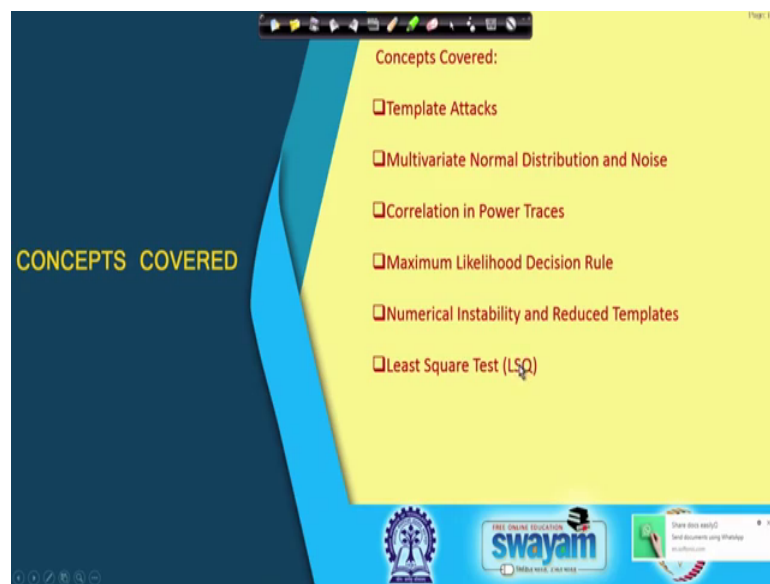


**Hardware Security**  
**Prof. Debdeep Mukhopadhyay**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 38**  
**Power Analysis – XIV**

So, welcome this class on Hardware Security. So, today we shall be continuing our discussions on power attacks.

(Refer Slide Time: 00:23)

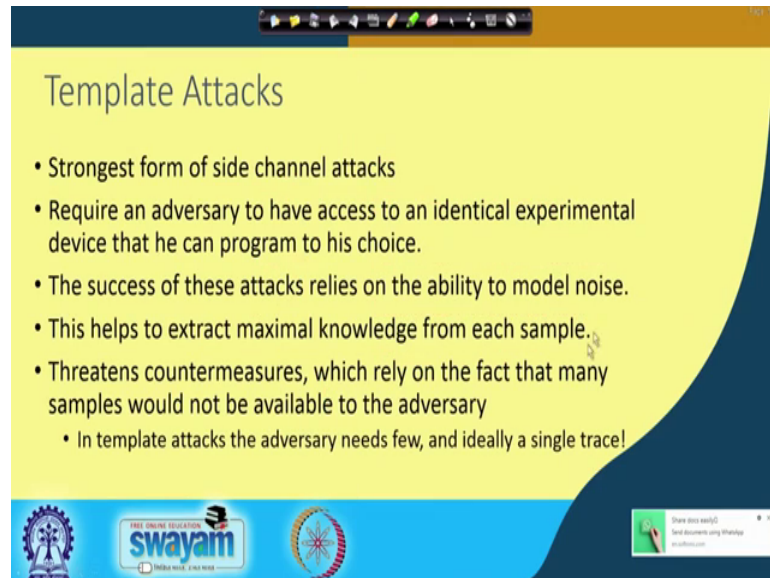


In particular we shall be talking about a new type of power analysis which we essentially have not been discussed I have not discussed before is a what is called as template attacks. So, it is basically a form of simple power attacks and essentially I would say like is one of the most powerful forms of side channel attacks ok.

So, we shall be discussing about a multivariate normal distribution you know like formulation or formalization of noise which is very central to understanding say template attacks. We shall be discussing about correlations in power traces and finally, on maximum likelihood decision rule which we used to determine the key in case of template attacks. I shall be talking about some numerical instability problems which can come up because of the formalization of template attacks and discuss about the technique which is called reduce templates to remove it. And essentially conclude with at you know

like a modification of the basic template analysis is called as the Least Square Test or LSQ.

(Refer Slide Time: 01:23)



Template Attacks

- Strongest form of side channel attacks
- Require an adversary to have access to an identical experimental device that he can program to his choice.
- The success of these attacks relies on the ability to model noise.
- This helps to extract maximal knowledge from each sample.
- Threatens countermeasures, which rely on the fact that many samples would not be available to the adversary
  - In template attacks the adversary needs few, and ideally a single trace!

swayam  
INDIA RISE, INDIA RISE

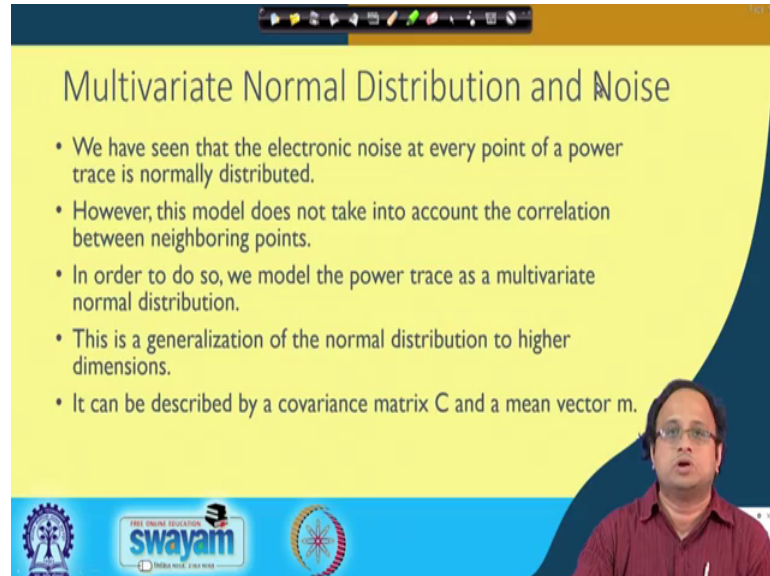
So, to start with as I said the template attacks is like one of the strongest forms of side and attacks where we required an adversary to have access to an identical experimental device. So, that you can program to his choice; basically he can give inputs he can varies the key ok. So, he can basically pretty much play around with the device. Now the success of these is of this attacks relies on the ability to model noise like as you have seen like in the previous side chain attack. So, right then noise was a deterrent because noise essentially was reduce in the SNR or the Signal to Noise Ratio which was adversely affect the success rate of my attack.

But here that we will take a different approach, we will basically try to kind of model noise in a very accurate manner and hence we will try to use it to extract maximum knowledge about from every sample. That means, from each sample we will be trying to estimate very accurately the noise and from there we will be trying to estimate or you know extract maximum knowledge and that is why the final attack can be done with very few traces probably even ideally with only one ok.

And it can threatened countermeasures because many of the counter measures are developed with this assumption that the attacker will not have access to significant

number of power traces. But as in template attacks we require very few ideally probably only a single trace. Therefore, many of these counter measures can be challenged ok.

(Refer Slide Time: 02:47)



The slide is titled "Multivariate Normal Distribution and Noise". It contains a list of five bullet points. At the bottom right of the slide, there is a small video inset showing a man with glasses speaking. The slide also features logos for "swayam" and "INDIA RISE, EDUCATION RISE" at the bottom.

- We have seen that the electronic noise at every point of a power trace is normally distributed.
- However, this model does not take into account the correlation between neighboring points.
- In order to do so, we model the power trace as a multivariate normal distribution.
- This is a generalization of the normal distribution to higher dimensions.
- It can be described by a covariance matrix  $C$  and a mean vector  $m$ .

So, therefore, write this brings us to the modelling of noise like till now we have been looking at noise right. Essentially we have been focusing about every point in the trace in the power trace and we have seen right that the electronic noise essentially at that every point was kind of having a Gaussian distribution that we discussed in the previous class.

So, basically we were you know like looking at every point in a discrete manner ok. But on the other hand if you look or think about it right, electronic noise at every point of the power trace though it is normally you know like it is kind of normally distributed, but they are not completely independent of each other. So, what I mean like if you look at adjacent points or time instances in the power trace, then maybe correlations for noise. And that essentially you can imagine like suppose your processing you know like or doing an AES operation or some kind of transformation the inside your chip.

Then successive time instances right will essentially have essentially you are kind of you can think of like you are basically doing the same operation and you are distributing it over time. So, therefore, right for example, it may happen that you are doing an operation over  $r$  rounds ok; an inside the round that I can some operations which were doing across clock cycles. So, therefore, right you can imagine there suppose you are doing a

multiplication which is kind of spread across clock cycles, then you can expect that there would be correlations in the combinations that you are doing ok.

So, this is the single part of the power trace, but the interesting thing is an even the electronic noise at every point of the power trace right, essentially there is an amount of correlation between them. Whereas, if you take to distance point say points, then the correlation between the power traces are expected to be less; I mean the correlation between the noise components are expected to be less. So, therefore, right this brings us to this multivariate normal distribution of modelling of the power trace where you know and it is basically a generalization of the normal distribution to higher dimensions.

So, it can be described by a covariance matrix and a mean vector  $m$ . So, in the previous case right we have basically looking at every time instances and we have seen that new and sigma we are used to characterize it. Here since we are looking at multiple time instances. So, rather than processing on one single scalar mean value, we will basically processing on a vector mean or a mean vector and also a covariance matrix.

(Refer Slide Time: 05:09)

The slide is titled "Definition of Multivariate Gaussian Model". It contains the following text:

- The probability density function (pdf.) of the multivariate normal distribution:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\mathbf{C})}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x} - \mathbf{m})' \cdot \mathbf{C}^{-1} \cdot (\mathbf{x} - \mathbf{m})\right)$$

Handwritten notes in red ink next to the equation include "Mean vector" and "Covariance matrix". Below the equation, it says  $\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]$ .

- The covariance matrix  $\mathbf{C}$  contains the covariances  $c_{ij} = \text{Cov}(X_i, X_j)$  of the points at time index  $i$  and  $j$ .
- The mean vector  $\mathbf{m}$  lists the mean values.  $m_i = E(X_i)$  for all points in the curve.
- When filling  $\mathbf{C}$  and  $\mathbf{m}$  into the above equation, the pdf for the vector  $\mathbf{x}$  is returned.

The slide also features logos for "swayam" and "Free Online Education" at the bottom, and a small video feed of a presenter in the bottom right corner.

So, basically right I mean this brings us to this formalization or probability density function of the multivariate normal distribution. So, you can see that  $f(\mathbf{x})$  is equal to  $1$  by square root of  $2\pi$  whole to the power of  $n$  determinant of  $\mathbf{C}$ . So,  $\mathbf{C}$  is the covariance matrix here and  $\mathbf{m}$  is essentially the mean vector ok. So, you can see that we have written  $e$  to the power of minus half  $\mathbf{x}$  minus  $\mathbf{m}$  note that  $\mathbf{x}$  is essentially also a vector ok. So, you

are basically doing a vector subtraction and then you are multiplying it you know like you are basically doing a transpose and you have multiplying it with the inverse of the covariance matrix and you have multiplying this with the vector  $x$  minus  $m$  ok.

So, therefore, right basically the idea is that you have got certain points of interests. So, suppose maybe you know right suppose as an examples think that suppose we are observing in the power trace 10 time instances. So, therefore, previously we will looking all of them independently, but now we are trying to process them as a vector ok.

So, therefore, you can imagine that if I have got in my power trace for example, you know like this like suppose this is my this is this is a power trace that I have right. And I observe different time instances in a window in a window of time. So, suppose this is my total window of operation and I am observing the corresponding power values which I denote as say  $x_1, x_2, x_3$  and so on till  $x$  maybe  $x_t$  because suppose or maybe  $x_n$  like if  $n$  is my total window size ok.

So, therefore, right now this stands for my vector  $X$ , this stands for my vector  $X$  and I basically find out the mean vector also in this fashion because every time instance right here is now also you know like a is essentially a statistical distribution. So, we basically take out take the take the mean of every point right. And then that gives me  $y$  vector I mean that gives me my vector or mean vector and I also similarly calculate the covariance.

So, the covariance basically as written over here is basically the covariance of the points at index  $i$  and  $j$ . So, therefore, right if I observe a here there is a time instance suppose the time instances  $i$  and there is a time instance say  $j$ , then covariance of  $x_i$  and so, this brings us to a definition of the multivariate Gaussian model. So, this is the generalization of the normal distribution that we have already seen. So, here you can imagine that we rather you know like observing a single point of the power trace are essentially observing a window. So, we are basically observing a window of time instances.

(Refer Slide Time: 07:53)

Definition of Multivariate Gaussian Model

- The probability density function (pdf.) of the multivariate normal distribution:  
$$f(x) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(C)}} \cdot \exp\left(-\frac{1}{2} \cdot (x - \underline{mm})' \cdot C^{-1} \cdot (x - \underline{mm})\right)$$
  
Handwritten notes:  $x = [x_1, x_2, \dots, x_n]$ ,  $\underline{mm} = [m_1, m_2, \dots, m_n]$ ,  $C = [C_{ij}]$
- The covariance matrix  $C$  contains the covariances  $c_{ij} = \text{Cov}(X_i, X_j)$  of the points at time index  $i$  and  $j$ .
- The mean vector  $\underline{mm}$  lists the mean values.  $m_i = E(X_i)$  for all points in the curve.
- When filling  $C$  and  $\underline{mm}$  into the above equation, the pdf for the vector  $x$  is returned.

So, therefore; that means, like suppose this is my power consumption. So, these are previously we are observing like at one time instance which is my which was my point of interest. But now I am observing a window of time instances. So, therefore, this is my corresponding window ok. So, therefore, every time instance you can imagine that this is like  $x_1, x_2$ . So, these are you know like the various samples that you are getting at different time instances. So, let us write  $x_1$  to  $x_n$  to  $x_n$  for example,. So, this basically makes my vector  $X$ ; this is my vector  $X$ .

So, therefore, what I now do is basically I calculate the mean at every point ok. So, every point basically I when I am varying it I am basically getting different values at that point and I take a mean at that point and therefore, right I now get a mean vector. So, previous day I was getting only one  $\mu$  value. So, now, I am getting a  $\mu$  I am getting a you know like a mean vector.

So, this mean vector is essentially denoted here as this  $\underline{mm}$  value and now and also right along with it; we basically find out we also want to kind of find out the covariance at different between two time instances. For example, suppose  $x_i$  is one time instance and  $x_j$  is another time instance ok. And I want to calculate the covariance between these two time instances. So, that essentially is basically denoted by  $C_{ij}$  and therefore, you can imagine that I can basically make a matrix right which I called as the correlation matrix

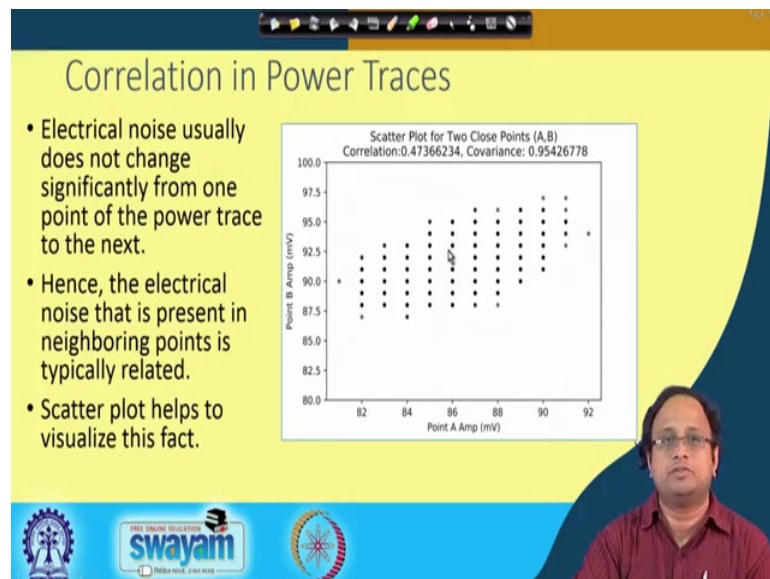
where all these values  $C_{ij}$  basically stands for the correlation or I mean stands for the covariance between two some instances  $i$  and  $j$ .

So, you can imagine that you know like this would be a symmetric matrix because  $C_{ij}$  and  $C_{ji}$  would be same and therefore, this matrix would be in essentially a symmetric matrix. So, therefore, in this fashion right I can get the mean vector  $m$  which basically least the mean values  $m_i$  which is equal to  $E X_i$  for all points in the curve and you will also have  $c$  which is usually nothing, but filling up the covariance matrix.

So, when filling up  $C$  and  $m$  like if I get the value of  $C$  and  $m$  and then plug it into the equation, then the pdf or the probability density function of the vector  $x$  is returned. So, therefore, where I am observing a power trace; so, this power traces say call  $X$ . I basically get the corresponding pdf or the probability density you know like function for this vector  $X$  is basically returned for any given vector  $x$ . You can calculate that by just plugging in this and calculating this value.

So, that is essentially you know that the background behind the multivariate Gaussian modelling.

(Refer Slide Time: 10:19)



And now what we will try to do is we will basically try to observe this for a really power trace. So, this is the case study that we did when you basically implement say an AES



algorithm or AES architecture on circular board as we have seen in our side channel setup.

And now we see that electrical noise right really does not significantly change from one point of the power trace to a subsequently close point. So, so what we do here to we basically can easily study this by something which is called as a scatter plot. So, in the scatter plot in the x axis, we point you know like the say the you know the amplitude. For example, which essentially basically I am taking two close points A and B and we are trying to plot the amount of power which is consumed at two close points and you can see from this scatter point right; that means, there is a kind of correlation which you can observed that you like which essentially can be modeled by essentially a straight line.

So, ideally right there will be a very you know like you can see that there is a correlation and if you know like get the best fit. For example, you will find that there is a line that you can draw over here and you can understand that is an amount of correlation. So, this correlation right is essentially as a of the like the covariance basically is also another way of measuring this dependence.

So, therefore, right we observe that electrical noise really indeed does not change across close by points and the electrical noise that is present in neighboring points is therefore, typically quite related there is an amount of correlation between them.

(Refer Slide Time: 11:47)

The slide is titled "Correlation in Power Traces" and features a yellow background. On the left, there is a list of bullet points. On the right, there is a scatter plot titled "Scatter Plot for Two Distant Points (A,C)" with the following statistics: Correlation: -0.01901006, Covariance: -0.03637775. The x-axis is labeled "Point A Amp (mV)" and ranges from 82 to 92. The y-axis is labeled "Point C Amp (mV)" and ranges from 80.0 to 100.0. The scatter plot shows a grid of points with a few outliers. At the bottom of the slide, there are logos for "swayam" and "INDIA'S OPEN EDUCATION" along with a small video feed of a man in a red shirt.

**Correlation in Power Traces**

- The linear relationship between two points of a trace is based on covariance or correlation.
- $Cov(X,Y)=E(XY)-E(X)E(Y)$
- Estimation of Covariance:
- Type equation here.

Scatter Plot for Two Distant Points (A,C)  
Correlation: -0.01901006, Covariance: -0.03637775

Point C Amp (mV)

Point A Amp (mV)



But if you observe the same thing over you know like far of points for example,. You find that that correlation has been destroyed. And you can indeed observe this that you know like what we have done is basically if you see in this plot we basically measure the correlation. The correlation turns out to be something like 0.5 which is quite high.

And if you compare it with this correlation and it something like minus 0.01 which is quiet less and therefore, you can you need understand that when you are taking too far of points and you are trying to correlate the amount of power consumed then there is less dependence across them. So, this essentially gives us a good amount of understanding about how the noise behaves across time instances and basically gives us a an ability to noise model the noise ok.

So, therefore, the so, so the so, this you know like the covariance or correlation are therefore, statistical tools to measure the linear relationship between two points of the power trace and as we know that covariance of X comma is Y is nothing, but e of X or the expectation of X Y minus EX into EY and that essentially can help me to estimate the covariance.

(Refer Slide Time: 12:49)

**Correlation in Power Traces**

- The linear relationship between two points of a trace is based on covariance or correlation.
- Correlation Coefficient is also a measure for the linear relationship between two adjacent points in the trace.

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Scatter Plot for Two Distant Points (A,C)  
Correlation:-0.01901006, Covariance:-0.03637775

The closer the points more is the correlation!

swayam

And likewise right you can also calculate the correlation coefficient and correlation coefficients essentially are nothing, but you know like giving us the ability or to measure the linearity relationship between two adjacent points in the trace like covariance ok.

So, typically right as we know that the you know like the correlation. For example, can be found out by this formula which is nothing, but the covariance of X comma Y divided by the square root of variance X and variance Y. So, the covariance of x comma y is estimated by this by this numerator which is like sigma i equal to 1 to n x i minus x bar into y i minus y bar divided by the square root of the variance of the product of the variances.

And this you can observed that you know like that if I if I basically taking this formula and I calculate the correlation then this correlation is quite small when I am considering too far of points in the power trace. Whereas, if i take two success points or two close by points, then the correlation is quite high ok. So, you can see it something like 0.5 which is pretty high.

So, on the other hand right, you can observe the covariance also see cannot give you the similar estimation. So, you can see the covariance is something like minus 0.36. So, you kinds of tells that it is kind of negatively correlated, but if you see the absolute value also the covariance is much less compare to here there is the covariance is as highest 0.95 ok; so which shows that there is a more dependence when you are considering close by points.

(Refer Slide Time: 14:17)

The slide is titled "The AES Architecture and Template building Phase". On the left, there are two diagrams of the AES round function. The top diagram shows a sequence of blocks: "Add Round Key" (receiving "Keymem"), "Buffer", "Byte Sub", "Shift Row", "Mix Column", "Buffer", and "Add Round Key" (receiving "Keymem"). The bottom diagram shows a similar sequence but with an additional "Add Round Key" block receiving "Keymen" before the final "Buffer" block. In the center, there is a text box: "We keep the plaintext same and vary say the first byte of the key matrix,  $K_0$ . Depending on the Hamming Weight of  $K_0$ , we create 9 templates." To the right of this text, there are handwritten red notes and diagrams. One diagram shows a bit pattern  $\{0, 1, 1, 1, 1, 1, 1, 1, 1\}$  with a circled 1 and a plus sign, and another diagram shows a bit pattern  $\{0, 1, 1, 1, 1, 1, 1, 1, 1\}$  with a circled 1 and a minus sign. Below these, it says "Note that there are some templates where there are more values possible: Number of instances with Hamming Weight w is:  $\binom{8}{w}$ ". At the bottom right, there is a small video inset of a man speaking. The slide also features logos for "swayam" and "THE INDIAN EDUCATION" at the bottom.

So, therefore, now what we will try to do is win this background we will be trying to look into the AES architecture and try to see how we can build the template. So, for this

right we will basically again look into the AES architecture. So, this is an you know nothing, but a recapitulation of how the AES works it. In fact, right we will be considering only at the first state where I am basically taking the plaintexts I am doing an addition round key with a with the key.

So, where basically what I mean is I am basically concentrating on the first part where I have got the plaintext, but I am considering only on the zeroth byte. And I am XORing it with again you know like once like 16 bytes, but I am again focused only on the first part which is my  $K_0$  that is the first part of the key and I am trying to basically build templates.

So, what I do is we basically keep the plaintext same and vary say the first byte of the key matrix  $K_0$  ok. So, we basically keep this  $P_0$  constant and we basically vary this part of the key. So, basically right I mean this  $K_0$  can take 256 values, but rather than dividing it into or building templates of you know like 256 classes, we basically make templates depending upon their hamming weights ok. So, therefore, I will have 0 to 9 0 to 8 possible; that means, 9 possible hamming classes and depending upon the hamming weight of  $K_0$  are basically form individual templates.

So, you can easily understand that there will be 9 templates, but there are some templates whether a more values possible compared to the other ones. For example, right if I just tell you that the template that the hamming weight is  $w$ , then the number of instances you will basically fall into that template class right is  $\binom{8}{w}$  choose  $w$  which is not same if I vary  $w$  ok. For example, if the weight is 0, then there is only one possibility which is all 0 likewise if the hamming weight is 8, then again that is one possibility. But if there are you know like if the hamming weight is one, then there are eight possible choices ok.

So, therefore, this kind of varies and therefore, the number of; so, this estimation is I mean right now you know like since we have got different amount of samples right, it will not be exactly accurate. But we will see that we will try to take this as an example to understand the basic concept of how to build templates and how to apply template analysis ok.

So, now, what we will do is basically we will keep on you know like the you will keep plaintext same say the I am basically try to build templates in this fashion. And so, what we will do is basically we will take this and try to create templates in this fashion.

(Refer Slide Time: 16:45)

**Creating Templates**

- We target say the key XORring step, by keeping the plaintext byte,  $P_0$  constant and changing the other 15 bytes.
- For each template again we keep  $K_0$  from a Hamming class say,  $w$ , and vary the other 15 key bytes.
- We target the first part of the power traces when the Add Round Key is participating in the underlying computation.
- We choose 10 time instances and calculate the mean vector  $\mathbb{M} = \mathbb{m}$ , and Covariance matrix  $\mathbb{C} = \mathbb{c}$ .
- Thus for every  $(P_0, K_0)$ :  $T_{P_0, K_0} = (\mathbb{m}, \mathbb{c})_{P_0, K_0}$  is a template for  $K_0$  belonging to the Hamming Class  $w$ .

The slide also features a video feed of a presenter in the bottom right corner and logos for 'swayam' and other educational institutions at the bottom.

So, we target say the key XORring step by keeping the plaintext by  $P_0$  constant and changing the other 15 byte. So, so, basically we need large number of samples to create the templates. So, we basically vary the remaining 15 bytes of the plaintext, but we keep  $P_0$  constant. So, we keep  $P_0$  fixed ok.

For each template again we keep  $K_0$  from a hamming weight class say suppose I choose  $K_0$  to be from the 0 class so; that means,  $K_0$  is held to 0 whereas, the remaining 15 bytes of the  $K_0$  I can vary at my I can vary. So, now, we therefore, we target the first byte of the. So, basically we have the power traces and we know that we probably have an understanding about you know like when the first; we basically target the first part of the power trace because we know that they are the add round keys participating in the underlying computation. You can of course, make this process more systematic, but at this point I am assuming that we have an kind of knowledge about this fact that you know that when in the power trace this component is coming into play.

So, now, we just choose arbitrarily ten time instances again these are important choice and calculate the mean vector  $\mathbb{m}$  which is equal to  $\mathbb{m}$  and the covariance matrix which like  $\mathbb{C}$  which takes the value of  $\mathbb{c}$  or small  $\mathbb{c}$ . And therefore, for every  $P_0 K_0$  choice so, the understand that I am you know like holding a  $P_0$  to a fixed value. And I am you know like taking different values of  $K_0$  depending upon the hamming class  $w$  and then I am building templates. So, all these templates are essentially you know like define are

defined by this parameter of  $m$  and  $c$ . So, therefore, they are kind of you know like define by this parameter of  $m$  and  $c$  which I have already denoted. So,  $m$  is again the mean vector and  $c$  is the covariance matrix.

So, therefore, I basically build template for  $K=0$  which belongs to different hamming class  $w$  ok. So, the different hamming class  $w$  is there will be 9 possible templates in this fashion. So, let us try see is an example to understand how it would look like.

(Refer Slide Time: 18:47)

**A Tale of Two Templates**

- Let us create two templates with Hamming Weight 0 and 4.

Mean Vector for Class 0 - Hamming Weight 0  
 [31.08695652 31.56521739 50.91304348 60.86956522 51.7826087 41.69565217 46.73913043 59.56521739 64.04347826 58.26086957]

Covariance Matrix for Class 0 - Hamming Weight 0  
 [[2.90118577 2.22134387 2.05335968 1.87549407 1.79249012 2.70948617 3.06916996 2.13043478 1.54150198 1.88537549]  
 [2.22134387 4.16600791 4.5513834 2.80494783 1.44664032 2.63438735 4.10869565 3.62055336 2.20158103 1.98221344]  
 [2.05335968 4.5513834 6.71936759 3.94268775 2.11660079 2.83596838 5.06719368 4.5513834 3.04940711 2.38735178]  
 [1.87549407 2.80434783 3.94268775 3.39130435 2.06126482 2.00395257 3.46442688 3.21343874 2.64229249 1.94464403]  
 [1.79249012 1.44664032 2.11660079 2.06126482 3.17786561 1.24901186 2.34980237 2.08300395 2.23715415 1.9229249 1  
 [2.70948617 2.63438735 2.83596838 2.00395257 1.24901186 4.03952569 3.82608696 2.36166008 1.65019763 1.94664032]  
 [3.06916996 4.10869565 5.06719368 3.46442688 2.34980237 3.82608696 5.29249012 4.01778656 2.96640316 2.61660079]  
 [2.13043478 3.62055336 4.5513834 3.21343874 2.08300395 2.36166008 4.01778656 4.07509881 2.83794466 2.11857708]  
 [1.54150198 2.20158103 3.04940711 2.64229249 2.23715415 1.65019763 2.96640316 2.83794466 3.22529644 1.8972332 ]  
 [1.88537549 1.98221344 2.38735178 1.94464403 1.9229249 1.94664032 2.61660079 2.11857708 1.8972332 2.29249012]]

And for example, this at you know like so, I am talking about two templates here you know like. For example, in the first case you have a humming weight 0; that means, it is all 0 and in the second case I will form a template where the template I will essentially the  $K=0$  will have a hamming weight of 4 ok.

So, there are eight chose possible things. So, therefore, here is an observed that the first thing in the temperate is the mean vector ok. So, mean vector would essentially has got 10 points because I am observing 10 time instances in the power traces. So, this is my mean and this is my covariance matrix ok. So, this is my corresponding covariance matrix you can observe that it is symmetrical symmetric matrix. So, basically we get a covariance matrix like this we again make a template for hamming weight 4.

(Refer Slide Time: 19:27)

Template for Hamming Weight 4

```
Mean Vector for Class 1 - Hamming Weight 4
[31.84227642 31.93333333 50.49430894 61.27100271 52.58373984 42.69376694 46.9696477 59.66233062 64.62547425
58.69322493]

Covariance Matrix for Class 1 - Hamming Weight 4
[[7.18172583 4.63698482 2.56824507 3.32747385 5.11652294 5.52488404 3.98544874 2.62838298 3.1274474 4.57197855]
[4.63698482 6.28893709 4.90607375 3.3087491 3.15607375 4.61677513 5.11836587 4.1461316 2.97230658 3.07281273]
[2.56824507 4.90607375 6.21214221 3.42182512 1.98136077 3.12217167 4.58062977 4.52220958 3.05345308 2.17721196]
[3.32747385 3.3087491 3.42182512 4.18899059 3.39974516 3.1611261 2.97731895 3.07257846 3.18072824 3.1314466 ]
[5.11652294 3.15607375 1.98136077 3.39974516 5.53270726 4.35076627 2.72597041 2.11912685 3.13902615 4.24913056]
[5.52488404 4.61677513 3.12217167 3.1611261 4.35076627 5.72558477 4.1436285 2.88677701 2.89066863 3.94884603]
[3.98544874 5.11836587 4.58062977 2.97731895 2.72597041 4.1436285 5.40254895 3.9040621 2.70934195 2.72549924]
[2.62838298 4.1461316 4.52220958 3.07257846 2.11912685 2.88677701 3.9040621 4.47756615 2.77910546 2.15232057]
[3.1274474 2.97230658 3.05345308 3.18072824 3.13902615 2.89066863 2.70934195 2.77910546 3.6042314 2.92734188]
[4.57197855 3.07281273 2.17721196 3.1314466 4.24913056 3.94884603 2.72549924 2.15232057 2.92734188 4.45139117]]
```

So, this is my you know like the template for hamming weight four and I essentially keep these two things as my templates.

(Refer Slide Time: 19:37)

The Attack Phase

- Later on we use these template characterizations to identify the unknown key bytes from a power trace from the device under attack.
- This means, we evaluate the probability density function of the multivariate normal distribution  $(\mathbf{m}, \mathbf{c})_{P_0, K_0}$  and the power trace of the device under attack.
  - Given a power trace  $t$  of the device under attack, and a template  $(\mathbf{m}, \mathbf{c})_{P_0, K_0}$  we compute the probability:

$$p(t; (\mathbf{m}, \mathbf{c})_{P_0, K_0}) = \frac{\exp(-\frac{1}{2}(t - \mathbf{m})' \mathbf{c}^{-1} (t - \mathbf{m}))}{\sqrt{(2\pi)^n \det(\mathbf{c})}}$$

Note that  $t$  is accumulated by keeping  $P_0$  say fixed and choosing arbitrary values for the remaining 15 plaintext bytes. The key is unknown here and we have no handle. In a setting, where we don't have access to the input also, we can target the output of  $P_0 \oplus K_0$  for building the templates.

And now I basically go to the attack phase. So, in the attack phase right the attacker essentially we will use this previously done template characterization to identify the unknown key bytes from a power trace from the device and the attack. So; that means, you have now the device and the attack which has been given to you note that you do not know the internal key now because key is unknown to you and need to determine. So, I

am assuming this point right that what I will do is basically I will try to get a power trace from this you know like un[der]- device under attack and there I will try to correlate or you know like match with my existing templates. So, this means that we basically what we will do is we will evaluate the probability density function on the pdf of the multivariate normal distribution and which is essentially denoted as  $m, c$  suffix by  $P_0$  and  $K_0$  and the power of the device under attack. So, basically I have the template  $m, c$  and I have a power trace which I denote by say the vector  $t$  ok.

So, the vector  $t$  here denotes a given power trace. So, therefore, what I mean is that now you have got you know like this power trace. So, this is my corresponding power trace which is denoted as  $t$  and this is my corresponding template and I am trying to fi[nd]- basically I am trying to fit it into the multivariate normal distribution formula and I am trying to understand how much close is it to the actual template ok.

So, basically here I replace in one case I keep that you know like this  $t$  and I kind of evaluate this function with a given value of  $K$  so; that means, the template for  $K_0$  I again reevaluate this reevaluate this for with you know like the with the template for say  $K_1$ . So, therefore, right which what I do is we usually given a power trace of the device under attack and a template  $m, c$  we basically compute this by just plugging instead of  $x$ , we plug in  $t$  ok. So, instead of  $x$  we plug in the value of  $t$  here and note that  $t$  is basically accumulated by keeping basically the trace that you are accumulating we are basically getting it by keeping  $t_0$  say fixed and choosing arbitrary values for the remaining 15 plaintext bytes ok.

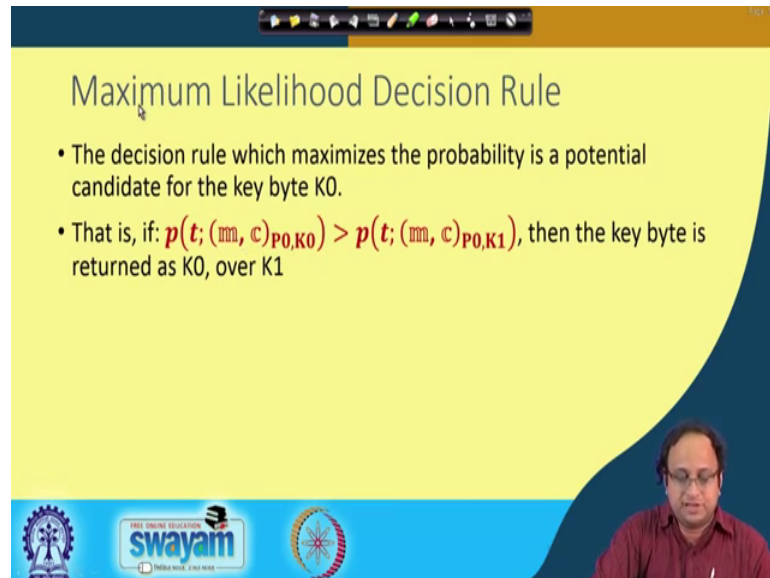
So, the remaining plaintext 15 bytes I am not bothered, but I am assuming in this case that I have got kind of a control on the input I do not know what is the internal key and I have got no knowledge about the about the secret key. So, the key is unknown here and we have got no handle on that. You can easily you know like generalize it to a setting where you do not have access to the input although I am assuming here the that I have got an access to the input by you know like targeting. For example, the output of  $P_0$  comma XOR with  $K_0$  and building template at that point. So, in that case like just the knowledge of  $P_0$  we will give you the corresponding key value ok.

. So, therefore, right now I am just you know like doing with template in this fashion, but you can journalize it to is to scenario probably where you do not have access to the input



even, but you know the input of course,. so. So, therefore, right I mean. So, therefore, with this background right what we do now is ok. So, let clear these up.

(Refer Slide Time: 22:47)



Maximum Likelihood Decision Rule

- The decision rule which maximizes the probability is a potential candidate for the key byte K0.
- That is, if:  $p(t; (m, c)_{P0,K0}) > p(t; (m, c)_{P0,K1})$ , then the key byte is returned as K0, over K1

So, now what we do is we will be discuss rule which is called as the maximum likelihood decision rule which is basically the decision rule which maximizes the probability which essentially you know like gives me the potential candidate for the key byte K 0.

So, for example, right we have developed template for say K 0 where the hamming weight is 0 and we have you like found out the we have also created a template where the hamming weight is say 4. And that is denoted as K 1; K 1 would stand for all those keys whose hamming weight is 4. So, now, I take a power trace t and if I plug it into this formula and if I get that you know like this is greater than this, then the maximum likelihood decision rule tells me that the key by which I will return is K 0 over K 1. So, I basically have the preference for K 0 compare to K 1.

(Refer Slide Time: 23:39)

The slide is titled "Case Study for Unknown Key" and contains the following text:

- Unknown key has a  $K_0$  value of Hamming Weight 4
- We compute the probabilities as mentioned wrt. the templates for 0 and 4.
- Hamming Weight 0 Prob: 0.00000725
- Hamming Weight 4 Prob: 0.00001790
- This shows that it more likely that  $K_0$  has a Hamming Weight it 4, which is indeed correct!

The slide also features a video inset of a man speaking in the bottom right corner, and logos for "swayam" and "INDIA WIDE, 24x7 WIDE" at the bottom.

So, therefore, right let us see what happens with our case study. So, we basically have an unknown key which has a  $K_0$  value of humming weight 4 in our case. And we compute the probabilities as mentioned with respect to the template for 0 and 4. So, we basically observe a power trace for the for 0 key and we basically develop the template for 0 and 4 as we have already discussed in the previous class in the in the previous slide.

And then we observe we basically take the power trace for the unknown key which essentially in this case have a humming weight 4. And we calculate the two probabilities right we basically in one case calculate the likelihood with respect to with respect to 0 and in other case we calculate the likelihood with respect to 4 ok.

. So, you see that the likelihood with respect to 4 is more compared to that with 0 indicating that you know like it is more likely that the  $K_0$  will have a hamming weight of 4 which is essentially the correct result.

(Refer Slide Time: 24:35)

For Another Few Runs

- Trial 2:
  - Hamming Weight 0 Prob: 0.00000000
  - Hamming Weight 4 Prob: 0.00000336
- Trial 3:
  - Hamming Weight 0 Prob: 0.00000000
  - Hamming Weight 4 Prob: 0.00000228

• In all cases Hamming Weight 4 has a more likelihood from the template analysis

The slide features a yellow background with a dark blue curved border on the right. At the bottom, there are logos for Swamyam and other educational institutions, along with a small video inset of a man in a red shirt.

So, if you repeat this right for few more runs for example, you see still see that you know that the this probability is more than this probability indicating that indeed this is probably the most preferable candidate and likewise for another run also we get similarly consistent result. So, in all cases the hamming weight 4 has a more likelihood from template analysis.

(Refer Slide Time: 24:55)

Numeric Problems in Template Analysis

- While performing the template analysis, we can get into 2 important numeric problems:
  - These are related to the covariance matrix.
  - Size of the matrix depends on the number of important points, say  $n$ .
  - This must be chosen carefully.
  - There can also be problems arising out of the requirement of the existence of its inverse.
  - Also the exponent tends to be smaller and leading to inaccuracies.

The slide features a yellow background with a dark blue curved border on the right. At the bottom, there are logos for Swamyam and other educational institutions, along with a small video inset of a man in a red shirt.

So, now, there can be you know like some numeric problems which can arise in temperate analysis. So, while performing the temple analysis we can get it into two

important numeric problems. The first problem is which is related to the covariance matrix as we know that we have to basically calculate the covariance matrix. And the covariance matrix right, we will essentially the dimension on the covariance matrix. We will determine on the number of important items like important you know the window size. So, therefore, the size of the matrix depends on the number of important point say  $n$  and this must be chosen carefully. Because you if you make it very less then maybe you will lose information and you will make it very high really computationally quiet complex.

and there can also be problems arising out of the requirement of the existing of it inverse remember that we had a determinant  $c$  which needs to be existing right. And therefore, right that probably can if there some numeric problems right then with respect to a numerical act inaccuracies and so on. You know like floating point (Refer Time: 25:48) stuff like that then you may will it go into a scenario where the inverse does not exist. Also the exponent right tends to be smaller and lead to inaccuracies because you have an exponential vector and if it becomes too small then again there can be inaccuracies.

(Refer Slide Time: 26:05)

The slide is titled "Numeric Manipulations" and contains the following text:

- Let us take the logarithm of the value  $p(t; (m, c)_{P0, K0})$
- Thus,
 
$$\ln p(t; (m, c)_{P0, K0}) = -\frac{1}{2} \left( \ln((2\pi)^n \det(C)) + (t - m)' c^{-1} (t - m) \right)$$

The template that leads to the smallest absolute value of the logarithm of the probability indicates the correct key.

- Again if,  $\ln p(t; (m, c)_{P0, K0}) < \ln p(t; (m, c)_{P0, Kl}), \forall l \neq 0$ , then  $K0$  is the predicted class.

The slide also features logos for "swayam" and "INDIA'S OPEN UNIVERSITY" at the bottom, along with a small video feed of a man in a red shirt.

So, one way of elevating this is basically you take the logarithmic of the value. So, therefore, if I take the logarithm of the value and remember the probability is lesser than 0; I mean sorry lesser than 1. So, then right it be would implies that the template that

now leads to the smallest absolute value of the logarithm of the probability will indicate the correct key ok.

So, in the previous case right the probability we have maximizing. So, now, if I take the logarithm right then it will means that I have to have to look for the smallest absolute value. So, therefore, if I take the ln on both sides you see that this is my corresponding equation. So, therefore, the in this case right my decision rule will be that if I come find out the ln of the probability right with respect to  $K_0$  like the template for  $K_0$  and if this lesser than this then  $K_0$  is my predicted plus.

(Refer Slide Time: 26:51)

**Reduced Templates**

- To further avoid problems with the covariance matrix we set covariance to the identity matrix:
  - We are neglecting the effect of the covariances between the points in the window we are observing.
  - This template is called reduced template.
- Thus we have,

$$p(t; m) = \frac{\exp(-\frac{1}{2}(t - m)'(t - m))}{\sqrt{(2\pi)^n}}$$

- Again taking logarithms,

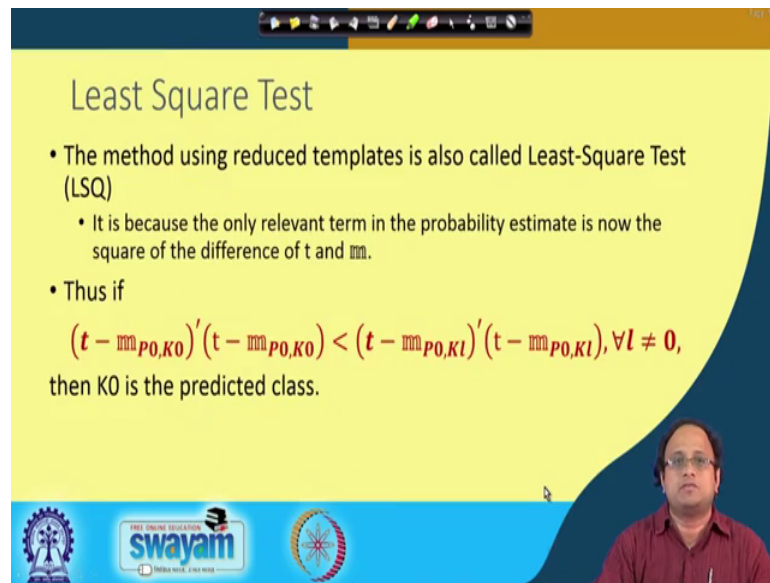
$$\ln p(t; m) = -\frac{1}{2}(\ln((2\pi)^n) + (t - m)'(t - m))$$

swayam

So, likewise right I mean you can actually this means has to something which is called as reduce templates which is to further avoid the problems with the covariance matrix. We actually send the covariance matrix. So, we basically make it identity; that means, in a way right we are neglecting the effect of the covariance is between the points in the window and this template is what is technically called as the reduced template. So, you can if you test take this reduced template and you define this equation, then you see that we have removed c because c in this case is the identity. And therefore, and it has got no play here no role here and therefore, this is the corresponding simplification that I get ok.

. So, again if I take logarithm is because of the reason which I said, then this is the factor which you get.

(Refer Slide Time: 27:35)



### Least Square Test

- The method using reduced templates is also called Least-Square Test (LSQ)
  - It is because the only relevant term in the probability estimate is now the square of the difference of  $t$  and  $m$ .
- Thus if

$$(t - m_{p0,K0})'(t - m_{p0,K0}) < (t - m_{p0,Kl})'(t - m_{p0,Kl}), \forall l \neq 0,$$

then  $K0$  is the predicted class.

FREE ONLINE EDUCATION  
swamyam  
INDIA WISE, LEAD WISE

So, therefore, right now right we will again. So, this method right is also called which is where we using this reduce templates is also called as a Least Square Test, LST. It is because the only relevant terms in the probability estimate is now the square of the difference of  $t$  and  $m$  like if you see the formula right you have got two parts where this parts is essentially has got no role over here it to compared. But this is the most important part which is nothing, but the vector  $t$  minus  $m$  transpose multiplied with  $t$  minus  $m$  which is pretty much nothing, but the square of the differences. So, therefore, right essentially you are the idea is again like the this is essentially lesser than this ok, then you will basically vote for  $K0$  as your predicted class.

(Refer Slide Time: 28:17)

LSQ in our Case Study

- Trial 1:
  - Hamming Weight 0 Squared Difference: 19.07939509
  - Hamming Weight 4 Squared Difference: 32.18028070
- Trial 2:
  - Hamming Weight 0 Squared Difference: 57.86200378
  - Hamming Weight 4 Squared Difference: 36.24640536
- Trial 3:
  - Hamming Weight 0 Squared Difference: 94.03591682
  - Hamming Weight 4 Squared Difference: 66.37973869

So, if you now take this and apply again to our case study again for the three trials that you have seen you see that the hamming weight of 0 I mean if we just compare right you will see that it is the interesting thing that initially right. You got a wrong observation right; you see that this is larger than this, but for certain for the other runs you see that this is 36 compared to 67, this 66 compared to 94. So, therefore, I will probably vote for 4 has being the correct hamming class. So, note that we got an error in this in one of the trials which is also perfectly possible.

(Refer Slide Time: 28:51)

**Conclusions**

Correlation across points in the trace leads to a multivariate normal distribution of noise.

Template attacks try to accurately model this noise.

Template matching is based on the maximum likelihood principle.

Reduced Templates help in performing template analysis with less numerical instability.

**References:**

1. Stefan Mangard, Elisabeth Oswald, Thomas Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards, Springer.
2. Suresh Chari, Joysula Rao and Pankaj Rohatgi, Template Attacks, CHES 2002.
3. Christian Rechberger, Elisabeth Oswald, Practical Template Attacks.



So, therefore, to conclude what we discussed correlation across the points in the trace leads to a multivariate normal distribution of noise and template attacks try to use this by accurately model the noise. And template matching is based on the maximum likelihood principle and reduce templates help us in performing template analysis with less numerical stability. Some of the references that I have used for this discussion is something that you can also have a look.

So, this is the textbook essentially it is also often commonly called as the DPA book written by Stefan Mangard, Elisabeth Oswald and Thomas Popp. And there are some other references like papers which you can lead to with the classic paper on template attacks which is published in chess in 2002 and followed by a note on practical template attacks by Christian Rechberger and Elizabeth Oswald. So, with this I would like to thank you for your attention and we shall continue in the next class.

Thank you.