probability will be 0 and if this happens like this $H O$ given L will be 0 and it will not differ with k . So, therefore, it will not work.

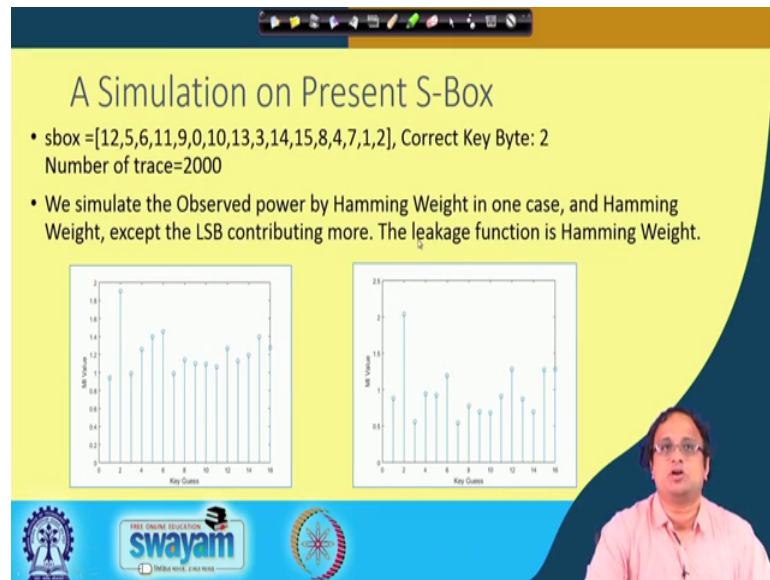
(Refer Slide Time: 36:21)



So, an easy way out therefore, is that make the leakage non bijective say for example, let us make it hamming weight for example. You can also make it say the last two bits or the last 3 bits. So, in this case if I make it you know like the last for example, the hamming weight then this is a corresponding plot of the m_i value with the key gusses and you can see that in this case the correct key base is give the is the first key base and which is unit correct ok. This is my correct key guess ok.

So, this is the starting key this is the next key. So, you can get you can see that you can get peaks. Likewise also note that you can get ghost peaks like there are some keys right which you for which you can get wrong possible peaks as well.

(Refer Slide Time: 36:57)



So, here is a further example on a present S box. So, present s box is being denoted here. So, in this case the correct key byte is at specific location the number of trace is 2000. Again you see the now right you can get a corresponding peak. So, in this case the leakage is modelled again by the hamming weight and we indeed get a corresponding peak at the corresponding key guess value.

So, in one case I have kind of simulated the power with the with only hamming weight, in one case I have a kind of plotted with the hamming weight except that one bit is contributing more to the leakage ok. So, we are basically added and extra intensive for that bit ok. This is just to see our see our simulation works.

(Refer Slide Time: 37:33)

A Real Life Example

- Target: AES-128 implemented on Sakura-G
- Power Trace are acquired by varying plaintexts at random:
 - The experiment focuses on the first key byte of last round key.
 - Correct last round key:
0x4A3315ED79D3FE1FD8B1E90D5133BE2D
 - q=8000 power traces are gathered, $O_{x_i}(t)$, $i=1$ to q.

swayam

So, here is a real life example where we basically target an AES-128 implemented on a Sakura-G platform and the power trace which is acquired by varying plaintext at random the experiment focuses again. So, this is my correct key this is a 128 bit key and I again focus only on the first key byte. So, we take 8000 power traces ok.

(Refer Slide Time: 37:55)

MI on AES-128 Implementation

- Compute $f_k(x) = Sbox^{-1}(x \oplus k)$, where x is the ciphertext.
- The leakage function, L is say the lower 4 bits of $f_k(x)$: so we have 16 such leakage classes.

Distribute the x values into the 16 bins along with their observed power leakages, O_x

Now calculate $H(O|L_k)$ for all these guessed keys k . Note there are 256 guessed possible.

L_0 L_{15}

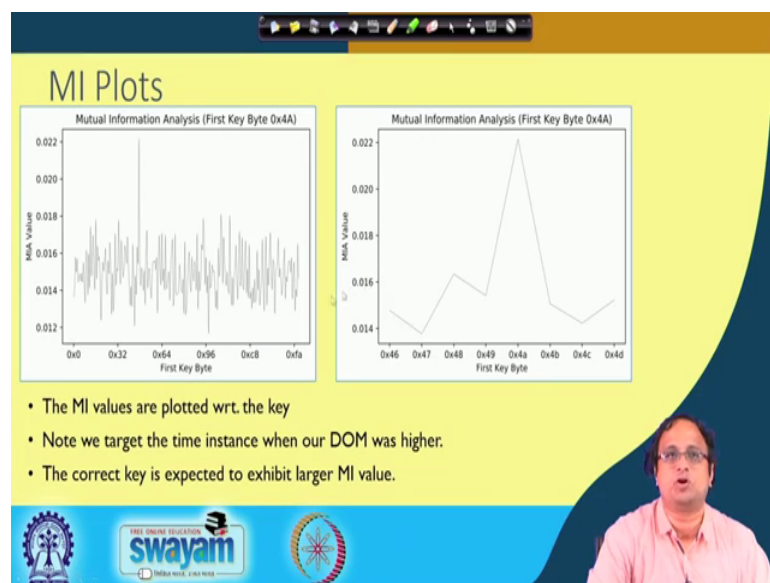
swayam

We observe the corresponding observables O and then do a similar mutual information of analysis attack. So, therefore, what we do is we calculate we are doing from the cipher text now. So, therefore, we do S box inverse of x XOR k where x is a cipher text. Now

the leakage function is say the lower 4 bits of f_k or it could be the upper 4 bits of f_k as well. So, therefore, the moment we choose 4 bits we have got 16 classes or 16 bins.

So, you have got 16 bins. So, for every bin we can calculate $H(O \text{ given } L_k)$. So, note that $H(L_k \text{ given } L_k)$ means that I am guessing the key and depending upon that I am calculating this is $H(O \text{ given } L_k)$ ok. Note that there are 256 you know like guessed possible values here and therefore, we distribute the x values in to the 16 bins along with their observed power leakages O_x .


(Refer Slide Time: 38:43)



We estimate this $H(O \text{ given } L)$ and therefore here is a plot of the mutual information. Again you can observe that for the correct key guess right here you will get a peak here and here this is zoomed version.

So, you can see that the key is quite visible. So, again note that if you go back right and see the correct key the correct key was 4 a and indeed right and the 4 a position, you get a peak here ok. So, this shows very nicely that the mutual information attack works and it is working correctly.

(Refer Slide Time: 39:07)



The slide is titled "Conclusions" and "References". The text on the slide is as follows:

Conclusions

- Mutual Information is fundamental to understanding leakage
- We discussed on use of MI for Side Channel Analysis
- Works well when the leakage is not linear, unlike DoM or CPA.
- We discussed case studies to see MIA being effective side channel tool.

References:

1. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, Bart Preneel: Mutual Information Analysis. CHES 2008: 426-442
2. Nicolas Veyrat-Charvillon, François-Xavier Standaert: Mutual Information Analysis: How, When and Why?. CHES 2009: 429-443

The slide also features a video inset of a man in a pink shirt speaking, and logos for Swamyam and other educational institutions at the bottom.

So, therefore, the mutual information is quite fundamental to understanding leakage we discussed on the use of MI for side channel analysis. It works well when the leakage is not linear for example, you know like DoN or CPA which works probably like the better when the leakage is linear.

But it can even work in the MI can your MI will work way better when the leakage is not linear and we discussed several case studies to see how MIA works and is an effective side channel analysis tool. So, here a couple of interesting papers which you can also read one version which is published in CHES in 2008 followed by another paper in CHES 2009, which tells us exactly or gives a more descriptions on where to use mutual information and actually where were not to use mutual information analysis ok. So, with this I would like to thanks to you and we shall again join in the next class.

Thank you.