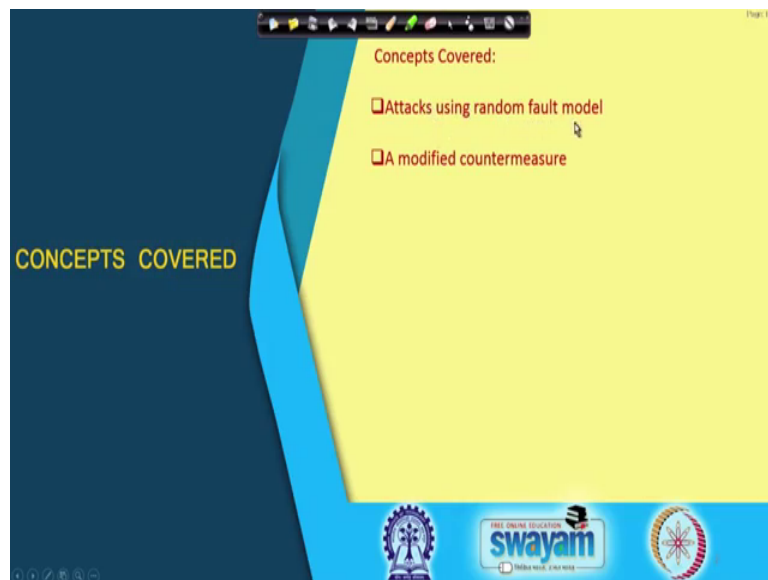


Hardware Security
Prof. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 54
Infective Countermeasures for DFA (Contd.)

So, welcome back to this class on Hardware Security. So, we shall be continuing our attack discussion on the infective countermeasure that we started to discuss in the last two classes.

(Refer Slide Time: 00:26)



So, in particular we shall be talking about the attack using random fault models and finally, we shall try to modify this counter measure to prevent these attacks or these attacks.

(Refer Slide Time: 00:32)

A Proposed Infective Countermeasure

Input: P, k^j for $j \in \{1, \dots, n\}$, (β, k^0) , $(n = 11)$ for AES-128

Output: $C = \text{BlockCipher}(P, K)$

- 1: State $R_0 \leftarrow P$, Redundant state $R_1 \leftarrow P$, Dummy state $R_2 \leftarrow \beta$
- 2: $C_0 \leftarrow 0, C_1 \leftarrow 0, C_2 \leftarrow \beta, i \leftarrow 1$
- 3: **while** $\text{do} \leq 2n \text{ do}$
- 4: $\lambda \leftarrow \text{RandomBit}()$ // $\lambda = 0$ implies a dummy round
- 5: $\kappa \leftarrow (i \wedge \lambda) \oplus 2(-\lambda)$
- 6: $\tau \leftarrow \lambda \cdot \lfloor i/2 \rfloor$ // τ is actual round counter, 0 for dummy
- 7: $R_i \leftarrow \text{RoundFunction}(R_i, k^i)$
- 8: $C_i \leftarrow R_i \oplus C_2 \oplus \beta$ // infect C_i to propagate a fault
- 9: $\varepsilon \leftarrow \lambda(-i \wedge 1) \cdot \text{SNLF}(C_0 \oplus C_1)$ // check if i is even
- 10: $R_2 \leftarrow R_2 \oplus \varepsilon$
- 11: $R_0 \leftarrow R_0 \oplus \varepsilon$
- 12: $i \leftarrow i + \lambda$
- 13: **end while**
- 14: $R_0 \leftarrow R_0 \oplus \text{RoundFunction}(R_2, k^0) \oplus \beta$
- 15: **return** R_0


Source: Sikhar Patranabis and Debdeep Mukhopadhyay (Eds.), Fault Tolerant Architectures for Cryptography and Hardware Security, Springer.

$i = 1, \dots, 2n$
 $\lambda = 0 \Rightarrow \text{Dummy} \Rightarrow \kappa = 2, \tau = 0.$
 $\lambda = 1 \Rightarrow \kappa = \text{lsb}(i), \tau = \lfloor i/2 \rfloor$

Red. Cipher

Dummy

Red. Cipher



So, this is the infective countermeasure that we were discussing. And we already saw an attack using a constant fault model. What we were studying is whether we can do an attack using a random fault model as we have done in the classical DFA on AES.


(Refer Slide Time: 00:48)

Classical DFA on the Countermeasure: Relaxing the Restrictions of FDTC 2013 Attack

- We can still attack using a **random byte fault** model as in the classical DFA.
- We start with an assumption that there is no dummy round after this fault injection except the final compulsory dummy round.

- The attack targets the penultimate round of AES, e.g. in case of AES128, input of 9th round is the target.
- Fault f in I_0^9 , i.e., first byte of the top row in the input of 9th cipher round
- Countermeasure infects faulty computation thrice
 - ▶ After the execution of 9th cipher round
 - ▶ After the execution of 10th cipher round
 - ▶ After the execution of compulsory dummy round

$$I_0^9 \oplus \begin{pmatrix} I_0^9 \oplus f & I_4^9 & I_8^9 & I_{12}^9 \\ I_1^9 & I_5^9 & I_9^9 & I_{13}^9 \\ I_2^9 & I_6^9 & I_{10}^9 & I_{14}^9 \\ I_3^9 & I_7^9 & I_{11}^9 & I_{15}^9 \end{pmatrix}$$



So, right; so, here we basically have a you know like where we stopped in the last class. So, there is a random fault that has been induced in the 0th byte of the 9th round cipher. And therefore, right we have an infection shown here. Now this fault right would propagate through three subsequent computations where the fault will basically infect

further. One is after the execution of the 9th cipher rounds, second time is up to the execution of the 10th 10th cipher round and finally, after the execution of the compulsory dummy round.

(Refer Slide Time: 01:14)

The slide, titled "Differential after 9th Cipher Round", illustrates the propagation of a differential through a cipher round. It features three main components:

- Initial Differential Matrix:** A matrix representing the differential after the 9th round:

$$R_0 \oplus R_1 = \begin{pmatrix} A & 0 & 0 & 0 \\ B & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ D & 0 & 0 & 0 \end{pmatrix}$$
- Legend:** A box defining the values of A, B, C, and D:

$$\begin{aligned} A &= 2f' \\ B &= f' \\ C &= f' \\ D &= 3f' \end{aligned}$$
- Block Diagram:** A schematic of a cipher round with four parallel paths (A, B, C, D) and a central function block. A fault ϵ is introduced at the input of the function block. The paths are connected to XOR gates and the function block.

Below the diagram, two equations describe the state after the round function:

$$R_x \leftarrow \text{RoundFunction}(R_x, k^c)$$

$$R_0 \leftarrow R_0 \oplus \epsilon$$

The slide also includes a matrix for the differential after the round function:

$$R_0 \oplus R_1 = \begin{pmatrix} A \oplus \text{SNLF}[A] & 0 & 0 & 0 \\ B \oplus \text{SNLF}[B] & 0 & 0 & 0 \\ C \oplus \text{SNLF}[C] & 0 & 0 & 0 \\ D \oplus \text{SNLF}[D] & 0 & 0 & 0 \end{pmatrix}$$

At the bottom, there are logos for Swamyam and other educational institutions.

So, therefore, let us see pictographically a pictorially you know like how the faults will propagate. So, the first right is essentially the when you are basically doing this computation. Remember that you know like after the 10th after the 9th round cipher you basically measure the difference by taking these XOR which is essential epsilon. So, therefore, this epsilon gets further modified by SNLF and therefore, you get SNLF epsilon.

So now, what you have till now is we had an initial fault f. Now, because of the 9th round cipher this fault will propagate to the column ok. To we will propagate it was in the first byte it will propagate in the first column. So, therefore, imagine that the values are A B C D and that stands for something like 2 of f dash and B is f dash C is f dash and D is 3 of f dash ok.

So, note that this f dash essential is a result of the fault being applied on the, you know like on the individual on the fault like for example, f we had a fault which is equal to f. That got modified by the S box to something which is f dash and then writes the f dash was further modified by the mixed columns. And therefore, you have this relationship as we have seen in the context of classical DFA on AES.

So now what we essentially would like to do is basically write I mean see that what happens subsequent to this. So, for example, like subsequent to this you can observe that what will happen is that the cipher round will get affected first. So, if the cipher gets affected first then what will happen is that, this these differences like A B C D these are the various values of epsilon that are coming over here. And therefore, they will get modified into SNLF A, SNLF B, SNLF C and SNLF D and that will modify the cipher.

So, therefore, now if I compute the XOR of R 0 and R 1 remember that R 1 is my redundant round output which is essentially correctly computed because the 9th round redundant cipher was already computed. And that is the correct value that we was computed. And therefore, the error here was A B C D, but now because of this contamination will become a plus SNLF A, it will become B plus SNLF B and so on.

So, this is the result of this infection where this infection is you know like R 0 is basically R 0 XOR with the further error.

(Refer Slide Time: 03:31)

Differential after 9th Cipher Round

The dummy round also gets infected.

$A = 2f'$
 $B = f'$
 $C = f'$
 $D = 3f'$

Before,

$$R_2 = \begin{pmatrix} \beta_0 & \beta_4 & \beta_8 & \beta_{12} \\ \beta_1 & \beta_5 & \beta_9 & \beta_{13} \\ \beta_2 & \beta_6 & \beta_{10} & \beta_{14} \\ \beta_3 & \beta_7 & \beta_{11} & \beta_{15} \end{pmatrix}$$

After the infection,

$$R_2 = \begin{pmatrix} \beta_0 \oplus \text{SNLF}[A] & \beta_4 & \beta_8 & \beta_{12} \\ \beta_1 \oplus \text{SNLF}[B] & \beta_5 & \beta_9 & \beta_{13} \\ \beta_2 \oplus \text{SNLF}[C] & \beta_6 & \beta_{10} & \beta_{14} \\ \beta_3 \oplus \text{SNLF}[D] & \beta_7 & \beta_{11} & \beta_{15} \end{pmatrix}$$

$R_2 \leftarrow R_2 \oplus \epsilon$

So now there will also be an infection because of this right on the dummy round. Because remember that there is also an infection on the dummy round which happens which is here ok. So, for example, this dummy round is also getting affected because of the fault, but these dummy round initially write has a value of all betas. So, this is shown here as beta 0 beta 1 beta 2 and so on.

But now because of this error this particular column of betas will get affected by SNLF A, SNLF B, SNLF C and SNLF D. So, you will have beta plus SNLF A, beta plus beta 1 plus SNLF B beta 2 plus SNLF C and this will become beta 3 plus SNLF D and that is the result of this equation R 2 is R 2 XOR with epsilon.

(Refer Slide Time: 04:12)

The slide, titled "Differential after 10th Cipher Round", contains the following mathematical content:

- $$A = 2f',$$

$$B = f',$$

$$C = f',$$

$$D = 3f'$$
- $$R_0 \oplus R_1 = \begin{pmatrix} z_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & z_1 \\ 0 & 0 & z_2 & 0 \\ 0 & z_3 & 0 & 0 \end{pmatrix}$$
- $$z_0 = S[l_0^0 \oplus A \oplus SNLF[A]] \oplus S[l_0^0]$$

$$z_1 = S[l_1^0 \oplus B \oplus SNLF[B]] \oplus S[l_1^0]$$

$$z_2 = S[l_2^0 \oplus C \oplus SNLF[C]] \oplus S[l_2^0]$$

$$z_3 = S[l_3^0 \oplus D \oplus SNLF[D]] \oplus S[l_3^0]$$
- $$R_x \leftarrow RoundFunction(R_x, k^x)$$
- $$R_0 \oplus R_1 = \begin{pmatrix} m_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_1 \\ 0 & 0 & m_2 & 0 \\ 0 & m_3 & 0 & 0 \end{pmatrix}$$
- $$R_0 \leftarrow R_0 \oplus \epsilon$$
- $$m_j = z_j \oplus SNLF[z_j], j \in \{0,1,2,3\}$$

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

So now what will happen is that, this will again as I said that the next infection will happen after the 10th cipher round. So, because in the redundant round there is no you know like propagation of this error, but this error will propagate in after the 10th cipher round. And there what will happen is that because of this right the round function corresponding to this remember that you already had the error as shown here as you know like 2 f dash, f dash, f dash and 3 f dash.

Now, because of this particular round function computation, this round function will have it is own shift row and mixed columns and so on. So, because of the shift row this column, this disturbance in this column, we will now gets spread into the first column and all the individual columns because of the shift row. So, therefore, right this value of z 0, if you observe is nothing but in one case right it was I 0 10. For example, this is the correct ciphered input at the, you know like of the 10th round. Whereas, the corresponding faulty one essentially was infected by A plus SNLF A ok. And likewise here it was B plus SNLF B and here it was C plus SNLF C and here it was D plus SNLF

D, which essentially gets XOR by an S box and again this is my corresponding differential and therefore, this differentials stands for z_0 likewise for z_1 , z_2 and z_3 .

So now after this right again there will a contamination of the cipher round, because of this equation $R_0 + \epsilon$. And therefore, this z_0 will get further contaminated and become SNLF z_0 ok. And therefore, now if I take the XOR of R_0 and R_1 , I will get some m_0 , m_1 , m_2 and m_3 are this positions, but this m_i for example, or m_j will basically be nothing but the corresponding z , XOR with the corresponding SNLF z .

So, it will be m_0 will be equal to z_0 XOR with SNLF z_0 , m_1 will be z_1 XOR with SNLF z_1 , m_2 will be you know like equivalently is a you know like z_2 XOR with SNLF z_2 and so on.

(Refer Slide Time: 06:15)

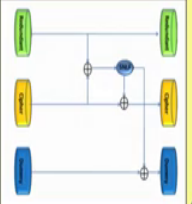
Differential after 10th Cipher Round

The dummy round is also infected.

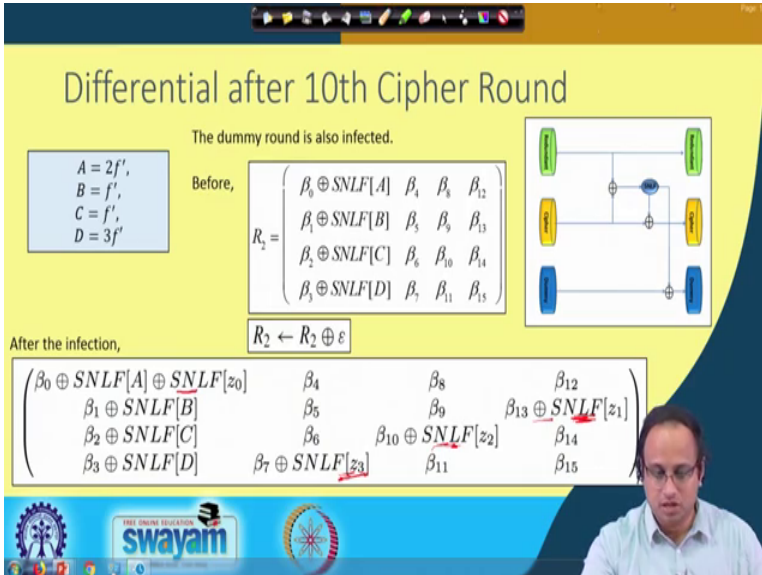
$A = 2f'$
 $B = f'$
 $C = f'$
 $D = 3f'$

Before,

$$R_2 = \begin{pmatrix} \beta_0 \oplus \text{SNLF}[A] & \beta_1 & \beta_8 & \beta_{12} \\ \beta_1 \oplus \text{SNLF}[B] & \beta_2 & \beta_9 & \beta_{13} \\ \beta_2 \oplus \text{SNLF}[C] & \beta_6 & \beta_{10} & \beta_{14} \\ \beta_3 \oplus \text{SNLF}[D] & \beta_7 & \beta_{11} & \beta_{15} \end{pmatrix}$$



After the infection, $R_2 \leftarrow R_2 \oplus \epsilon$

$$\begin{pmatrix} \beta_0 \oplus \text{SNLF}[A] \oplus \text{SNLF}[z_0] & \beta_4 & \beta_8 & \beta_{12} \\ \beta_1 \oplus \text{SNLF}[B] & \beta_5 & \beta_9 & \beta_{13} \oplus \text{SNLF}[z_1] \\ \beta_2 \oplus \text{SNLF}[C] & \beta_6 & \beta_{10} \oplus \text{SNLF}[z_2] & \beta_{14} \\ \beta_3 \oplus \text{SNLF}[D] & \beta_7 \oplus \text{SNLF}[z_3] & \beta_{11} & \beta_{15} \end{pmatrix}$$


So, therefore, right after this what will happen is again the dummy round corresponding to these will effected remember the dummy round diffusion was now in the previous to this was in this column.

Now, what will happen is because of this particular error remember that error right essentially is nothing but SNLF of z_0 , SNLF of z_1 , SNLF of z_2 and z_3 . But the positions are here, the positions right for this particular error are at this position, this position and this position and this position ok.

So, therefore, write there will be this contaminations which are effecting further this you know like XOR of R 2. So, this is your you know the value of beta; the value of R 2. So, the what will happen is there already we had this diffusion in the first column. So, this diffusions will stay like or this is the disturbance of SNLF A, SNLF B, SNLF C and SNLF D will state, but further write there will be additional infections because of these values. So, this will be the state after the 10th cipher round.

Now, right there will be a further step after this and that essentially will be because of the compulsory dummy round

(Refer Slide Time: 07:27)

Further Infection due to Final Compulsory Dummy Round

$$R_0 \leftarrow R_0 \oplus \text{RoundFunction}(R_2, k^0) \oplus \beta$$

Also, $\text{RoundFunction}(R_2, k_0) \oplus \beta = \text{MC}(SR(S(R_2) \oplus S(\beta)))$

Here, $S(R_2) \oplus S(\beta)$ is:

$S[\beta_0 \oplus \text{SNLF}[A] \oplus \text{SNLF}[z,]] \oplus S[\beta_0]$	0	0	0
$S[\beta_1 \oplus \text{SNLF}[B]] \oplus \text{SNLF}[\beta_1]$	0	0	$S[\beta_0 \oplus \text{SNLF}[z,]] \oplus S[\beta_0]$
$S[\beta_2 \oplus \text{SNLF}[C]] \oplus \text{SNLF}[\beta_2]$	0	$S[\beta_0 \oplus \text{SNLF}[z,]] \oplus S[\beta_0]$	0
$S[\beta_3 \oplus \text{SNLF}[D]] \oplus \text{SNLF}[\beta_3]$	$S[\beta_0 \oplus \text{SNLF}[z,]] \oplus S[\beta_0]$	0	0

$\therefore \text{MC}(SR(S(R_2) \oplus S(\beta)))$

F_1 0 0 0	F_1 0 0 0	$2F_1 + F_2$ F_3 $3F_4 + F_5 + F_6$ $3F_7$
F_2 0 0 F_4	0 0 F_4 F_5	$F_3 + 3F_2$ F_3 $2F_4 + 3F_5 + F_6$ $2F_7$
F_3 0 F_2 0	F_3 0 F_4 0	$F_3 + 2F_2$ $3F_3$ $F_4 + 2F_5 + 3F_6$ F_7
F_4 F_4 0 0	0 F_3 F_4 0	$3F_3 + F_2$ $2F_3$ $F_4 + F_5 + 2F_6$ F_7

So, what will happen is because of the final compulsory dummy round, you will basically execute this operation ok. So, remember right this part if I just concentrated on the round function R 2 comma k 0 XOR with beta because of the flow that we mention even in the previous attack, what you can do is essentially you can take the mixed column and the shift row column and then internally you have got s of R 2 XOR with s of beta. Remember that R 2 is not anymore beta.

So now if you take the XOR of S or R 2 and S or beta, then you observe here that everything here is like this positions for example, beta 4, beta 5, beta 6, beta 8, beta 9, beta 12, beta 14, beta 15 and beta 11 remains the same as that in the original beta matrix. So, therefore, if I apply S on them sub bytes on them and take the XOR, then S of R 2 XOR with s of beta, these positions I will still get 0. And whereas, right at these positions

I will get further diffusion and the diffusion will be because of you know like S of beta 0 XOR with these value likewise see you know like these value XOR with S of beta 1 and this value XOR with s of beta 2 and so on.

So, you know actually right this will be yeah so, basically I am just concentrating the difference the actually ok. So, therefore, write you can make this correction that this is S of beta 1 XOR with SNLF beta or B XOR with S of beta 1 ok. So, I am considering here so, let me make this correction this will be actually S, this will be S and this will also be S. So, I am considering the difference of the S box outputs.

So, likewise here already we have the S S. So, somehow this is the mistake over here. So now, if you apply you know like shift rows and mixed columns. So, you can kind of you know like denote these values as f 1, f 4 and so on basically so, just kind of substituted this f values for compactness.

And now, because of the shift row and the mixed column what will therefore, happen is because of the shift rows this value would right will get shifted like this and accordingly like I am considering one position shift. So, this f 7 will come here, this f 4 will come to this position and you will get this arrangement likewise write this f 2, f 5 and f 2 will basically swap so, you will have this value. And likewise f 3 and f 6 will shift by one location to the right so, you will get this.

So now, if I apply mixed columns again you have this matrix. So, if you apply then finally, right you will get this as your result. So, therefore, right what essentially is the meaning of this particular or influence of this is essentially observed here.

(Refer Slide Time: 10:01)

Further Infection due to Final Compulsory Dummy Round

$$R_0 \leftarrow R_0 \oplus \text{RoundFunction}(R_2, k^0) \oplus \beta$$

Thus the final difference is:

$$T \oplus T^* = \begin{pmatrix} m_0 \oplus 2F_1 \oplus 1F_2 & 1F_3 & 3F_4 \oplus 1F_5 \oplus 1F_6 & 3F_7 \\ 1F_1 \oplus 3F_2 & 1F_3 & 2F_4 \oplus 3F_5 \oplus 1F_6 & m_1 \oplus 2F_7 \\ 1F_1 \oplus 2F_2 & 3F_3 & m_2 \oplus 1F_4 \oplus 2F_5 \oplus 3F_6 & 1F_7 \\ 3F_1 \oplus 1F_2 & m_3 \oplus 2F_3 & 1F_4 \oplus 1F_5 \oplus 2F_6 & 1F_7 \end{pmatrix}$$

The slide also features logos for Swamyam and other educational institutions at the bottom.

That if you now take this difference right, this difference will further affect R_0 . And therefore, the final difference between the correct and the wrong right essentially will be this. Because remember that already we had a diffusion which was like this ok.

Now, because of the further diffusion which is coming because of these difference you will essentially get you know like this contaminating the differential, but you will also have the original m_0 , m_1 , m_2 , and m_3 differential being also there. But now; that means, right that the differences that we have basically infused or rather the infections that we have created, we observe that they are not random ok. For example, if I observe these 3 bytes I can easily get the value of f_3 . And therefore, I can remove this difference and get the original value of m_3 . Likewise, if I you know like considered for example, these 3 positions you have got f_1 and f_2 and therefore, right using that you can again obtain the value of f_1 , f_2 .

(Refer Slide Time: 11:06)

Now, the key!

$$T \oplus T^* = \begin{pmatrix} m_0 \oplus 2F_1 \oplus 1F_2 & 1F_3 & 3F_4 \oplus 1F_5 \oplus 1F_6 & 3F_7 \\ 1F_1 \oplus 3F_2 & 1F_3 & 2F_4 \oplus 3F_5 \oplus 1F_6 & m_1 \oplus 2F_7 \\ 1F_1 \oplus 2F_2 & 3F_3 & m_2 \oplus 1F_4 \oplus 2F_5 \oplus 3F_6 & 1F_7 \\ 3F_1 \oplus 1F_2 & m_3 \oplus 2F_3 & 1F_4 \oplus 1F_5 \oplus 2F_6 & 1F_7 \end{pmatrix}$$

F_1, F_2 F_3 F_4, F_5, F_6 F_7

Removing this effect, then we get the differential after the 10th Cipher Round infection we come to:

$$T \oplus T^* = \begin{pmatrix} m_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_1 \\ 0 & 0 & m_2 & 0 \\ 0 & m_3 & 0 & 0 \end{pmatrix}$$

Each $m_j = z_j \oplus SNLF[z_j], j \in \{0,1,2,3\}$
 Assuming, SNLF is finite-field inverse in $GF(2^8)$, we have 2 solutions for each z_j .

We know T, thus there are 2^4 solutions for T^* .

So, likewise you have got you know like here you know like for example, you can observe that what I basically saying is that you can solve this equation get the value of f_1, f_2 you can solve these get the value of f_3 . Likewise, here you can solve and get the value of f_4, f_5 and f_6 here you can solve the value of f_7 . And you can remove these disturbances and you can get the original value of the differences you know like m_0, m_1 and m_2 and m_3 which gets exposed to the adversary.

So, therefore, right the adversary now has got the value of m_j which stands for z_j XOR with SNLF z_j . If you remember previously and remember again the SNLF z_j is a finite field inverse. So, therefore, you would expect that there will be 2 solutions for each z_j . And therefore, the number of values right for z_0, z_1, z_2 and z_3 will be 2 to the power of 4 because every value of z_0 will have 2 possible solutions.

So, therefore, right if I already know the value of T. So, the question is how many values of T^* will be there. So, you see that if I know the value of T which I have obtain from the cipher then the possible value of T^* can be maximum 2 to the power of 4. Because the difference right here for example, m_0, m_1, m_2 and m_3 this can now take you know like potentially 2 to the power of 4 values ok. And therefore, right there will be equally equal amount of you know I mean the XOR essentially also can be 2 to the power of 4, I mean there can be 2 to the power of 4 solutions for T^* .

(Refer Slide Time: 12:30)

The Equations for Key-Retrieval

T and T* are 10th round outputs.
Now, we can guess the 10th round keys (a quartet), and check for the expected relations in 9th round output.

$$R_0 \oplus R_1 = \begin{pmatrix} A \oplus SNLF[A] & 0 & 0 & 0 \\ B \oplus SNLF[B] & 0 & 0 & 0 \\ C \oplus SNLF[C] & 0 & 0 & 0 \\ D \oplus SNLF[D] & 0 & 0 & 0 \end{pmatrix}$$

- 2^4 values of T^* gives $2^4 * 1036$ candidate values for 4 bytes of k^{11} .
- Repeating the attack with another pair of faulty and correct ciphertext gives almost 2 candidate values.
- Total 8 faulty ciphertexts required to retrieve all 16 bytes of k^{11} .

$$2 \cdot f' \oplus SNLF[2 \cdot f'] = S^{-1}[T_0 \oplus k_0^{11}] \oplus S^{-1}[T_0^* \oplus k_0^{11}]$$

$$1 \cdot f' \oplus SNLF[1 \cdot f'] = S^{-1}[T_{13} \oplus k_{13}^{11}] \oplus S^{-1}[T_{13}^* \oplus k_{13}^{11}]$$

$$1 \cdot f' \oplus SNLF[1 \cdot f'] = S^{-1}[T_{10} \oplus k_{10}^{11}] \oplus S^{-1}[T_{10}^* \oplus k_{10}^{11}]$$

$$3 \cdot f' \oplus SNLF[3 \cdot f'] = S^{-1}[T_7 \oplus k_7^{11}] \oplus S^{-1}[T_7^* \oplus k_7^{11}]$$

$A = 2f'$
 $B = f'$
 $C = f'$
 $D = 3f'$

swayam

So, what you will basically now do is you will have the T and the T star which are your 10th round outputs. And the rest of the attack is very simple. So, what you basically do is that now you basically just guess the 10th round key which is a quartet of the key and check for the expected relations in the 9th round output. So, remember that the 9th round output this was my disturbance where the disturbance was only in this column.

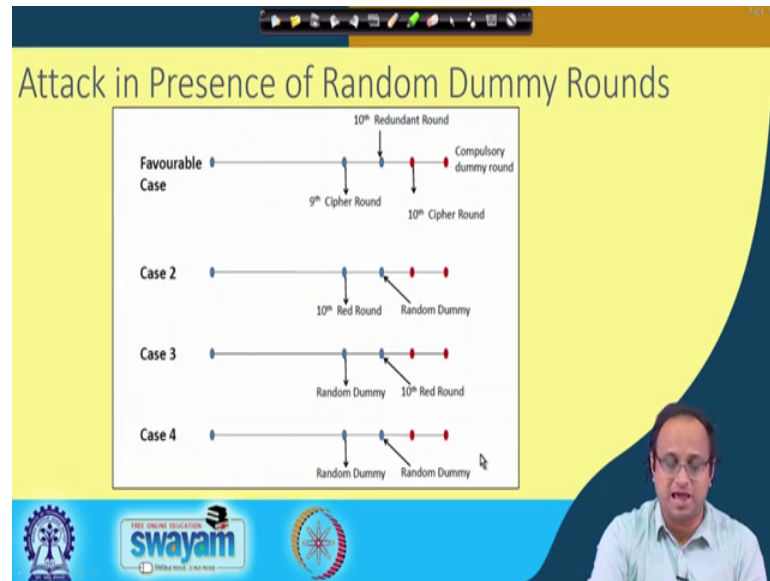
So, this right essentially is nothing $2 \cdot f'$ dash XOR with SNLF of $2 \cdot f'$ dash. And that you can again obtained by inverting S inverts and applying it on T_0 XOR with the 11th round k_0 ; that means, the last round k_0 , XOR with S inverts applied on T_0 star XOR with k_0^{11} . So, likewise you can do it for the other ones and therefore, you can again start making guesses of this k quartets and trying to see how many of them solves. So, this essentially quite similar to the classical DFA on AES that we have seen.

So now the complexity can be worked out like this. That is there are 2 to the power of 4 values of T star as I already said. And if I solve this equation right roughly 1036 candidate values for the 4 bytes of k^{11} will survive this test. And therefore, right there are 2 to the power of into 1024 candidates value because T star can also take 2 to the power of 4 values.

So, if you repeat this attack with another pair of faulty and correct cipher text then it gives at most 2 candidate values. So, therefore, you see that with one pair of faulty and correct cipher text we have been able to obtain 4 key bytes. So, therefore, if you want the

entire AES 128 key you need 8 faulty cipher texts ok, to retrieve all the 16 bytes of k 11. So, therefore, these attack shows that if there are no dummy rounds after the 9th round you know like; I mean after the 9th round cipher when you inflicted the fault, then you can do this attack in a very simple manner I mean you can do this attack in this using this technique.

(Refer Slide Time: 14:26)



But now, we will also see that we will relax this condition and see that the attack can be made toward even if there are random dummy rounds. For example, the favorable condition right is depicted in this diagram; that means, I am trying to induce the fault at the 9th round cipher follow. So, there can be you know like this possibilities there can be this possible cases.

So, what can happen is that there is a following 10th redundant round. There is a 10th cipher round and there is a final compulsory dummy round which is happening. Or it can also happen that when I am inducing the fault. The 9th cipher round has already taken place before and this is the 10th redundant round which is taking place.

And then right I mean, there is a random dummy round which is happening after this ok. And you know pretty much that is your you know like a essentially therefore, right this is your random dummy round and the random dummy rounds come comes after this. And that is followed by may be the 10th cipher round and the compulsory dummy round ok.

So, that this situation is perfectly possible. Or it can also be happening in this way that when I am inducing the fault, the fault happens in the random dummy round and after that the 10th redundant round gets executed. That is followed by the 10th cipher round and that is followed by the compulsory dummy round that is also possible.

Likewise, it is also possible then when I am inducing the fault right then the fault is in this random dummy round. Then again there is a random dummy round and that is followed by you know like 2 executions of the actual and the redundant cipher and of course, like there is a final compulsory dummy round. So; that means, right this essentially has to be the 10th cipher round.

Remember right that in this countermeasure since the final round is a compulsory dummy round. It is always that this particular round is a 10th cipher round. Because the 10th cipher round is always the penalty ultimate things. So, there is no ambiguity about this ok. So, there is essentially no confusion about the location of the 10th cipher round

(Refer Slide Time: 16:28)

The slide is titled "Identifying the Desired Faulty Ciphertexts". It contains the following mathematical expressions:

$$(T \oplus T^*)_{(4-(i+1))\%16} = (T \oplus T^*)_{(4-(i+1))\%16+1}$$

$$(T \oplus T^*)_{(4-(i+1))\%16+2} = 3 \cdot (T \oplus T^*)_{(4-(i+1))\%16}$$

$$(T \oplus T^*)_{(4-(i+3))\%16+2} = (T \oplus T^*)_{(4-(i+3))\%16+3}$$

$$(T \oplus T^*)_{(4-(i+3))\%16} = 3 \cdot (T \oplus T^*)_{(4-(i+3))\%16+2}$$

Below these equations is a matrix representation of the transformation $T \oplus T^*$:

$$T \oplus T^* = \begin{pmatrix} m_0 \oplus 2F_1 \oplus 1F_2 & \begin{matrix} 1F_3 \\ 1F_4 \\ 3F_5 \end{matrix} & \begin{matrix} 3F_4 \oplus 1F_5 \oplus 1F_6 \\ 2F_4 \oplus 3F_5 \oplus 1F_6 \\ m_2 \oplus 1F_4 \oplus 2F_5 \oplus 3F_6 \end{matrix} & \begin{matrix} 3F_7 \\ m_1 \oplus 2F_7 \\ 1F_7 \\ 1F_7 \end{matrix} \end{pmatrix}$$

The slide also features logos for Swamyam and other educational institutions at the bottom.

So, therefore, right the ideal condition of the favorable condition is this ok. And we can work out few things few interesting things. The first thing is that suppose I induce the fault and I want to know whether my fault that I have induced is good or bad ok. So, the first thing which I do is the I get this differential remember my if the fault was located, in the correct position then this is the differential that we already worked out, this is the nature of the differential. What does it say? It says that if I observe for example, these

positions like these differential position and these differential position, then right this and these differentials must be same. If they are not same then; that means, there is something wrong, I can discard. Likewise if you observe this right this and this differential must be same this must be twice these differential.

So, therefore, just by observing the differential I can understand that whether my fault induction was good or bad. If it was bad I can just remove those cases from my side. So, I can guide my fault attack in a much more intelligent manner.

(Refer Slide Time: 17:22)

Attack in Presence of Random Dummy Rounds

- Number of random dummy rounds : d
- Total number of rounds : $22 + d + 1$
- Target round of fault injection : $(22 + d - 2)^{\text{th}}$ RoundFunction.
- $(22 + d)^{\text{th}}$ RoundFunction: 10^{th} cipher round.
- \therefore The probability of $(22 + d - 2)^{\text{th}}$ RoundFunction being a 9^{th} cipher round: $\frac{(19+d)! / ((19)!(d)!)}{(21+d)! / ((21)!(d)!)}$
- If $d = 20$ then the probability that 40^{th} RoundFunction is a 9^{th} cipher round is nearly 0.26.

The slide also features a video inset of a man speaking in the bottom right corner and logos for Swamyam and other organizations at the bottom.

Now, we can also workout what is the probability that I am able to get that successful event ok. So, remember right that there are like the number of random dummy rounds which can be there d for example, this is important to kind of keep in mind that if the random dummy rounds are you know like increased just arbitrarily then your countermeasure will be showing inefficient that it nobody will adopt it. So, we can basically pretty much assume that the number of dummy rounds can be you know like be something which is practical ok. So, therefore, let that practical number be d , but this can happen in a any arbitrary order.

So; that means, right remember that there are already 22 rounds which are part of your, you know like actual cipher and redundant round, because that is 2 into 11 plus d dummy rounds plus a final compulsory dummy round ok. 1So, when I am saying random dummy rounds I am not including in the final compulsory dummy round. So, therefore, right now

my target position is basically $22 + d$ minus second round function because when I am inducing the fault this is my you know like position where I want to induce the fault and that essentially stands for $22 + d$ minus 2.

So, that is my position where I am say injecting the fault. So now, the question is right I mean also right keep in mind there are $22 + d$ th round is always your 10th cipher round. So, the probability that the $22 + d$ minus 2th round function is a 9th cipher round can we worked out easily as this equation ok.

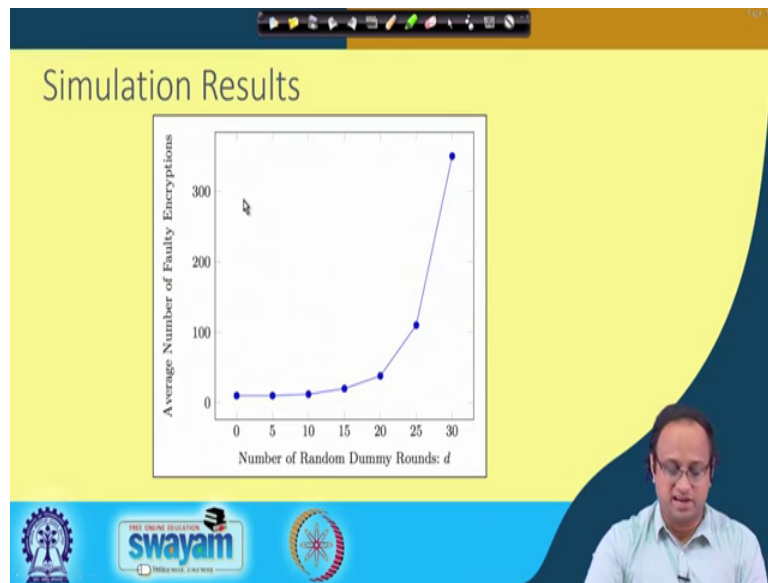
So, in the denominator you have all possible sequences. Remember there are 21, you know like 21 for example, you know like because you can exclude the final dummy round because that is nowhere coming into your you know like into your consideration as well as the final compulsory dummy round. So, you will leave out that then there 21 other rounds which are of consequence. Because it is $20 + 1$ redundant round; that means, the 10th redundant round.

So, there are 21 such rounds along with it there are d rounds which are dummy rounds which are randomly occurring. So, therefore, right if I consider any arrangement of them, then that will be $21 + d$ factorial and that I will divide of course, by 21 factorial and d factorial because they are you know like this is the by applying a simple arrangement argument basically.

So now likewise right if I consider in the numerator I am basically finding out the cases where I am more interested here; that means, right I am kind of assuming that, but other I am fixing the position ok. I am fixing the position which means that I am considering this case where this is my 9th cipher round. And therefore, right I mean I am basically align the others 2 kind of pretty much vary. And therefore, right this basically worked out something like $19 + d$ factorial divided by 19 factorial and d factorial.

And one can work out that if I put say d equal to 20 which means even if I allow you know like say double almost double the number of dummy rounds to occurred, then the probability of the fortieth round function is a 9th cipher round is nearly 0.26 ok. Which is quite high; that means, if I tries a 100 times then, 26 times I will be able to hit the correct position. And also keep in mind that you know you can guide your attack; that means, if there is a wrong case we were not using in your equations you just simply throw them because you know from the differential whether they are good or bad.

(Refer Slide Time: 20:49)



So, therefore, right a simulation results shown here; that means, if I increase the number of dummy rounds of course, the number of faulty encryptions we will increase, but at the same time even if d is 30 you still require around 300 encryptions or 300 faulty encryptions which is not really so, high to get the correct key. And remember the correct key means the entire correct key and this includes all the possible requirement of 8 faulty ciphers.

(Refer Slide Time: 21:16)

The last cipher round is always the penultimate round. The attacker can verify the target round using side channels. Merely, infecting all bytes in the output is not sufficient: the infection technique should result in a ciphertext which nullifies any effect to hypothesize the key. The countermeasure itself should not leak.

We use a random string called 'rstr', which is a 't' bit random string, where 't' is a constant. It consists of '2n' 1's corresponding to AES rounds, again out of which 'n' are redundant, and the rest cipher rounds. The remaining, ie. 't-2n' corresponds to dummy rounds. We use a Boolean function 'BLFN', which maps all 128 bit non-zero input to 1, and 0 to 0, ie. $BLFN(0)=0$

So therefore, right the flaws in the counter measured are can be summarized as so, therefore, the countermeasure is completely broken. And therefore, you know like they can shows the power of faults in context to krypton season, in context to developing secured cipher implementations. And what we see here is there are last cipher round is always the penultimate round which is a weakness in the countermeasure. The attacker can verify the target round using side channels. And merely infecting all bytes in the output is not sufficient. The infection technique should result in a ciphered which nullifies any effect to hypothesize the key. And the countermeasure itself should not leak.

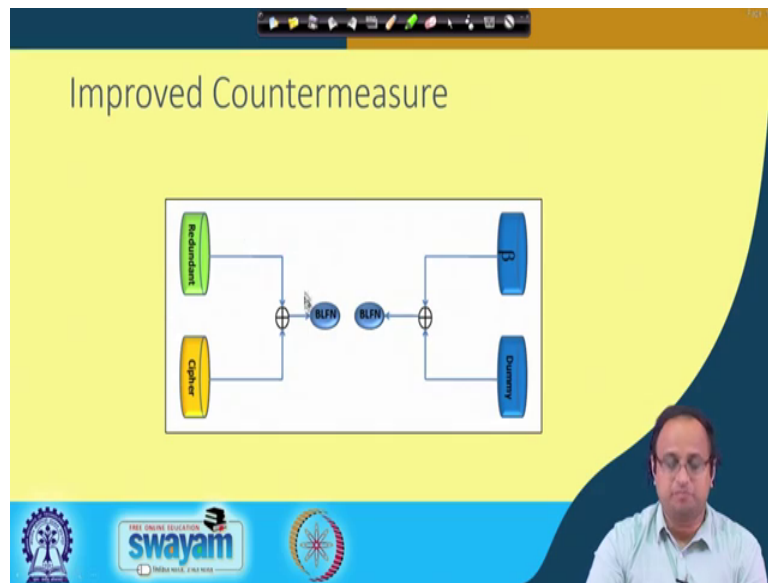
For example, in the previous case when we did the counter measure because of that measure of flow that we pointed out the counter measure itself was leaking. So now, we basically want to develop a better strategy for protections. So, what we do is that we now use a random string and call it as 'rstr', which is basically a 't' bit random string where 't' is a constant. So now, what we do is it basically we allow '2n' 1's and the 1's basically correspond to AES rounds, again out of which n are redundant ok.

So, we allow them you know like n redundant and n ciphers and that is why totally there are 2n you know like '2n' 1's which corresponds to AES round again out of which n are redundant and the rest are cipher rounds; that means, right there are t-T minus 2n positions which are essentially 0 and this stand for dummy rounds.

So now, what we do is we basically you know like allow we develop define a Boolean function and call this as 'BLFN'. BLFN which basically maps all 128 bit non-zero input to 1 and a 0 to 0; that means, a BNLf or BLFN applied on 0 will be equal to 0. So, this 0 will get mapped into 0, but for all other cases right it will map; that means, for all other 128 bit non-zero values this will result in 1 ok.

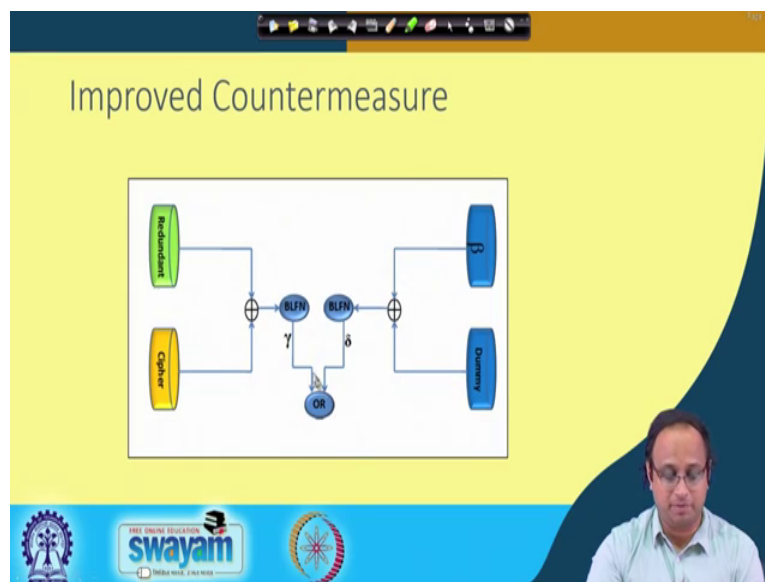
So, remember this is a Boolean function. As opposed to the previous SNLF function right which basically resulted in an you know like 8 bit value.

(Refer Slide Time: 23:27)



So now, here is a pictographic description of how the counter measure works. So, we basically apply this BLFN both at the XOR of the redundant and the cipher and the dummy and my reference beta.

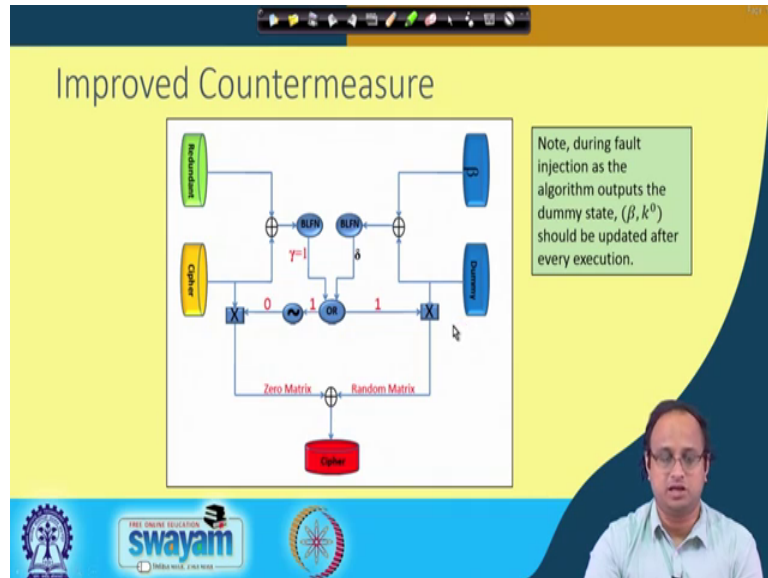
(Refer Slide Time: 23:45)



So now, this beta is a secret which I do not allow the attacker to know previously. So now what I do is basically right apply or you know like compute an odd function between the output of this BLFN function and the BLFN function for the XOR of the dummy and beta. Initialize again the dummy to beta; that means, if they; if I make it beta

then beta XOR with beta will be 0. And therefore, initially delta will be equal to 0 and for all situations when the redundant and the cipher at same this will also be 0 and therefore, gamma will also be 0 and therefore, 0 odd with 0 will also result in 0.

(Refer Slide Time: 24:17)



So, therefore, right this is the complete workout of the solution. So, therefore, here this particular value has been inverted and I will explain why and therefore, now you basically kind of pretty much multiply. So, this is a bitwise ending for example, which you do with the cipher. And in this other path you basically take this or output. So, this is the same output in both cases this or is basically again you know like applied and again you know like you do a bitwise ending with the dummy output and then you know like you XOR this results and generate the final cipher.

So, what does it imply? So, suppose you know like that gamma is equal to 0 and delta is equal to 0 which means it is a normal operation. Then this 0 and odd with 0 will result in 0 therefore, this will get inverted this will become 1. So, the cipher will pass whereas, if this is 0 this will get you know like ended with 0 and therefore, the dummy will become 0 and therefore, you will get the original cipher.

On the other hand, right if there is a fault for example, in this case the fault is suppose here in somewhere in the redundant or in the cipher, this will result in a non 0 value here which will get converted into gamma which is equal to 1 and therefore, this will becoming 0 and therefore, this cipher will not be passed.

So, what we will get, basically get passed is a random matrix. And therefore, the attacker will see this random matrix the same thing we will hold you know like if there is a fault in dummy. For example, then also this will result in the one and therefore, this again will be a random matrix. Note that during the fault injection as the output or the as a algorithm outputs the dummy state beta comma k 0. So, I mean dummy state is outputted; that means, beta and the corresponding k 0 must be updated after every execution ok.

So, therefore, we assume that this infective counter measure should update the value of beta after every execution.

(Refer Slide Time: 26:06)

The slide is titled "Improved Countermeasure" and features a list of four bullet points:

- 1. Fault injection in any of the cipher, redundant or dummy round \implies **Every** byte in the resulting ciphertext is infected with a different value.
- 2. The resulting infected faulty ciphertext is completely random.
- 3. More than one random dummy round after the last cipher round.
- 4. The improved countermeasure protects both SPN ciphers and Feistel ciphers.

Below the list, a text box states: "The modified infective countermeasure is secured in the random fault model. However, it is still vulnerable under a specific fault model called instruction skip model, which is beyond the scope of this work."

At the bottom of the slide, the authors are listed: "Sikhar Patranabis, Abhishek Chakraborty, Debdeep Mukhopadhyay: Fault Tolerant Infective Countermeasure for AES. J. Hardware and Systems Security 1(1): 3-17 (2017)".

The slide also includes logos for "swayam" and "INDIA WISE, LIFE WISE" and a video inset of a man speaking.

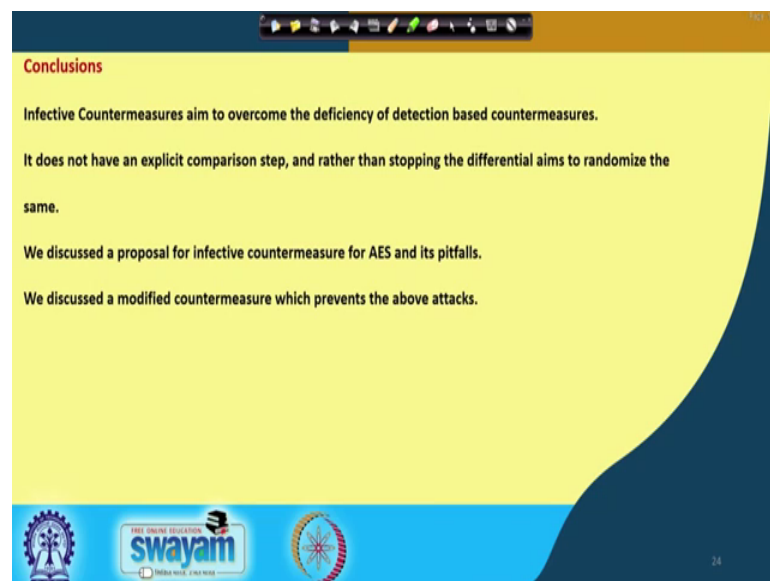
So, therefore, right in this case the fault injection in any of the cipher redundant or dummy round, we will result in every byte in the resulting cipher text is infected with a different value. Because every time it would be a random output which is being generated completely uncorrelated about the secret key.

The resulting infected faulty cipher text is completely random and more than one random dummy round you know like can happen after the last cipher round. Because remember that we had this rstr bit sequence. And it can pretty much happened that the cipher round or the final cipher round happens somewhere else after which you have got a sequence of dummy rounds ok. So, therefore, right more than one random dummy round can happen after the last cipher round.

So, the improved countermeasure protects not only against SPN, but also can be adopted for Feistel kind of ciphers. So, the modified infective countermeasure is therefore, claim to be secured in the random fault model at least the attacks that we have seen previously will not work; however, it can still be vulnerable against a specific fault model which we are not keeping in the scope of our discussion which is called as instructions keep model. So, there could be sudden fault models where because of the fault and particular instruction can get skipped or even an instruction can get converted into some other instruction ok.

However, this is beyond the scope of this work. And if you are interested then you can have a look at this particular reference where we have more discussions on this infective countermeasure and how it can be made vulnerable against and instructions keep attack. And as well as how it can be made secured against even against such kind of threats.

(Refer Slide Time: 27:39)



So, to conclude infective countermeasures aim to overcome the deficiency of detection based countermeasures. It does not have an explicit comparison state as we have seen in the detection counter measures. And rather than it does not stop the differential aim, it basically aims to randomize the output and make it kind of infected or uncorrelated with the secret key. We discussed a proposal for infective countermeasure for AES and also discussed about it is pitfalls. And finally, we discussed about a modified countermeasure

which essentially is more robust and which essentially can prevent the attacks at least that we have seen in the constant fault model and also in the random fault model.

So, this brings us to the end of this class and thanks for your attention.