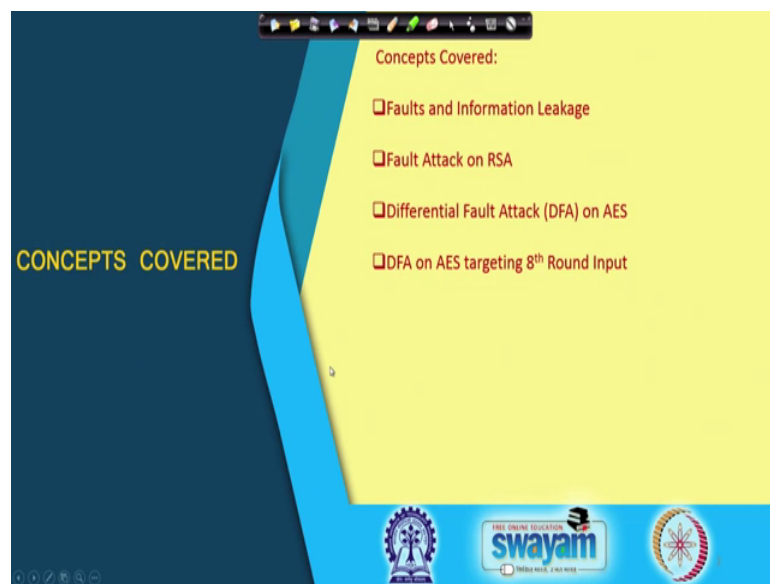


Hardware Security
Prof. Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 46
Fault Analysis of Cryptosystems

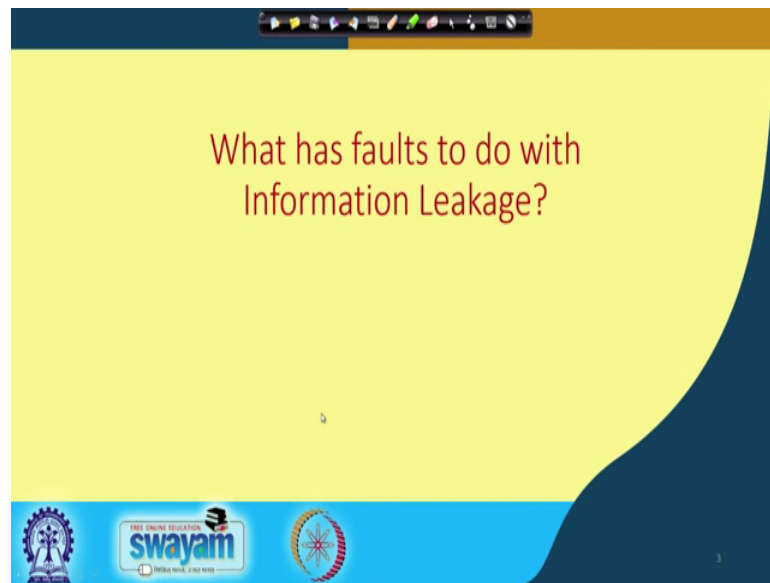
So, welcome to this class on Hardware Security. So, we have been studying about side chain attacks and in particular we have been discussing about power attacks.

(Refer Slide Time: 00:33)



So, in today's class we shall be trying to see a new form of side channel analysis which is called as fault attacks or Fault Analysis. We shall start with trying to understand what is the relationship between faults and information leakage, I will be discussing about a classic fault attack on the RSA cryptosystem, and then we shall be trying to look into how to perform a particular fault type of fault attack which is called as differential fault attack, which is basically a combination of fault attacks with differential cryptanalysis and try to apply it on AES. And then we shall be trying to look into DFA on AES targeting specifically the 8th round input of AES 128.

(Refer Slide Time: 01:07)



So, first of all what has got faults to do with information leakage? So, let us try with a very high level description of this with very you know like an example very simple example.

(Refer Slide Time: 01:15)



So, imagine that there is there is a dealer jack. So, this particular example has been taken from the sorcerer's apprentice guide to fault attacks which was shown in FDTC 2006 which is a workshop on fault attacks co held with chess and this particular example

shows that there is a dealer jack who is a supplier of cards and planes and wants to essentially sell it to a customer who is called say Dino.

So, the idea is that you can see that these are toy cars and toy planes. So, therefore, each car costs around dollar 3 and suppose each plane costs around dollar 5. So, jack asks Dino how will you pay? So, Dino says that he will say dollar 15 by postal order and immediately you can understand that there are 2 possible transactions. One it can be that there are 5 cars which are being procured or it can be that there are 3 planes which are being procured.

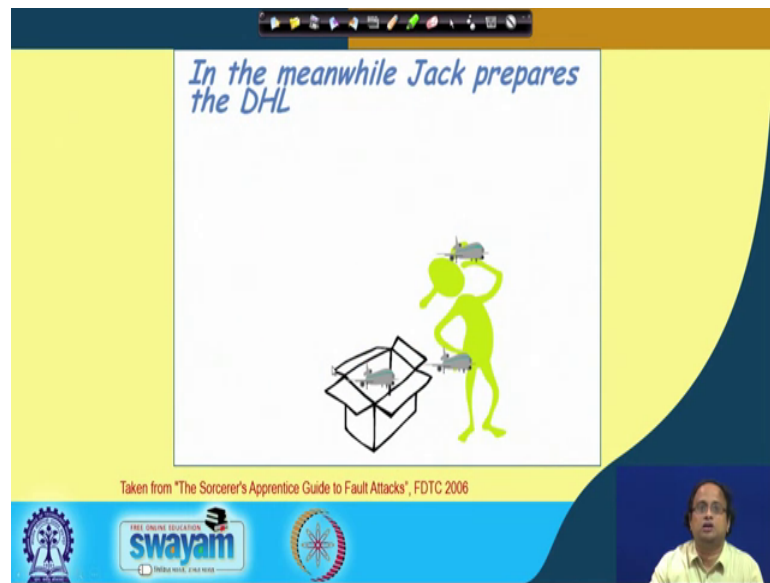
(Refer Slide Time: 02:23)



So, the question is right essentially it can be any of these transaction and of course, like you know like there is a malicious imagine that there is a malicious postman, who is key to know which of these transactions has taken place.

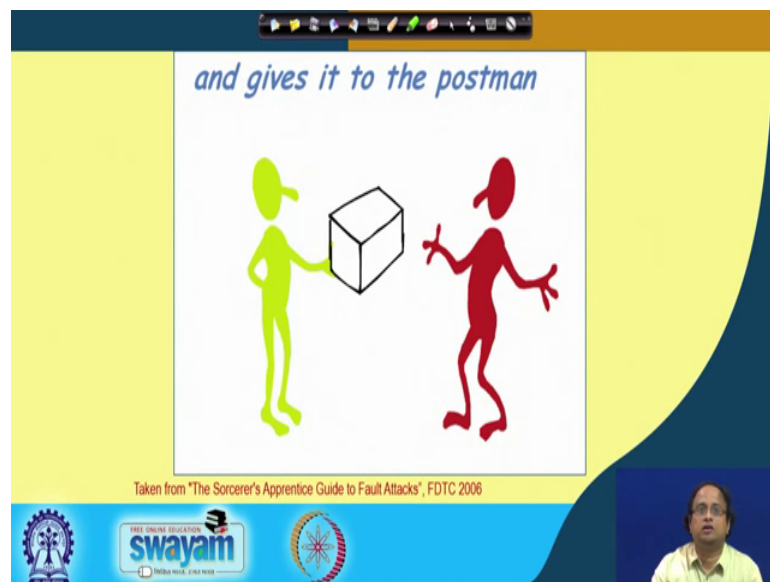
So, information theoretically it means that, there is a one bit of uncertainty I want to know whether it is transaction 0 or transaction 1.

(Refer Slide Time: 02:41)



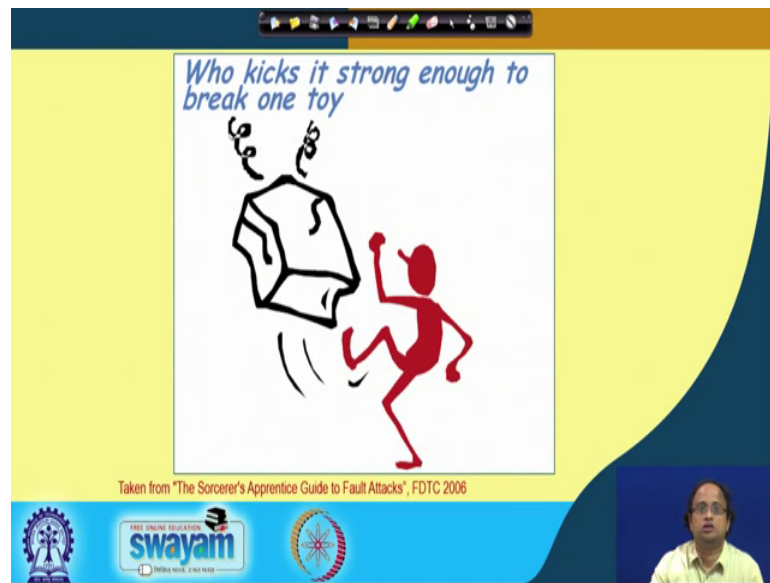
So, therefore, right meanwhile the Jack prepared the package and sends it or prepared it to courier it to the customer Dino and therefore, right it gives it to the postman.

(Refer Slide Time: 02:51)



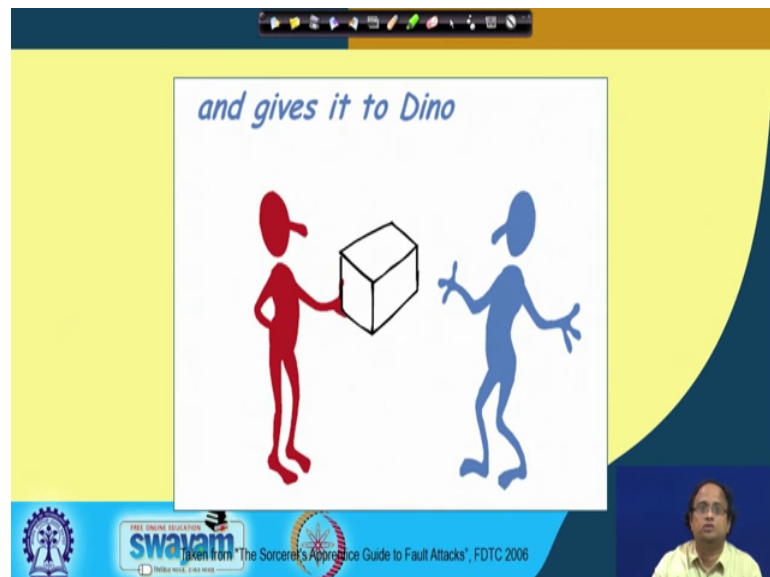
Now, this postman is malicious and curious.

(Refer Slide Time: 02:57)



So, therefore, right he does this. So, he basically kind of applies a fault or kicks it strong enough to break exactly one toy in the parcel. So, the fact that you know like he is able to break one toy can be called what is called as fault control ability ok. So, suppose he controls a fault so, that only one single toy has been broken.

(Refer Slide Time: 03:21)



So, therefore, right when he gives it to Dino and imagine that suppose there has been of course, like you know like that when we do online transactions, it is kind of assumed that if there is a broken item which you received to us, then we do not pay for it.

(Refer Slide Time: 03:39)

a week later he monitors Dino's postal order...

= $4 \times 3 = \$12$

= $2 \times 5 = \$10$

Lesson learned: Fault attacks can also extract secrets from tokens!

Hardware faults can have various sources:
voltage glitches, light beams, laser beams...

Taken from "The Sorcerer's Apprentice Guide to Fault Attacks", FDTC 2006

swayam

So, therefore, based upon that right of course you can understand since one of the toys has been broken. Therefore, Dino either pays dollar 12 or dollar 10 ok. And therefore, the moment the postman is able to see like which kind of payment the postman the customer does, he is able to understand which of these two transactions took place.

So, this is a very simple example, but kind of shows us that faults essentially can be used to extract secrets ok. And this has been a widely researched topic and there has been several attacks which people have developed on crypto systems, in particular right the fact I mean the fact that one can easily or you know like I would say because of the development of fault injection techniques, and the fact that you can actually inject faults with very high preciseness and also at low cost has given more impetus to this discourse.

So, hardware faults can have various sources for example, it can be induced by voltage glitches, by light beams by laser beams and so on and many other ways ok. You can probably envisage that if the system gets more and more complex, then the chances of having faults is also increased ok. And what we are studying here is that fault tolerance is extremely important in the context of cryptography because not only the fact that, you do a faulty execution, but you also divulge your secret key and therefore, it is of paramount importance.

(Refer Slide Time: 05:05)

Fault Attacks on RSA

- Only decryption is subject to attacks
- Assume:
 1. Attacker can flip a single bit in key d
 2. S and corresponding message M known to attacker

Decryption device generates \hat{M} satisfying

$$\frac{\hat{M}}{M} = \frac{S^{2^{d_i}}}{S^{2^d}} \pmod{N}$$

- If $d_i = 0$ then $\hat{M}/M = S^2 \pmod{N}$ ✓
- If $d_i = 1$ then $\hat{M}/M = 1/S^2 \pmod{N}$ ✓

Source: Koren and Krishna, Morgan-Kaufman 2007

So, let us try to you know like try to relook or look back at the RSA cryptosystem particularly with in the context of faults. So, imagine that there is a decryption and you know that in public key cryptography, when we do decryption we use the secret key. So, imagine that there is an attack, there is and of crypto system which is doing decryption and this is essentially subjected to a fault attack. So, the attacker of course, like in this kind of attacks we have to state something which is called as a fault model which basically states what is the power of the adversary.

So, here we assume for example, that the attacker can flip a single bit in key d and based upon this assumption what happens is as follows. So, therefore, imagine that you know the cipher that is obtained by encrypting a message M using RSA is say S ok; that means, I am using I am basically right doing a simple encryption using RSA and it works as follows. We basically take for example, the message M ; the message M is your message and you perform an encryption.

So, that is power of e and you do a modulo with a public component N and you obtain the cipher which is denoted S . So, this is equivalent to saying that I take S and I raise it to a secret d and I do a modulo N and I get back the original message M . So, now, imagine that there is you know like a crypto system which is doing this operation and there is a secret exponent inside it suppose imagine it is a hardware device and it is doing this computation that is S power of d modulo N to get you the plaintext.

So, now, the attacker injects a fault and imagine that it is able to key create a fault in the i 'th bit and therefore, right because of the fault it gets a faulty decryption done ok. So, imagine that in one case it gets a correct decryption done, in another operation it gets a faulty operation out. So, you can actually write pretty much this d as a binary exponent. For example, from d_{n-1} to say d_0 where your i th bit is denoted as d_i ok; imagine that this fault is able to flip d_i ok.

So, you can say that you know like how can I flip d_i . So, you know like in reality actually there may be several ways in which I can flip a particular bit ok. For example, you know like when there is a circuit which is in operation I can take for example, I can suddenly make a voltage fluctuation or I can suddenly change the clock of my supply ok, I can probably take an electromagnetic gun and shoot and that can lead to perturbations which can flip you know like randomly bits and if there is a probability and actually it can be a significant probability that a single bit gets changed or flipped.

So, there can be various ways of doing that and right now we are assuming that suppose you know like somebody is able to make a flip in this way so, what is the ramification of that. What is the implication of that on this on security of say RSA kind of algorithm. So, we know that if this has happened then if we just look and kind of elaborate this operation right you can easily observe that, what we are basically doing is this. We are getting M and M is s power of we know that this is nothing, but d_{n-1} till d_0 ok.

So, therefore, this is written in binary format and that is equal to nothing, but S to the power of $2^{d_{n-1} + 2^{d_{n-2}} + \dots + 2^{d_0}}$ ok. And now imagine that somebody is able to flip this d_i and therefore, d_i becomes \hat{d}_i because of the flip. So, d_i becomes say \hat{d}_i , it gets complemented 0 becomes 1 or a 1 becomes 0.

And therefore, right rather than getting M the attacker essentially gets S which is or S or its \hat{M} which is nothing, but S to the power of $2^{d_{n-1} + 2^{d_{n-2}} + \dots + 2^{\hat{d}_i} + \dots + 2^{d_0}}$. So, everything remains same except the i 'th bit becomes \hat{d}_i or \bar{d}_i plus so, on till d_0 . So, therefore, now if I take a ratio between them; that means, I take a ratio of M and \hat{M} , then everything cancels out except we get $2^{d_i - \hat{d}_i}$ ok. So, what does that imply?

So, that is essentially written over here and. So, what is written over here is just a reciprocal of that that is $M \hat{=} M^s \cdot 2^i \cdot d_i \hat{=} d_i \cdot 2^i$ or $d_i \hat{=} d_i \cdot 2^i \cdot M^{-s}$. So, now, there can be two possibilities d_i can be either 0 or d_i can be either. So, note that the attacker can easily compute this part that is the left hand part, the attacker also knows the value of S because it knows the cipher and therefore, it can check whether this equation is satisfied or whether this equation is satisfied. Note that if d_i is equal to 0, then I get you know like if d_i is equal to 0 then; that means, $d_i \hat{=} 1$. So, therefore, that means, right if d_i is equal to 0 you can observe that $d_i \hat{=} d_i \cdot 2^i \cdot M^{-s}$ is equal to $1 \cdot 2^i \cdot M^{-s}$ which is equal to 1 .

On the other hand, if d_i is 1 then this implies that this Δd_i will be equal to 0 minus it will be equal to 0 minus 1 and therefore, will be equal to minus 1 ok. So, therefore, in one case you will have S to the power 2 to the power of i or in another case you will have s to the power 2 to the power of minus 1 that is 1 by S to the power 2 to the power of i . And since the attacker already knows the left hand side and the right hand side it can easily check which of these 2 equations are satisfied and from there can get the information about whether d_i was 0 or whether d_i was 1. So, imagine that if the attacker is able to do this right on every bit, then it can basically on a linear using a linear time complexity algorithm, can easily retrieve the entire exponent without solving any complicated hard problem ok.

So, what it shows is that and you can also imagine that you know if even if I assume that the attacker does not know precisely which bit is flipped right he can basically very much you know like make a guess that suppose you know like the i 'th bit is guess i 'th bit is i th bit is flipped. And therefore, can check whether which of the solutions give rise give rise to a correct check correct equation and the probability that you will probably you know like get the correct value of i also determined is also significantly high using that. So, therefore, right I mean basically right I mean what it shows or what this simple exercise shows is that faults can be quite catastrophic to security ok. So, therefore, we need to properly take care of this kind of attack vectors.

(Refer Slide Time: 11:55)

Fault Attacks on RSA

- Assume that the attacker flips randomly a bit in d .
- Example: $(e, N) = (7, 77)$, $d = 43$ $d_5 d_4 d_3 d_2 d_1 d_0 = 101011_2$
 - Ciphertext = 37 producing $M = 9$ if no fault is injected and $\hat{M} = 67$ if a fault is injected
 - Search for i such that $9 = (67 \cdot 37^{2^i}) \bmod 77$ $i = 3 (d_3 = 1)$ since

$$(67 \cdot 37^8) \bmod 77 = (67 \cdot 53) \bmod 77 = 9$$

Source : Koren and Krishna, Morgan-Kaufman 2007

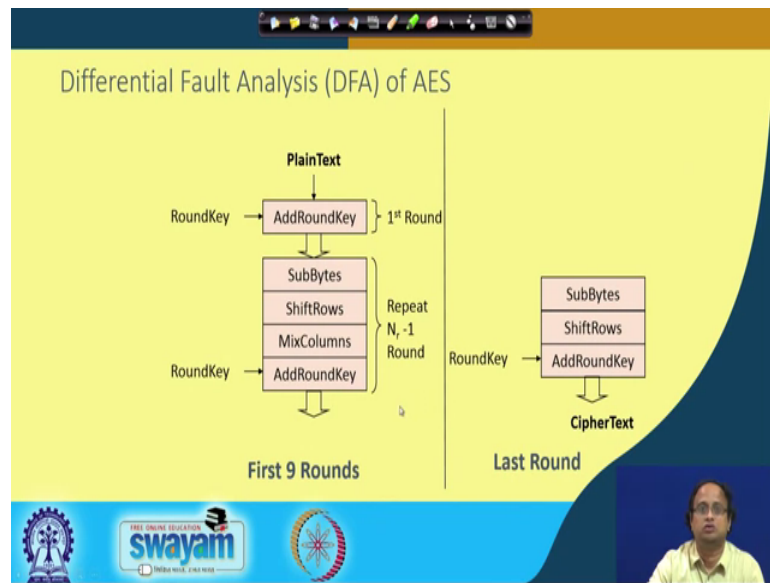
swayam

So, here it is a simple exercise or example to show that, and here is you know like e has been taken as at the RSA. So, e is 7 n is suppose 77 which is like a product of 7 and 11 two prime numbers and suppose your exponent is d equal to 43 and therefore, the corresponding key bit sequence is shown here. So, the cipher text is say 37 which is produced which is producing M equal to 9 if there is no fault which is injected and suppose you know like there is an i 'th bit which has been flipped and you get six seven as your faulty output.

So, even if I assume that I do not know the value of i will basically search for the value of i and I know that I can take any of these six possible values and therefore, I will just keep on guessing and I will find that for some case in this case i equal to 3, I will find that this equation is satisfied and as this equation is satisfied this reveals a value that d_3 is equal to 1. So, therefore, I can retrieve the value of the secret bit in this manner by just flipping that corresponding bit.

So, this example shows that faults can be applied for cryptographic systems and was essentially the basis of a very seminal paper written by Dan Boneh in 1997 and that has led to a flurry of works on applying fault attacks for various kinds of crypto systems.

(Refer Slide Time: 13:11)



In particular, we shall be trying to look into a DFA on AES, because AES is as we know the worldwide de facto standards on block ciphers. So, here is a quick recapitulation of what AES does we have already seen this. In particular we are talking about AES 128, but that is without loss of generality; it can be pretty much applied to other variants of a AES and also other kinds of block ciphers. So, of course, you have to you know like fine tune it to that algorithm.

So, therefore, if I take a plain text and if I do an add round key and then I do n you know like nine rounds pretty much with sub byte shift rows mix columns and add round key as the round constituents and then in the final last round I just do a SubByte ShiftRow and add RoundKey; I do not do any mix columns in the final last round.

(Refer Slide Time: 13:55)

Effect of Error on AES

- Plaintext:
32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
- 128-bit key:
2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
- Ciphertext:
39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

- A single error in the plaintext:
30 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
- Results in the ciphertext:
c0 06 27 d1 8b d9 e1 19 d5 17 6d bc ba 73 37 c1
- A single error in the key:
2a 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
- Results in the ciphertext:
c4 61 97 9e e4 4d e9 7a ba 52 34 8b 39 9d 7f 84

• A single-bit error results in a totally scrambled output

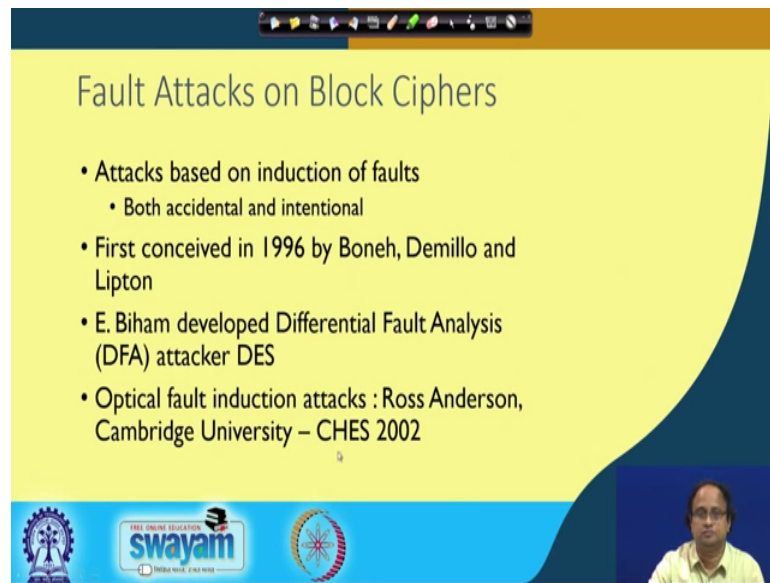
Source : Koren and Krishna, Morgan-Kaufman 2007

swayam

So, if I do this right I know that I am just let us just quickly study about if the effect of error on AES. So, this shows an example where there is a plaintext 128 bit and this is the corresponding ciphertext and you see that, if there is a single error in the plaintext for example, here I have it was 2 when I made it 0. This leads to a you know like at a completely random output you can see this ciphertext and this ciphertext has got no correlation almost right it is kind of completely random.

So, likewise if there is single bit error in the key also that leads to a random ciphertext and therefore, right it shows that it leads to a completely scrambled output. So, therefore, AES is a you know like a kind of a wonderful candidate for a pseudo random function in that case and therefore, we see that you know like even if there is a small bit error, then there is a wide diffusion in which we obtain in the. So, we get a very nice diffusion and confusion if there is a flip in the key or the flip in the plaintext and the output is kind of looks quite random.

(Refer Slide Time: 14:53)



The slide is titled "Fault Attacks on Block Ciphers" and features a yellow background with a blue wave-like shape on the right side. At the top, there is a navigation bar with various icons. The main content consists of a bulleted list:

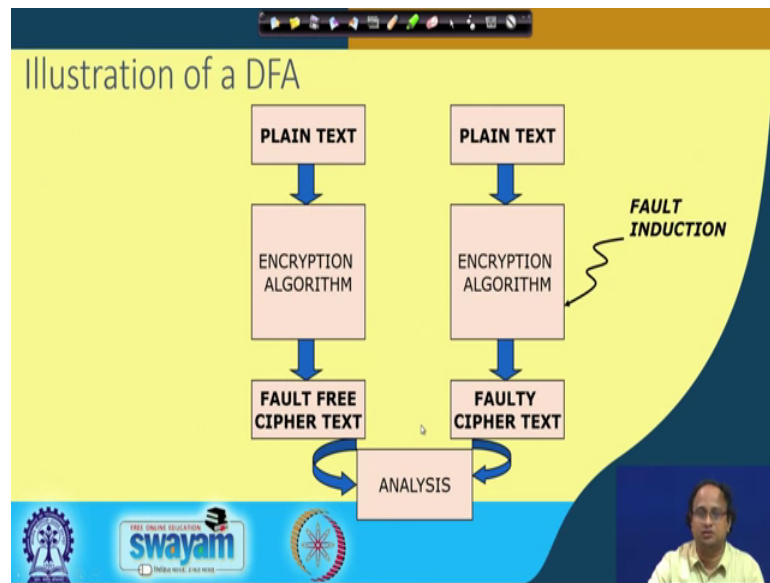
- Attacks based on induction of faults
 - Both accidental and intentional
- First conceived in 1996 by Boneh, Demillo and Lipton
- E. Biham developed Differential Fault Analysis (DFA) attacker DES
- Optical fault induction attacks : Ross Anderson, Cambridge University – CHES 2002

At the bottom of the slide, there are three logos: the Swamyam logo (Free Online Education), the logo of Anna University, and the logo of the Indian Institute of Technology (IIT) Madras. A small video inset of a man in a yellow shirt is visible in the bottom right corner.

So, in particular we shall be trying to look on fault attacks on block ciphers just to look see you know. So, therefore, we are considering the attacks based on induction of faults, note that faults can be intentional, but can also be accidental ok. So, what we are studying is that, the fault essentially can actually not only do a faulty computation. So, it is not that you just redo the computation, but it may happen that just because of the fault you have actually leaked your key and therefore, you know you actually need to change your key because your key is already compromised.

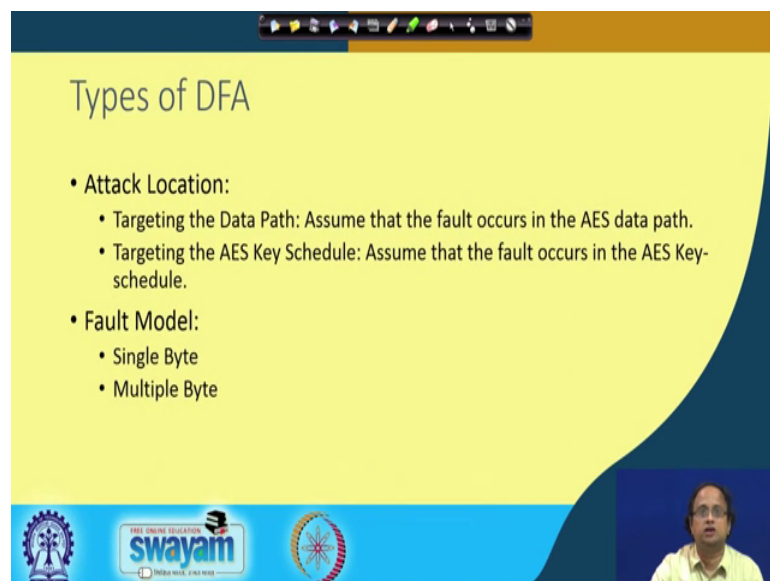
So, it was first conceived in 1996 by Boneh Demillo and Lipton as I said by its by the seminal paper to show that how cal faults can be catastrophic on cryptographic protocols. And later on there was a very nice work by Le Biham and Adi Shamir to show how differential fault analysis can be performed on DES and there was a paper in CHES in 2002, to show how optical fault inductions can be performed by a group by (Refer Time: 15:48) and Ross Anderson from Cambridge university to show that faults can be actually inflicted on cryptosystems with a very quite high precision at a reasonable amount of cost using laser beams or using optical techniques.

(Refer Slide Time: 16:03)



So, let us try to see how differential fault attacks work. So, in differential fault attacks you have got a plaintext, you get a fault free cipher text then you basically take the same plain text and you apply the encryption algorithm and now you basically do a fault induction. So, you get a faulty cipher text. So, differential fault analysis tries to combine the faulty fault free cipher text and the faulty cipher text, and then performs a differential analysis on that ok. And that is the basis of what DFA stands for or differential fault attacks.

(Refer Slide Time: 16:33)



So, there can be different types of DFA for example, you can target the data path for example, you can assume that the fault occurs in the AES data path when we are talking about AES; you can also target the AES key schedule ok. So, for example, you can assume that the fault occurs in the AES key scheduling algorithm and that also leads to a different variants of fault attacks.

You can also you know like I mean another very important characteristic when you are defining a fault attack is what is your fault model; that means, what is your assumption behind fault attacks what kind of faults are you assuming. So, two important kinds of faults is what I have written here, but there can be other different types of fault models. In particular I have talked about two random fault models; that means, I do not assume what is the value of the fault, I just assume that the fault is localized in a single byte when we are talking about AES. We know that AES state matrix can be conceptualized as for example, 16 bytes I am assuming that a fault is occurring in one of the bytes and it can take pretty much any random value ok.

In another case, I will assume later on that suppose a fault spreads across multiple bytes and therefore, right we will talk about that as a multi byte default model. So, on the passing I would like to mention here that there can be other kinds of fault models. For example; there can be a single bit fault model, where I assume just like as we have seen in the case of RSA the fault is in 1 bit. There can also be other kinds of fault models for example, stuck at 1 or stuck at 0 fault models; that means, you know like I am assuming that the fault is you know like stuck to 1 or 0. There could be other fault models also which can be utilized for example, I can assume a biased fault model, where I assume that the fault is not random, but it is probably biased towards certain values we will see the southern flavor of that in one of the subsequent talks.

So, right now here I am assuming random fault models; that means, I am assuming pretty much the fault can take any random value any arbitrary value.

(Refer Slide Time: 18:25)

The slide is titled "Single Byte Faults in known DFAs". It contains two main bullet points. The first is "Single Byte Fault", which includes two sub-bullets: "Attacker induces fault at the input of the 8th round in a single byte" and "Fault value should be non-zero but can be arbitrary". The second main bullet point is "Relaxing the requirements make the attack more practical", which includes three sub-bullets: "No knowledge required of the fault value", "Lesser bytes needed to be faulty", and "Lesser faulty cipher texts required". At the bottom of the slide, there are logos for "swayam" (Free Online Education) and "INDIA WISE, LEAD WISE". A small video inset in the bottom right corner shows a man speaking.

So, therefore, right we will be talking about in particularly a single byte faults that is attacker is inducing fault to the input of the 8th round in a single bite of a AES, and the fault should be non zero because if the fault is 0, then there is no fault. So, we will just assume the fault is non-zero, but it can take any value you can take any arbitrary any value. So, I will be relaxing the requirements, I will try to relax the requirements means because that makes a lot more practical, and you know like and fault attacks can be actually practical.

For example, right I mean what I will be trying to develop is develop a DFA where there is no knowledge requirement of the fault value and I will be or we try to develop at fault attack, which actually requires lesser bytes to be faulty and in particular, we will try we will kind of pursue a fault to see that whether the fault attack can be done with lesser and lesser number of faulty cipher texts. So, that you know like a single fault or maybe a double fault is able to retrieve the key only lift the key.

Because remember right when you are actually doing an attack on a real life world, you do not exactly have a control on the fault. So, a pretty much tie several for the injections and maybe luckily one of them faults in your desired fault model ok. So, therefore, it is important that you develop a DFA which requires lesser number of faults, because then the probability of getting such a fault is enhanced ok.

(Refer Slide Time: 19:47)

The slide is titled "State of the Art: DFA in DataPath (AES-128)". It lists five research papers:

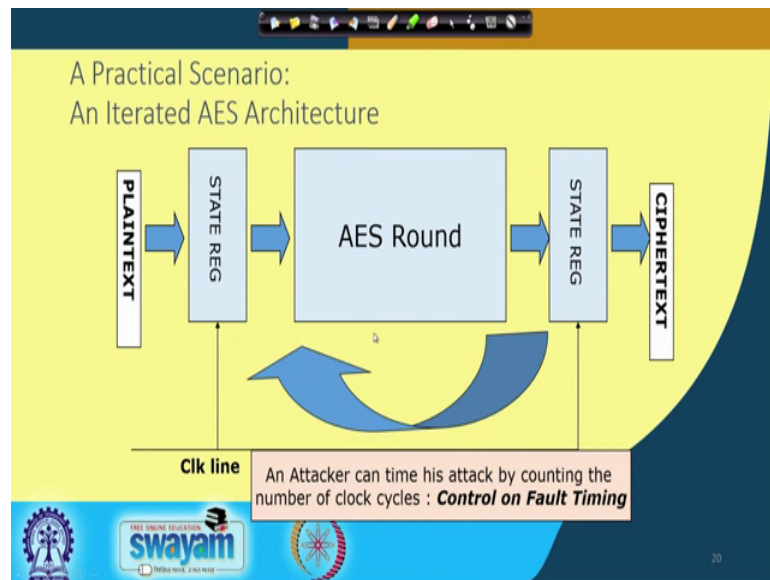
- **Piret et. al 2003 (CHES):** 2 faults for unique key, Time Complexity: 2^{40}
- **Mukhopadhyay 2009 (Africacrypt):** 2 faults for unique key, Time Complexity: 2^{32} ; showed attack possible with 1 fault.
- **Tunstall, Mukhopadhyay, Ali 2011 (eprint, WISTP):** 1 fault, key space: 2^8 , Time Complexity: 2^{32}
- **Ali, Mukhopadhyay 2011 (eprint):** Time Complexity: 2^{30}
- *Subidh Ali, Debdeep Mukhopadhyay, Michael Tunstall: Differential fault analysis of AES: towards reaching its limits. J. Cryptographic Engineering 3(2): 73-97 (2013)*

The slide also features logos for Swamyam and a small video inset of a man in the bottom right corner.

So, here is a brief history about how DFA on AES 128 evolved. For example, there were the people in 2003 in CHES by period and their collaborators to do a do fault attack with 2 faults for AES with the time complexity of around 2 power of 40, later on there was a work which we probably try to do on in Africa crypt try to do a fault attacks for you know like.

For unique key with the time complexity of 2 power of 32 we showed that the attack was possible even with one fault and later on it was further enhanced to do and reduce the key size to only 2 power of 8 values with the time complexity of 2 power of 32, which was further you know like enhanced later on to reduce to 2 power of 30. So, all these works have been summarized in this journal paper which is shown or published in journal of cryptographic engineering published in 2013.

(Refer Slide Time: 20:37)

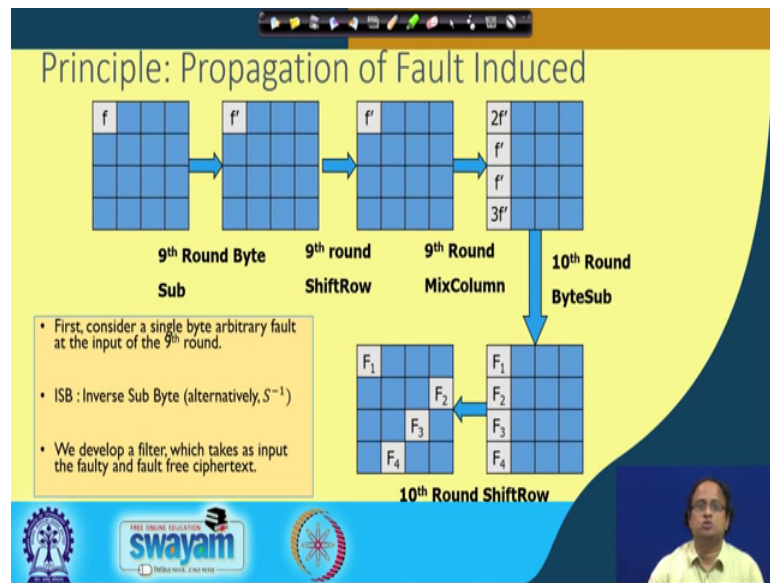


So, let us try to see you know like the context in which we are discussing the fault attacks. So, here you have got an internet architecture for fault for AES. So, pretty much; that means, and it is a very common architecture because you know like its a very popular architecture where we take AES round; that means, the AES round has been implemented on hardware and we are pretty much iterating the AES round 10 times just keep in mind that the last round does not have a mixed column.

So, we have already seen some flavor of such architectures previously. So, now, we are considering the clock line. For example, and we are trying to create a setup where suppose I create suddenly a fluctuation in the clock line where rather than you know like injecting the normal clock input, we suddenly make if you know like inject a fast clock and the objective is to create setup time violations in the circuit which will be manifested as faults ok.

Now, in other kind of setups we may probably try to do such a fault injection means by attacking the voltage line, where you create a sudden fluctuation on the voltage path. So, there are practical attacks also you know like which we will have mounted by maybe taking electromagnetic guns or maybe laser guns, but without loss of generality right let us assume that is a very simple demonstration or simple scenario, where we are trying to kind of inject the fault by creating you know like sudden clock injections in the clock path.

(Refer Slide Time: 21:59)



So, now let us try to see how the fault propagates. In particular just to understand the principle let us consider AES 128 where the fault as we induced on the input of the 9th round.

So, suppose imagine that the fault takes any random value. So, let us denote it by f which basically means it is any random value, it is any arbitrary value. So, remember that the byte sub is a bijective mapping the sbox in AES is bijective. So, therefore, it transforms f into some value say f' and then there is a shift. So, remember that if the fault is in the first row then there is no effect and the if it is in other rows, then you just shift to some other byte location ok, but that does not create a problem.

Because after this right the mixcolumns comes into play and if you remember right the mixed color matrix right then this is how the fault spreads. So, now, you can actually observe this diffusion from information theoretic point of view also ok. So, here for example, if I assume that I do not know anything about the value of the fault, then this essentially means that there is still 8 bits of entropy this byte can take any possible value.

So, now if you consider if you go ahead write like f' f' then essentially there is no you know drastic deduction of entropy, but if you come to this column for example, you suddenly see that if I fix the value of f' then the entire column gets fixed. So, now, right; that means, that if for example, if you do not do a fault right or do not make any assumption on the fault, then this byte can take any 32 or any 2 power of 32 values,

but suddenly now it can because of the fault it keeps constraint and it can take only 2 power of 8 values which means that there is a reduction on entropy, and that is you know like we will try to develop into a key retrieval process ok.

So, going ahead right this is how the byte sub will basically again convert it into some values like F 1 F 2 F 3 F 4 and then write because of the shift rows in the last round you will have F 1 F 2 F 3 and F 4 and remember these are the differentials because this is the xor between the correct operation and the faulty operation ok. So, now, we basically you know like get this value of F 1 F 2 F 3 F 4 because we get the ciphertext, we get the faulty ciphertext. If we exert them, then we get the value of F 1 F 2 F 3 and F 4.

(Refer Slide Time: 24:21)

The Patterns Gives the Following Equations

- $ISB(x_1+K_1)+ISB(x_1+F_1+K_1)=2[ISB(x_2+K_2)+ISB(x_2+F_2+K_2)]$
- $ISB(x_2+K_2)+ISB(x_2+F_2+K_2)=ISB(x_3+K_3)+ISB(x_3+F_3+K_3)$
- $ISB(x_4+K_4)+ISB(x_4+F_4+K_4)=3[ISB(x_2+K_2)+ISB(x_2+F_2+K_2)]$

So, now the question is with F 1 F 2 F 3 F 4 how do you get the 9th how do you get the 10th round key. So, you can imagine that you know like what we will try to do is, we will try to work behind we try to work backward. So, what we will try to do is, we will basically know that suppose you know like you have correct ciphertext is x 1 and your faulty ciphertext is x 1 plus F 1 remember the differential is F 1.

So, we will make a guess for the corresponding key the key in one case is K 1 and the other case is also K 1 because you are operating on the same key in both the correct ciphertext and the faulty ciphertext. So, now, we take an ISB. So, ISB stands for the Inverse S Box. So, we also denoted by S inverse so; that means, right if I basically you just do this that is I take the inverse of x 1. So, basically I have got two operations in

parallel. So, I have get x_1 which is my correct cipher and I get another operation where I get x_1 plus F_1 because of the fault. So, here is there is a fault right I get x_1 plus F_1 .

So, now what I do is I know that before this there is an x or; that means, I am just concentrating right now on one byte of the key ok. So, that is again you know you are doing the attack part by part. So, imagine that here K_1 is a corresponding key and this is also again K_1 . So, K_4 right I just take an exhort between them to come to these locations ok. So, remember in the last round before this we have got an S box because you have got an S box here you have got an s box here ok. So, if you want to go behind this right if you want to go behind this then you have to take S inverse you have to take the inverse S box and that is denoted here as ISB.

So, therefore, if I do this right I will probably come to specific state matrix location, I will come to you know like I will come to the get the differential here and then I will try to kind of find out the differential for the another byte location and then I will be trying to use their interrelationships in this equation ok. So, therefore, now if you for example, go back and just relook at the slide, you will probably see the context. For example, right if I just work behind and see the relationships for example, right.

So, for example, here you can see that if you observe this particular position, you see that this differential is twice this differential this differential is equal to this differential this differential is thrice this differential. So, therefore, we will work behind like from F_1 F_2 F_3 and F_4 and try to use these relationships in my equations.

So, therefore, you that is exactly what is done over here and you can observe that if you do that right essentially you see that you basically get that this differential. So, this is the first differential is equal to you know like is equal to double this equation. So, this is equal to double this differential like this is equal to the same as this differential and it is equal to thrice this differential ok. So, that is exactly what is done over here and now you basically guess K_1 K_2 K_3 K_4 .

(Refer Slide Time: 27:17)

The Patterns Gives the Following Equations

- $(\text{ISB}(x_1+K_1)+\text{ISB}(x_1+F_1+K_1)) = \frac{1}{2^{32}}$
- $\text{ISB}(x_2+K_2)+\text{ISB}(x_2+F_2+K_2) = \text{ISB}(x_3+K_3)+\text{ISB}(x_3+F_3+K_3)$
- $\text{ISB}(x_4+K_4)+\text{ISB}(x_4+F_4+K_4) = 3[\text{ISB}(x_2+K_2)+\text{ISB}(x_2+F_2+K_2)]$

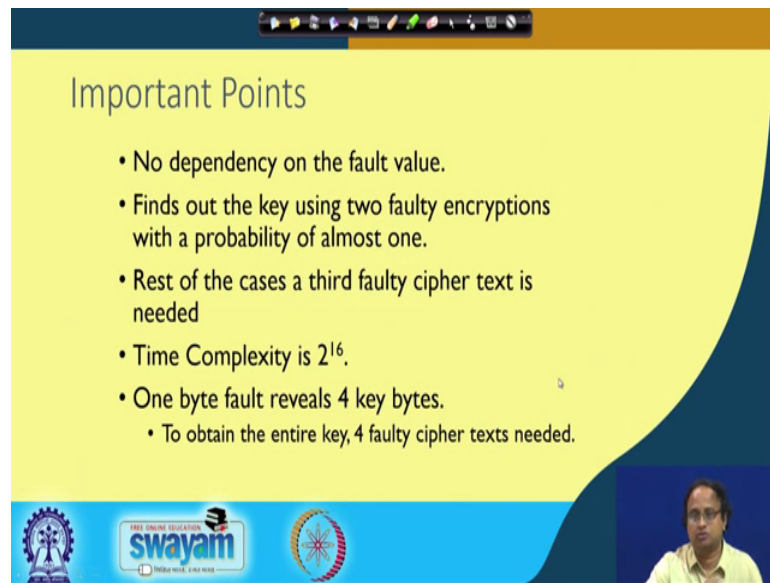
Handwritten notes: $2^{32} \times \frac{1}{2^{24}} = 2^8$, 2^{32} , $(\frac{1}{2^8})^3 = \frac{1}{2^{24}}$

So, you see that $K_1 K_2 K_3 K_4$ has got in the worst you know like generally is got 2 or 2 power 32 possible values but now if we just randomly guess a value of K_1 ok.

So, you know that the left hand side for example, gets fixed and therefore, the probability that a random choice of K_2 will satisfy this equation remember this is a bite right is 1 by 2 power of 8 and therefore, there are 3 such equations. So, totally right the probability that a random $K_1 K_2 K_3 K_4$ will satisfy this equation is 1 by 2 to the power of 8 whole power of 3 that is 1 by 2 power of 24.

So, therefore, right on an expected number of possible key K_1 quartets like $K_1 K_2 K_3$ which will satisfy this equation can be estimated to be 2 power of 8 therefore, you can pretty much assume that 2 powers how to do $K_1 K_2 K_3 K_4$ values will be now reduced to 2 power of 8 values. So, this is the basis of the differential fault attack on AES and we shall see you know like right now we can see that using one such fault, you are basically able to you are able to reduce the key size to 2 power of 8 values.

(Refer Slide Time: 28:21)



Important Points

- No dependency on the fault value.
- Finds out the key using two faulty encryptions with a probability of almost one.
- Rest of the cases a third faulty cipher text is needed
- Time Complexity is 2^{16} .
- One byte fault reveals 4 key bytes.
 - To obtain the entire key, 4 faulty cipher texts needed.

swayam
INDIA WISE, LEAD WISE

So, therefore, right one important point is that there is no dependence in the fault value because eliminated the fault from the equations and we are basically you know like doing this attack with the time complexity of 2 power of 16 because remember K 1 the 2 there are 2 key bytes which were which were guessing in parallel ok. So, therefore, the proportionality of time is with respect to 2 power of is to power of 16.

And one byte fault reveals 4 key bytes because you have ejected one fault and you have got 4 key bytes you have got one key quartet, but you know in the AES 128; there are 4 key bytes and I mean 4 key quartets. So, therefore, if you want the entire key you need 4 faulty ciphertext, if you are injecting the fault at the input of 9th round.

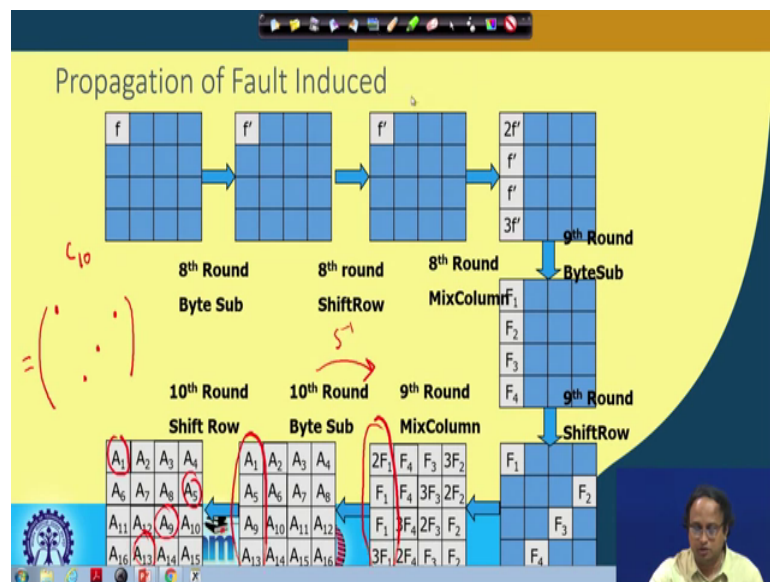
(Refer Slide Time: 29:01)

Important Points

- No dependency on the fault value.
- Finds out the key using two faulty encryptions with a probability of almost one.
- Rest of the cases a third faulty cipher text is needed
- Time Complexity is 2^{16} .
- One byte fault reveals 4 key bytes.
 - To obtain the entire key, 4 faulty cipher texts needed.

But it turns out you can do better by inducing the fault at the input of the 8th round and that is the sweetest spot so whereas, AES 128 is concerned.

(Refer Slide Time: 29:09)



So, let us try to look at how the fault propagates when you have got AES 128 ok. So, here you just you know like in depiction of that pictorially. So, you can see that you have got now the fault of the input of the 8th round, it gets converted to f dash again you know like this gets modified because of the 8th round MixColumns and then you get further $F_1 F_2 F_3 F_4$ and now you come to your you know like the 9th round shift rows and

therefore, this spreads in this way you have got F 1 F 2 F 3 F 4 and then when you do the mix columns you get you know like all these equations. So, that is why now if you target this then you get all the key all the key values or you should be at least getting equations, which are concerning all the 4 key quartets of AES 128 ok. And therefore, right you are you should be able to get with a single fault you should be able to get all the we should be able to get in the entire AES 128 key.

Remember that I have tacitly assumed this, but I have I should probably emphasize on this that is we are trying to get the 10th round of AES ok, but the key schedule rule of AES being reversible if I get the tenth round of AES, you can get the 9th round of AES key and therefore, like we can work we have behind and you can get the input key ok.

But in a block cipher where the key scheduling is not reversible you can still add up this by peeling the rounds. For example, as you get the tenth round and then you basically do a decryption of the last round and you come to the last 9th round cipher and then try to get the 8th round in a similar fashion ok. So, you can kind of peel off the rounds you know like and you can still get the input key it is more difficult, but still possible.

(Refer Slide Time: 30:45)

The Patterns Gives the Following Equations

- $ISB(x_1+K_{00})+ISB(x_1+A_1+K_{00})= 2[ISB(x_8+K_{13})+ISB(x_8+A_5+K_{13})]$ $2^{32} \rightarrow 2^8$
- $ISB(x_8+K_{13})+ISB(x_8+A_5+K_{13})= ISB(x_{11}+K_{22})+ISB(x_{11}+A_9+K_{22})$ $2^{128} \rightarrow (2^8)^{16} = 2^{32}$
- $ISB(x_{14}+K_{31})+ISB(x_{14}+A_{13}+K_{31})= 3[ISB(x_8+K_{13})+ISB(x_8+A_5+K_{13})]$ $2^{-96} = 2^{96}$

So, therefore, right in this case when you want to perform this attack on AES 128, remember now we would like to you know like exploit these equations. So, for example, right now if I want the you know like one key quartet then I will be targeting this these

equations like $2^{F1} F1$ and 3^{F1} and if you want to use this then you see that when you are working behind right you are working behind the Sbox which is this.

So, you are basically going you know like this is your S inverse and you see that here you are you have got these equations I mean these bites involve A1 A5 A9 and A13 ok. Now because of the shift rows these gets again located to these positions A1 A5 A9 and A13. So, therefore, write your equations will be concerned with A1 A5 A9 and A13, but remember that the cipher that you have essentially right will be. So, therefore, in the cipher bytes right you have to basically concern about these cipher byte locations ok. So, this is the 10th round cipher C10 for example so that is you know like. So, therefore, by using this you again derive similar equations and therefore, right you should be able to get the corresponding you will get the corresponding or understand the corresponding equations.

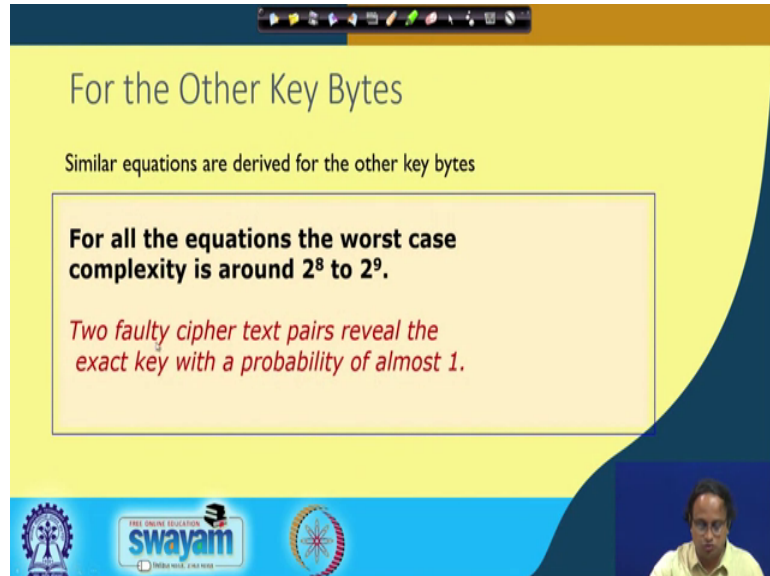
For example these are the equations here. So, you can see that I have in this equation is involved A1 A5 A9 and A13 as I have just mentioned. Again you know like this has got a complexity of $1 \text{ by } 2^{\text{power of } 20}$, I mean probability of $1 \text{ by } 2^{\text{power of } 24}$ while guess of the key quartet. So, the key quartet here involves these keys like K 00 K 13 K 22 and K 31.

So, these are the 4 key bytes which are involved and therefore, right again this reduces to from $2^{\text{power of } 32}$ to $2^{\text{power of } 8}$ values ok. Now imagine that you can try to get the equation for the next for the next column and similarly for the 3 other columns and therefore, the entire AES key right which is essentially had got $2^{\text{power of } 128}$ initial values will now get reduced into $2^{\text{power of } 8}$ whole power of 4 which is around $2^{\text{power of } 32}$ with one single fault ok.

So, now, for now you can actually try this exercise once more so; that means, you can again try another faulty cipher text and you can again you know like take another get another candidate for the AES key and you take an intersection between them. So, you can understand that if you have a reduction from $2^{\text{power of } 128}$ to $2^{\text{power of } 32}$; that means, with something like $2^{\text{power of } 96}$ as a probability of probability or as a filter probability, then if you take one more pair the expected number of key which will survive is expected to be only 1. So, therefore, you know you can pretty much expect

that if you take one more fault short or shorten the faulty value, then you will probably get a unique value of the key ok.

(Refer Slide Time: 33:39)



For the Other Key Bytes

Similar equations are derived for the other key bytes

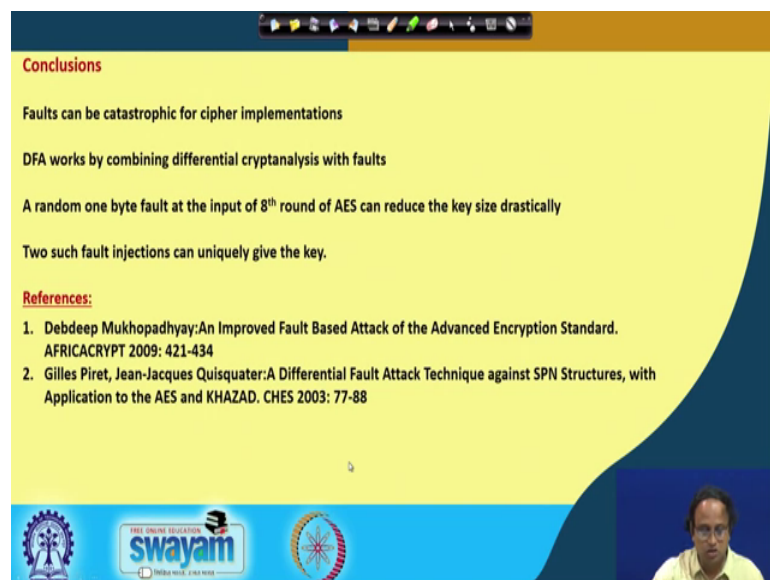
For all the equations the worst case complexity is around 2^8 to 2^9 .

Two faulty cipher text pairs reveal the exact key with a probability of almost 1.

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 institutions, and a small video inset of the presenter.

So, that is essentially the crux of how the fault attack works on AES and ah; that means, that you know like two faulty ciphertext will reveal the exact key, with a probability of almost 1.

(Refer Slide Time: 33:45)



Conclusions

- Faults can be catastrophic for cipher implementations
- DFA works by combining differential cryptanalysis with faults
- A random one byte fault at the input of 8th round of AES can reduce the key size drastically
- Two such fault injections can uniquely give the key.

References:

1. Debdeep Mukhopadhyay: An Improved Fault Based Attack of the Advanced Encryption Standard. AFRICACRYPT 2009: 421-434
2. Gilles Piret, Jean-Jacques Quisquater: A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. CHES 2003: 77-88

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 institutions, and a small video inset of the presenter.

So, to conclude faults can be catastrophic for ciphered implementations, DFA works by combining differential cryptanalysis with faults a random 1 byte fault and the input of 8

round of AES can reduce the key size drastically and two such fault injections can uniquely give the key ok. Remember that now we have got 2 power of 32 keys which we have which we can get with one single fault.

(Refer Slide Time: 34:15)

References:

- Debdeep Mukhopadhyay and Rajat Subhra Chakraborty, *Hardware Security: Design, Threats and Safeguards*, CRC Press

HARDWARE SECURITY
Design, Threats, and Safeguards
Debdeep Mukhopadhyay
Rajat Subhra Chakraborty

swayam
FREE ONLINE EDUCATION
INDIA'S MOST AFFORDABLE

So, here are 2 important references which we have followed in this work and in this presentation of course, you can always refer to the main textbook.

So, thank you for your attention.