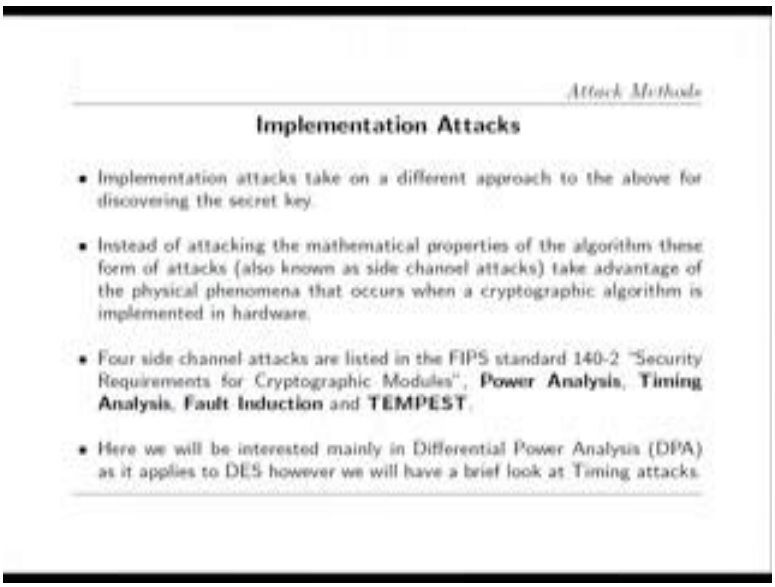
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 58
Implementation Attacks

(Refer Slide Time: 00:21)



So, we talk about implementation attack. So, this is basically so far we have seen for crypt analysis purpose, we have seen that we are looking at the mathematical form of the cryptosystem the mathematical how they bid like for differential attack. We will try to see the whether we can have some differential trails in the input and then you see that we have some output difference also coming with some significant probability. Or in the S-box or in the say linear crypt analysis or algebraic data we have seen whether we have some linear relationship between the input in between the plaintext along with the key with the ciphertext, we have system of equations. So, those are basically some mathematical, we look at the function of this mathematically then we have some attacks on that

So, this is other types of attack, this is called implementation attack. So, it takes on a different approach to the above all discovering the secret key. So, this is a different type of attack to discover the secret key. So, this is instead of attacking the mathematical properties of the algorithm, this form of attack take advantage of the physical phenomena

that occurs when the cryptographic algorithm is implemented hack in hardware. Suppose we are a running an encryption algorithm. So, suppose RSA, we are running a RSA encryption algorithm, so then you see when you we have hardware for running for this encryption or we have hardware to implement this AES. So, when AES is running this hardware we see the physical phenomena that means, we look at the power consumption of these, we look at the how much time is taking for each instruction. So, these are all physical phenomena. So, based on this physical phenomena, we try to gaze the key, so that is called implementation attack.

So, we take the advantage of physical phenomena that occur when a cryptographic algorithm is implemented in hardware or software. So, this is called side channel attack. Like as if somebody is sitting in the side of the computer where or some machines some system where this physical phenomena are being observed. So, this is that is why it is called side channel attack; somebody is sitting in the side and observing this power consumption, power fluctuation or the time delay. So, if we have a, we will talk about the timing attack like when you are calculate the exponentiation m to the power e for our RSA encryption scheme.

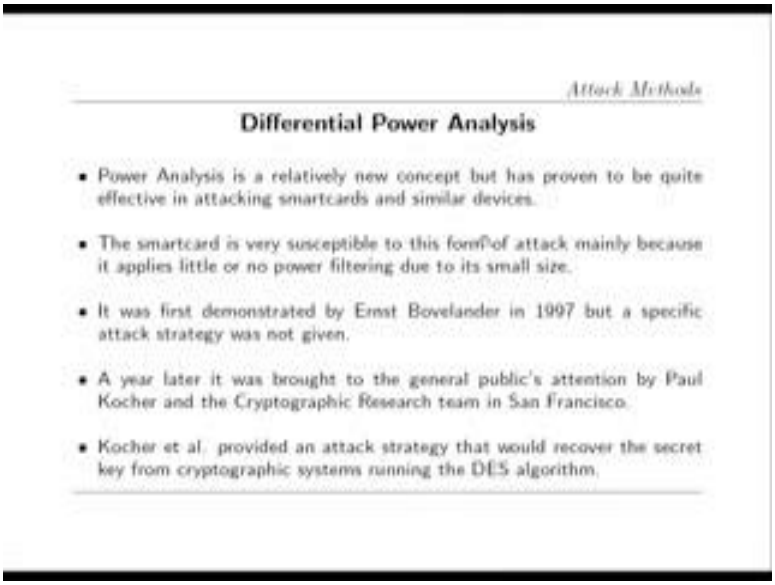
So, they are if e is the basically the e is the public key, but when we calculate c to the power d the Bob is calculating c to the power d to get the, so c is the m to the power e that is the ciphertext send to Bob, and Bob will calculate c to the power d. So, this is the secret key of Bob the when you see when Bob calculate c to the power d, now e it is a basically 0, 1 bits, now this is n square and multiply methods.

So, when we have only 0, then we will not go for the multiplication. So, when that loop has a when we have a one then we will go for multiplication. So, by looking at the time how much it is taking instruction. So, we can gaze the key, so that is called timing attack, so that is the physical phenomena that occur when we implement the cryptographic algorithm in hardware or software, so that is why it is called side channel attack. As if somebody is sitting in front of that machine or computer and looking this observing this physical phenomena.

So, there are four types of side channel attack are listed in the FIPS standard. So, the power analysis this is one of the side channel attack, timing attack, timing analysis fault induction and tempest. So, we will talk about mostly this power analysis Differential

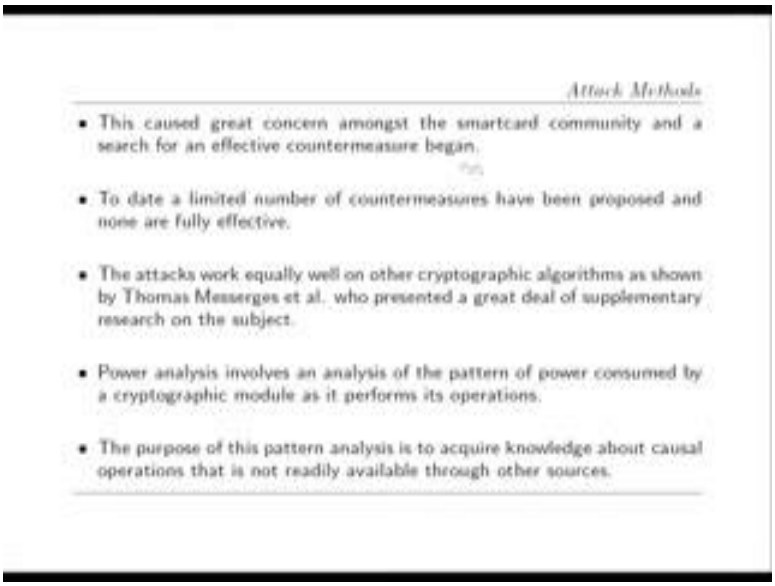
Power Analysis - DPA which is basically applied to DES to break DES, and then we talk about timing attack also in this lecture.

(Refer Slide Time: 04:54)



So, let us talk about differential power analysis. So, power analysis is relatively new concept, but it has proven to be quite effective to attacking a smartcard or similar devices. If a smartcard as very low power I mean the power or capacity, it is very low power capacity, so this power analysis is first implemented on the smartcard. And it was demonstrated by Bovelander in 1997, but specific attack strategy was not given. But later on this the same it was brought to the general public attention by Kocher and the Cryptographic Research Team in San Francisco. So, they brought this idea of the power analysis for the cryptographic breaking the cryptographic primitives. So, Kocher et al. provided a attack strategy that would recover the secret key from the cryptographic system running the DES algorithm. So, this is the generalized version of what is the basically a, but the basic power analysis attack was on the smartcard devices.

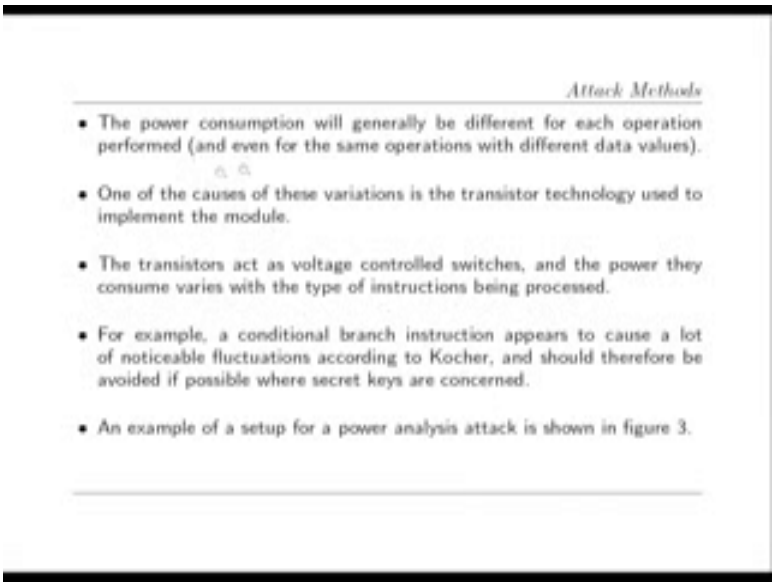
(Refer Slide Time: 06:09)



So, we will talk about that device I mean the, so this cause great concern and amongst the smartcard community and the search for an effective countermeasure began. To date a limited number of countermeasures has been proposed and none are fully effective. So, the smartcard this can be attacked by this power analysis. So, people start of having the countermeasure of to prevent this, but so far that some attempt is there, but so far not much progress in this propose are fully effective.

The attack work equally well on the other cryptographic algorithm as shown by the Thomas et al. who presented a great deal of supplementary research on this subject. So, power analysis involve an analysis of pattern of the power consumed by the cryptographic module as it perform its operations. So, suppose we are running a cryptographic algorithm, suppose say for example, suppose we are running a cryptographic encryptions scheme we are running encryption scheme in hardware. Now, this power analysis will just analysis the power consumption, it is doing when it is implementing in hardware, so that is the observation we have to make in power analysis. So, how much power consumption, power fluctuation, so we have to have a machine to detect that. So, we will come to know. The purpose of this pattern analysis is to acquire knowledge about the actual operations that is not readily available through other sources.

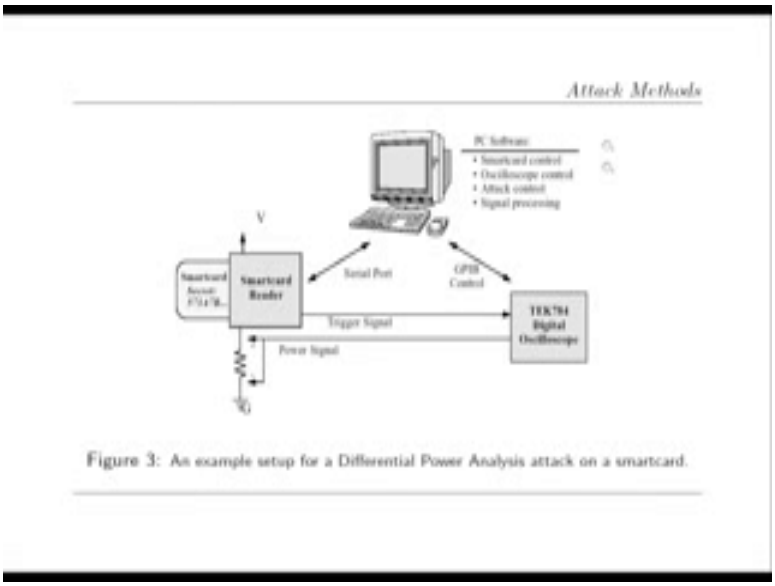
(Refer Slide Time: 07:56)



The power consumption will generally be different for each operation performed because it will depend on the different operation because it has different instruction. So, every algorithm, so suppose we have a encryption algorithm DES, AES, RSA. So, every algorithm has different type of instruction. So, depending on the instruction also the power consumption will be different and also the different data will give the different power consumption, power fluctuation so, the power consumption will be different for each operation performed.

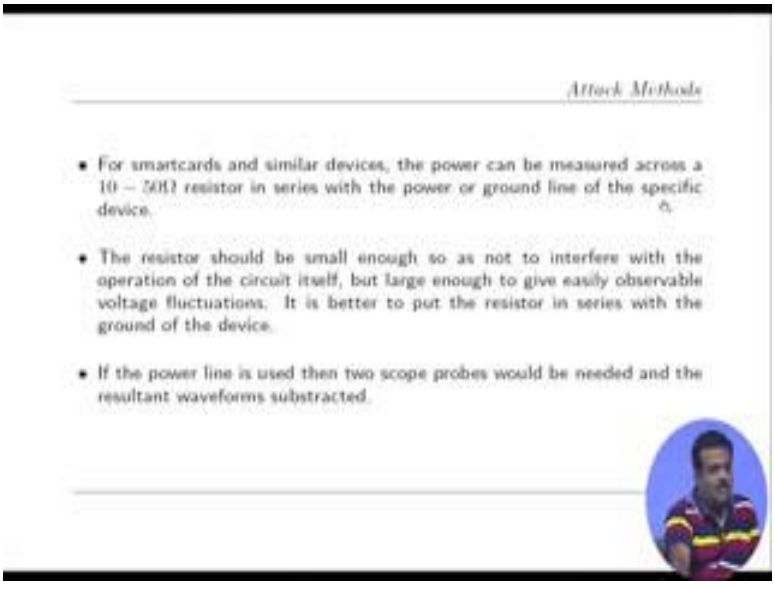
One of the cause of this variation is the transistor technology used to implement the module. The transistor act as a voltage controlled switches and the power they consume varies with the type of instruction being processed we have said this. For example, conditional branch instructions appear to cause a lot of noticeable fluctuation according to Kocher. If you have a conditional branch instruction in our code in our, that the algorithm which you are implementing may be some encryption algorithm if there is a conditional branch instruction; if there is a conditional branch then it will have a lot of noticeable fluctuation in the power by according to the Kocher. So, we have given an example of a setup of power analysis in figure 3.

(Refer Slide Time: 09:30)



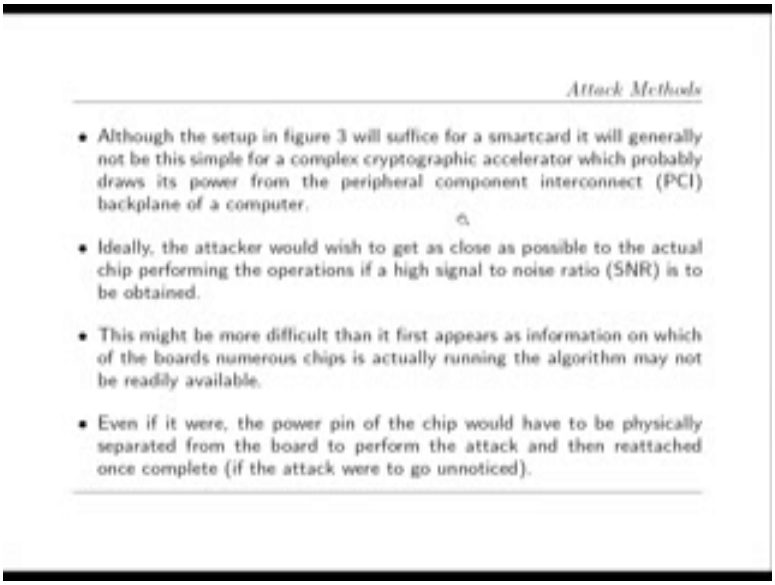
So, where is figure 3, yeah this is a figure 3. So, this is example of setup of power differential power analysis on a smartcard device. So, here we have a smartcard reader; and here we have in the connection, this is connected by a serially and this is the PC where we are analyzing the, but capturing the card fluctuation signal. So, this is the signal processing technique we are doing this PC software has this part, these are the feature smartcard control, attack control, signal processing.

(Refer Slide Time: 10:05)



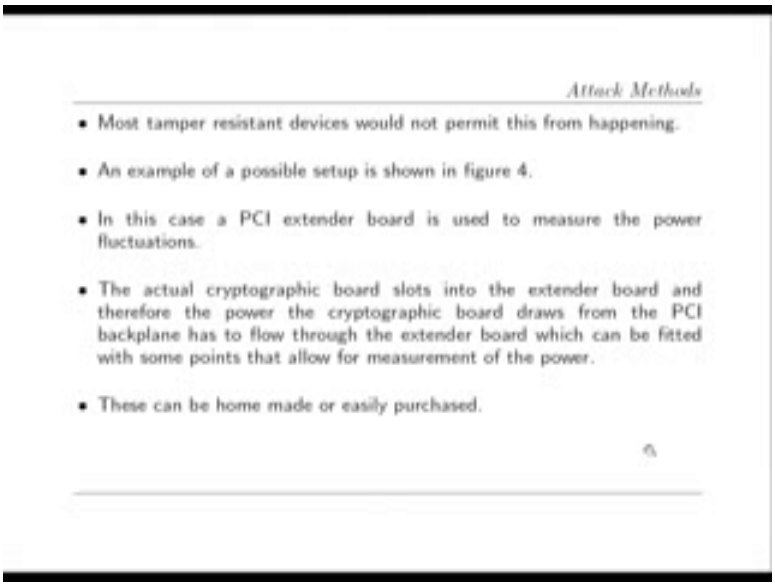
So, these for smartcard and similar devices the power can be measure across a 10 to 50 ohms register in a series with the power or ground line of the specific device. The register would be small enough so as not to interface with the operation of the circuit itself, but large enough to give the easily observable voltage fluctuations, so that we can easily get the power fluctuation or voltage fluctuation. It is better to put the register in a series with the ground of the device. If the power line is used then the two scope probes would be needed and the resultant waveform substracted. So, this is the simple device whether this is the device this is the setup for differential power analysis setup or smartcard. We have a general device, which will be working on any cryptographic algorithm. So, we will we will talk about that.

(Refer Slide Time: 11:06)

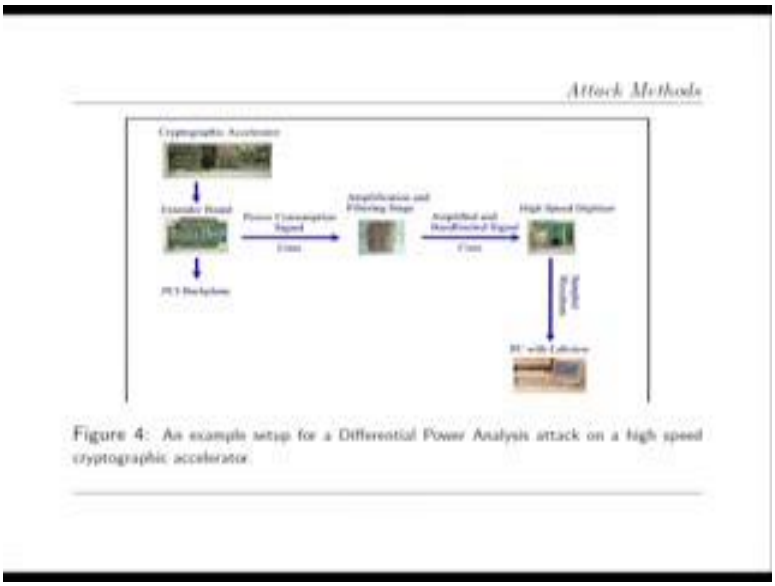


So, although the setup the design we have seen will suffice for smartcard, but it will generally not be this simple for complex cryptographic algorithm, which probably draw a power from the peripheral component interconnect - PCI. So, we have little more complicated circuit little more complicated design for this cryptographic algorithm to get the power fluctuation. Ideally, the attacker would wish to get a close as possible to the actual chip performing the operation if the high signal to noise ratio is to be obtained. This might be more difficult than it first appear as information on which the board numerous chip is actually running on the algorithm may not be readily available. So, even if were, the power pin of the chip would have been physically separated from the board to perform the attack then the reattach once complete.

(Refer Slide Time: 12:15)



(Refer Slide Time: 12:18)

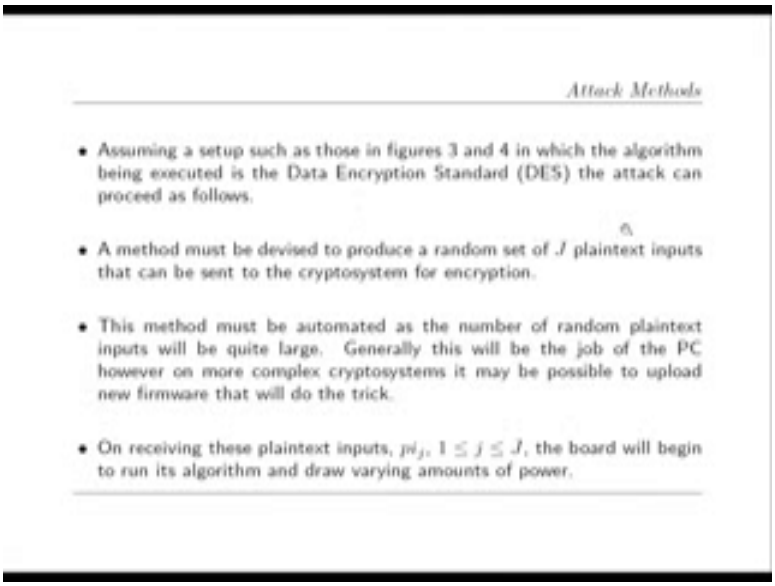


So, most we will we will come to the figure of that. So, this is basically the general figure than the smartcard one. So, this is general figure which is the setup of the differential power analysis for a high speed cryptographic algorithm. So, here this is the hardware to run the cryptographic, so encryption algorithm or decryption algorithm and it may take power from PCI back backplane. So, we need to have external board which is so capturing the power signal and it is ultimately sending to this computer - this PC, to analyze this. So, this is little more complicated than the simple smartcard, this is the

simple one, this is the simple smartcard power analysis attack on smartcard, but this is little complicated because it may take this may take power from the PCI black.

So, if the ultimately the signal is going to the PC to analyze. So, most tamper resistant device would no; so, this is the figure we have seen in this case the PCI extension board is used to measure the power fluctuation. The actual cryptographic board slot into the extender board and the therefore, the power of the cryptographic board draw from the PCI backplane has to flow through the external board which can be fitted with some point and that allows the measurement of the power. So, this system can be homemade or one can purchase this system.

(Refer Slide Time: 14:00)

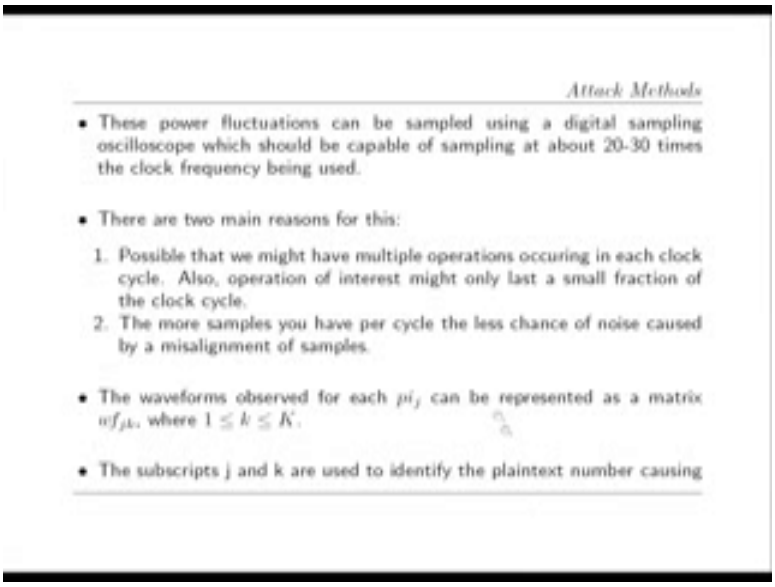


So, assuming this setup either 3 or 4, so we want to apply this on the DES. So, we want to apply this on the data encryption standard algorithm for this differential power analysis. So, for that, we have this setup, we have this model differential power analysis model, we have the chip and ultimately with this power fluctuation on the power consumption or power fluctuation signal should go to the computer. To attack on this DES - data encryption standard, this is the differential power analysis on DES. So, we choose J many plaintext randomly. So, this is a chosen plaintext setup or the known plaintext setup we can say, but this plaintext has to be chosen randomly, so that is why we can say this is chosen plaintext attack. So, we choose J many plaintext randomly and these we are denoting by p i j. So, j is starting from this small j is starting from 1 to j.

And then the method must devise the produce a random number random setup J plaintext input and that can send to the cryptosystem for encryption.

This method can be automated as number of random plaintext input will be quite large. Generally this will be the job of the PC however, on more complex cryptosystem is may be possible to upload the new firmware that will do the trick. So, ultimately we choose the J many plaintext randomly, so that should be done either by the PC or some complicated or some complex cryptosystem or some file new file should able to give that. So, why randomly, we will come to that, why need this random plaintext. Now we after receiving the plaintext we ask the board to begin the run the algorithm and draw the and it will draw varying amount of power because you ultimately want to encrypt this plaintext and get the ciphertext, so for that it will draw various amount of power.

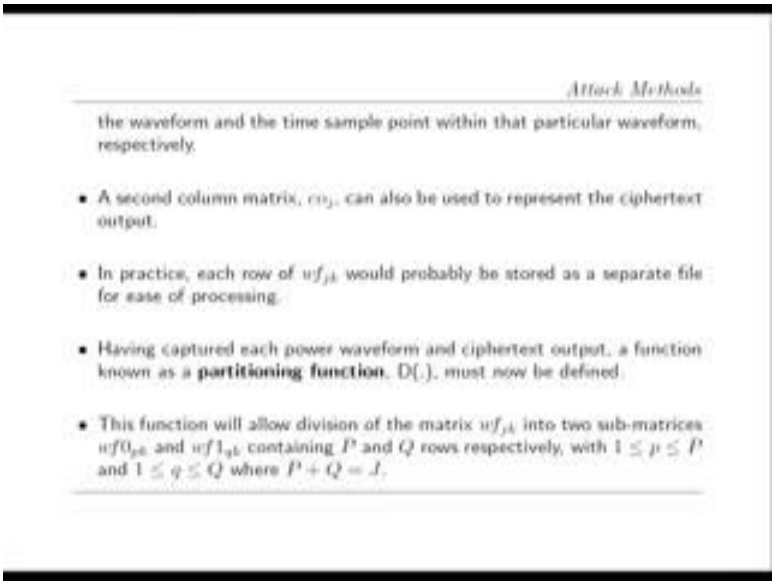
(Refer Slide Time: 16:25)



So, this power fluctuation can be sampled using the digital sampling which should capable of sampling at about 20 to 30 times the clock frequency being used. There are two major reason for this possible that we might have multiple operation occurring in each clock cycle. Also, operation of interest might only last for a small fraction of the clock cycle. And the more sample you have per cycle the less chance of noise caused by a misalignment of samples. The waveform observed. So, after running this, so we take the sampling digital sampling of the power fluctuation and suppose the waveform

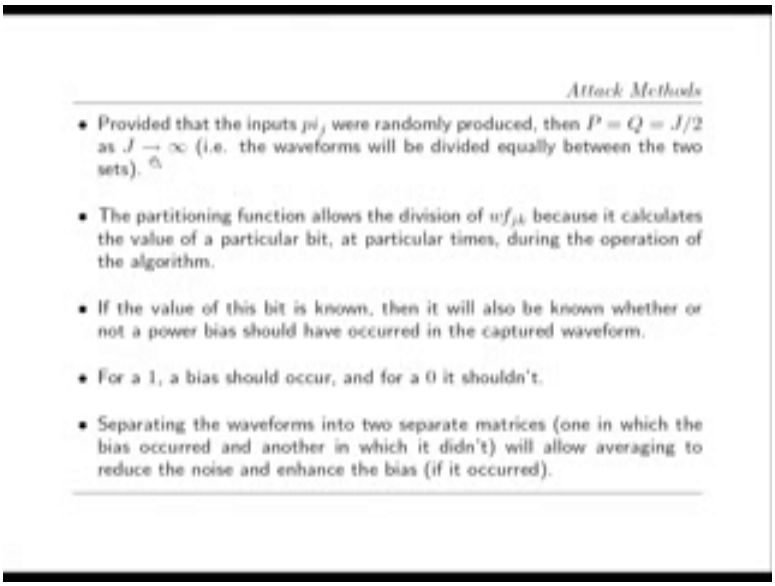
observed when we give the plaintext p i j is represented by the matrix w f j k. So, this is a matrix j and k are used; j is used for the plaintext.

(Refer Slide Time: 17:30)



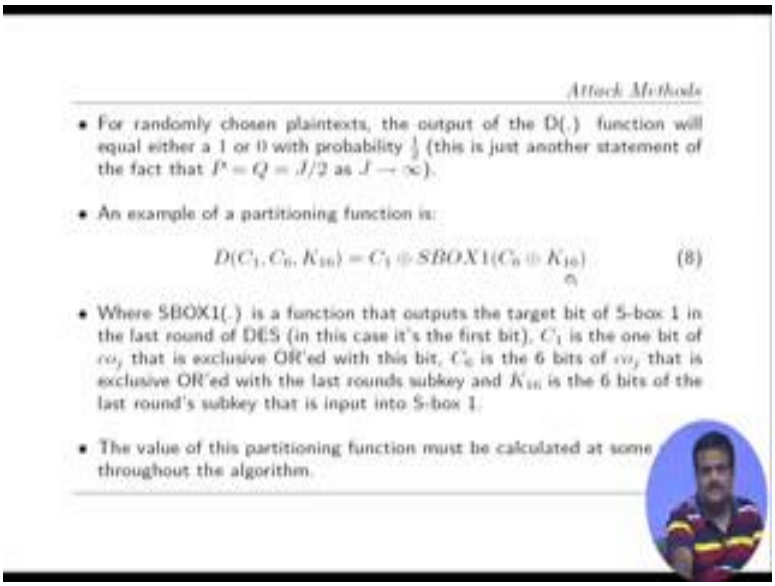
And we denote this c o j as the corresponding ciphertext. So, we have the plaintext p plaintext is denoted by what p i j, and c o j is the corresponding ciphertext. And this is the corresponding matrix which is called waveform matrix due to this is the sampling digital sampling on the power fluctuation. This is the waveform matrix w f i k. So, now, we partition this matrix into the rows this is the rows w f 0 p k and w f 1 p k. And suppose there are P rows which are having this 0 p k, and Q rows which are having 1 p k and such that p plus q is equal to j. So, this is the partition function which is denoted by D. So, it will partition the matrix into two sub matrixes this and this.

(Refer Slide Time: 18:35)



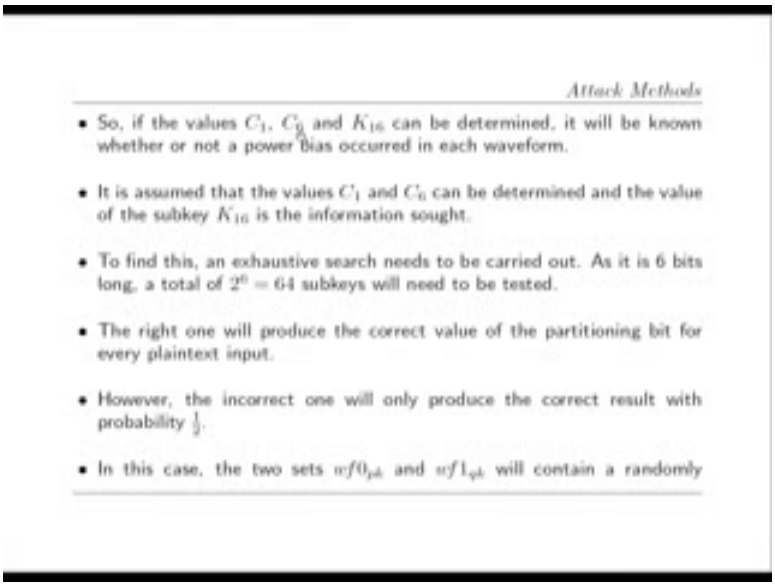
Now provided that the input p i j, so this is the this is the plaintext were randomly produce then if this input is truly random I mean if we produce this input randomly. Then as J tends to infinity P and Q will be J by 2, because there will be half, half that 0 this 1 for 1, the bias occur and 0, bias should not occur. So, the partition partitioning function allow the division w f the matrix, because it calculates the values of the particular bit, at particular times, during the operation of the algorithm. If the value of the bit is known, then it will be known whether or not the power bias should have occurred in the captured waveform. For 1, bias should occur; and for 0, bias should not, so and that if this is chosen randomly then this should be half, half. Separating the waveform into two separate matrix will allow averaging of the averaging to reduce the noise and enhance the bias, so that will do.

(Refer Slide Time: 19:46)



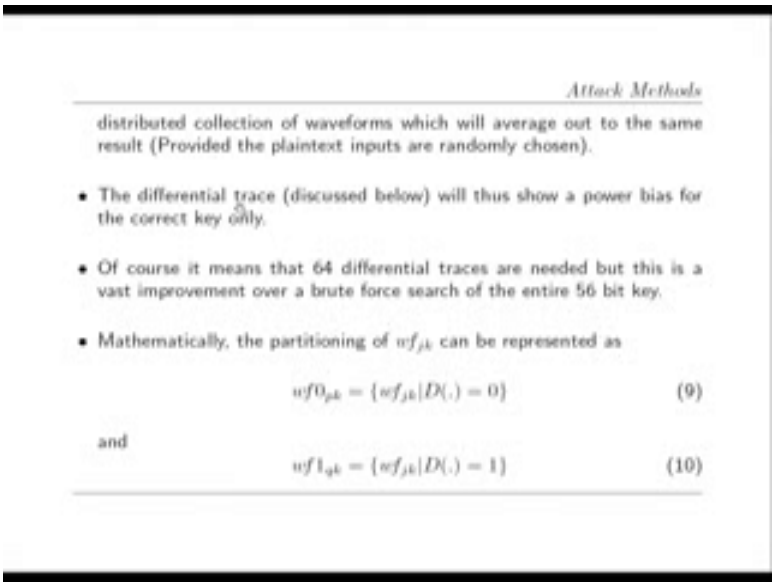
So, basically this is the partition partitioning and if we choose this p i j's randomly then it will be half, half basically. So, half of the bias will be half of half - 1 and half of will be half zero with probability half also as J tending to infinity. And this is an example of a partitioning function this is D C i C k and this is the last bit K this is the round key. So, round key K 16 equal to C 1 XOR with S-box 1; and the input of the S-box is C 6 XOR with K 16. So, this is the function coming from DES where S-box is the function that output the target bit of the S-box 1 in the last round of DES. C 1 is the 1 bit of the ciphertext c o j that is XOR with the bit C 6 that is the 6th bit of the ciphertext. And that is again after running the S-box that is again XOR with the no this c 6 is XOR with the last rounds of t and which is basically K 16 and this is of 6 bit. So, we will do the exhaustive search on this 6 bits to get the value. So, let us talk about how. So, 6 bits of the last separate key sub key and that is input into the S-box 1. The value of this partitioning function must be calculated at some point throughout the algorithm.

(Refer Slide Time: 21:38)



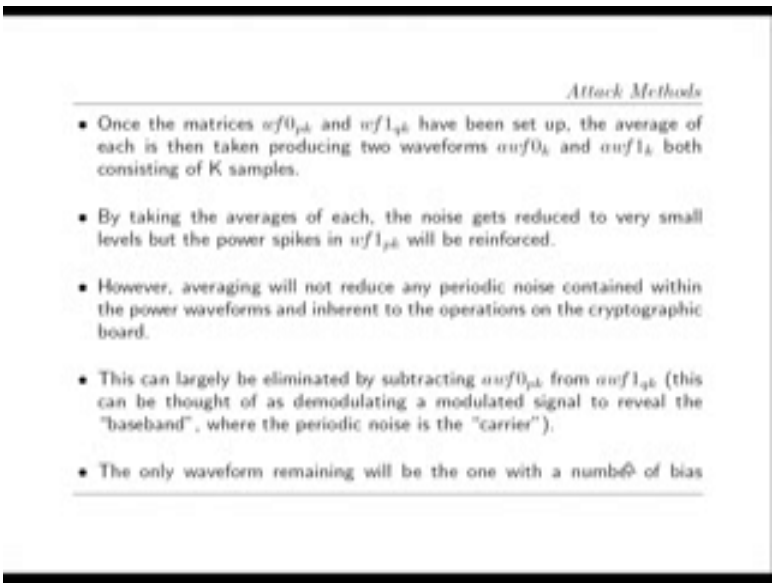
So, if the value C 1, C 6 and K 16 can be determined, it will be known whether or not power bias occurred in each waveform. So, it is assumed that the values C 1 and the C 6 can be determined and the value of the K 16 is information sought. So, to find this we do the exhaustive search on this K 16. So, this is 6 bits. So, there are total possibilities 2 to the power 6; that means, 64 possibilities possible sub keys are there. So, we will choose we will try this for all sub keys the right one will produce the correct value of the partitioning bit for every plaintext input. However, the incorrect one will only produce the correct result with probability half, because we are choosing this plaintext as random, so that is why it will give us half, half. So, in this case, the two set that bias know the w f 0 and w f 1 will contain a randomly distributed collection of waveform, which will average out the same result.

(Refer Slide Time: 22:51)



The differential trace will thus so the power bias of the correct key only of course, this means that 64 different traces are needed, but this is the this is a vast improvement over the brute force search of the entire 56 key. So, if you write it mathematically, so this partition of w f i k can be represented by this where this is giving us 0 this function and this is giving us w f 1 k it is giving us 1. So, this is equation 9, equation 10.

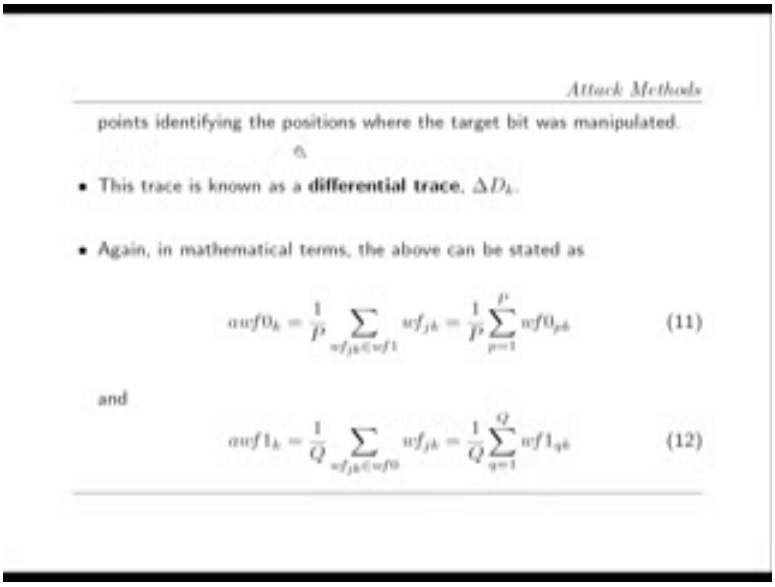
(Refer Slide Time: 23:34)



Now, once the matrix w f 0 and w f 1 has been setup, the average of each is then taking producing two waveform. So, a w f 0, this is the average. So, we need to divide by d. So,

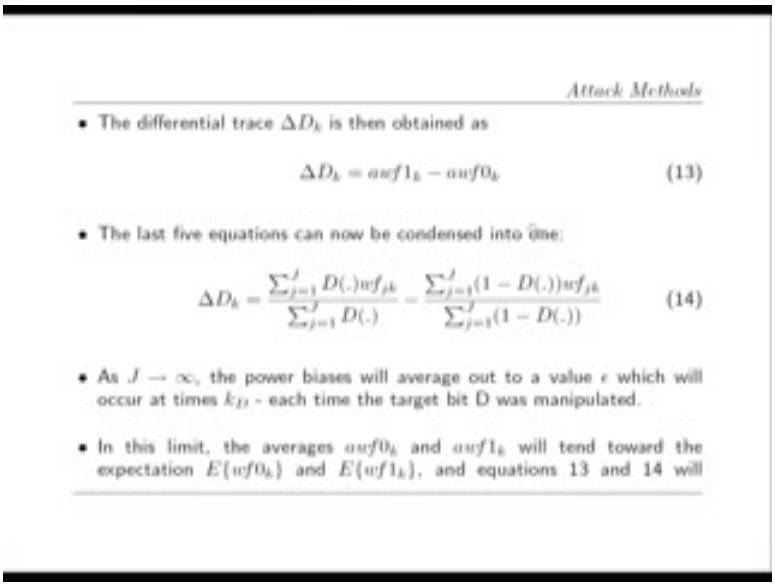
you look at that expression. So, and a w f 1 both consisting of k samples. So, we need to divide by k by taking the average of each, the noise gets reduced to a small level, but the power spike in w f 1 will be reinforced. So, however, averaging will not reduce any periodic noise contained within the power waveform and inherent to operation of the cryptographic board. This can largely be eliminated by subtracting this average of a f 0 and average of this w f 1. The only waveform remaining will be the 1 which with a number of bias point identified the position where the target bit is manipulated.

(Refer Slide Time: 24:40)



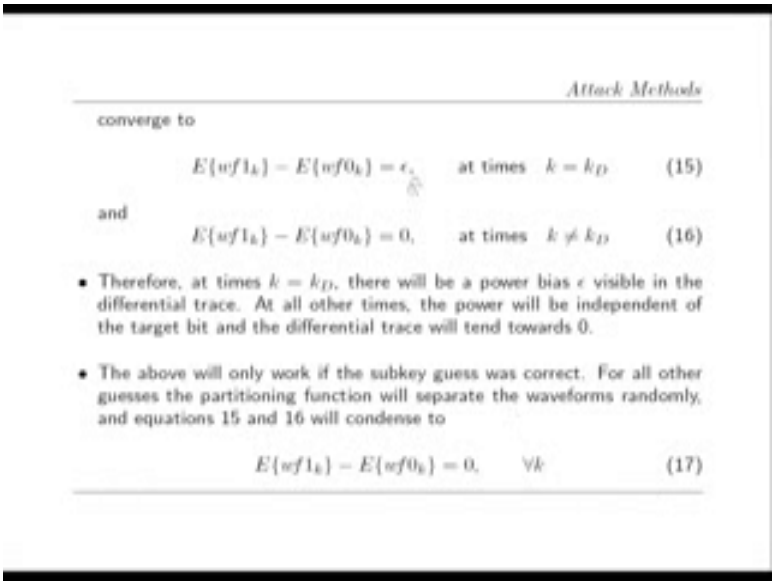
So, this trace is known as differential trace and this is denoted by delta D k. So, this is the expression for a f 0, and this is the expression for a f 1 because there are P many so this is the averaging on this and there are Q many of such row, so this is the averaging on this.

(Refer Slide Time: 25:09)



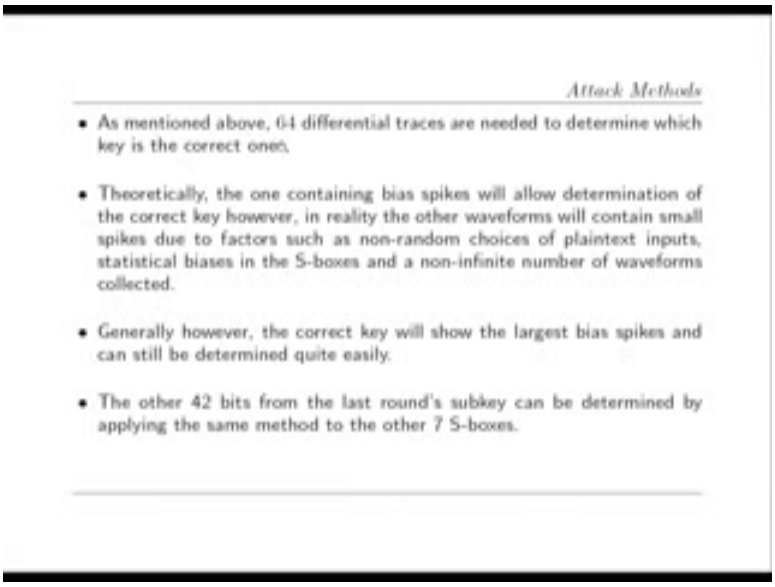
So, the difference of their mean, so we take the difference that is called differential trace, so delta of D k. So, if we just combined, if we just write the expression, so it will be in this form. So, if J tending to infinity then the power bias will be average out the value epsilon which will occur at times D k. So, each time the target bit D was manipulated. So, in this limit the average so on a f 0 and average on a f 1 will tend towards the expected value of w f 0 and w f 1.

(Refer Slide Time: 25:54)



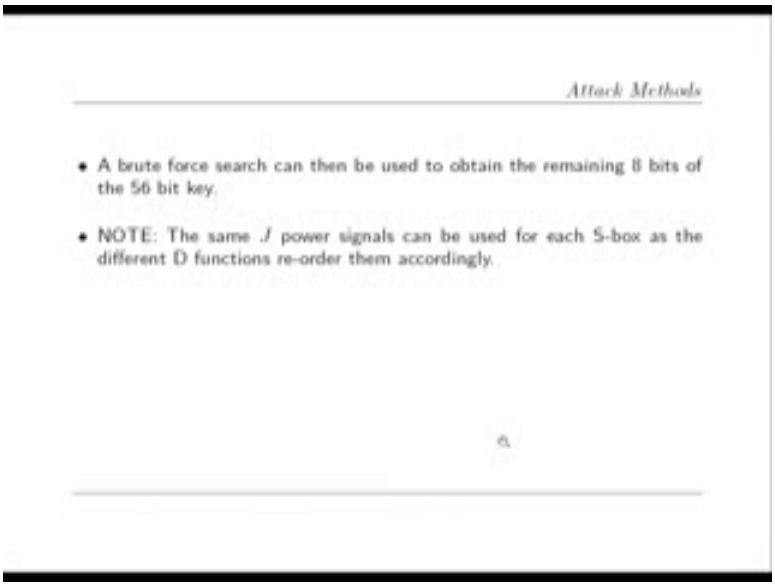
And the equation 3 and 14 will converge to epsilon and 0. Therefore, at time k is equal to k D, there will be a power bias epsilon visible in the differential trace. And at all times, the power the power will be independent to the target bit and the differential trace will tend towards 0. So, above will work only if the sub key guesses are was correct. For other guesses the partitioning function will separate the waveform randomly and the equation 15 and 16 will condense to this 0.

(Refer Slide Time: 26:33)



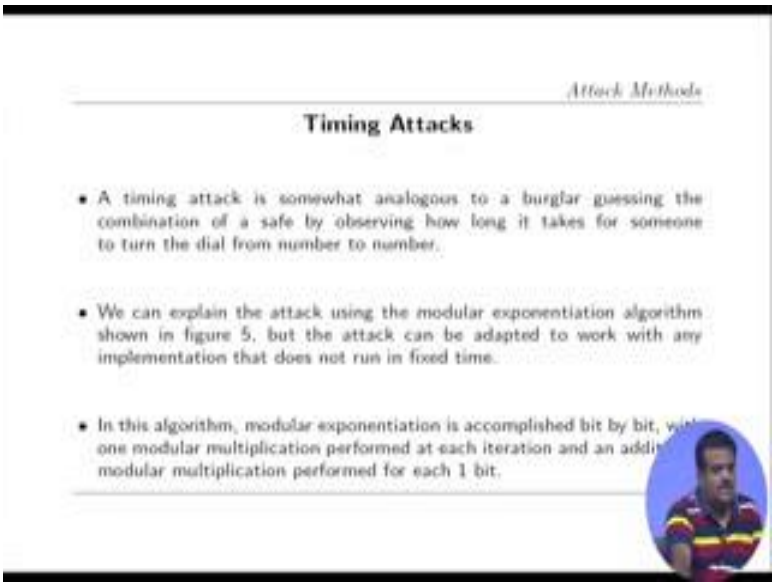
So, as mentioned above 64 different key. So, this is this is the exhaustive search on the sub key. So, 64 different trace are needed to determine which key is correct one. So, theoretically, the one obtaining the bias spike will allow determination of the correct key however, in reality the other waveform will contain the small spike due to the factors such that non-random noise of the choice of the plaintext, and some statistical bias will be there in the S-box. So, this may affect our correct choice of the key. So, generally; however, the correct key will show the largest bias spike and that can still determined quite easily. The other 42 bits from the last rounds sub key can be determined by applying the same method to the other 7 S-box.

(Refer Slide Time: 27:27)



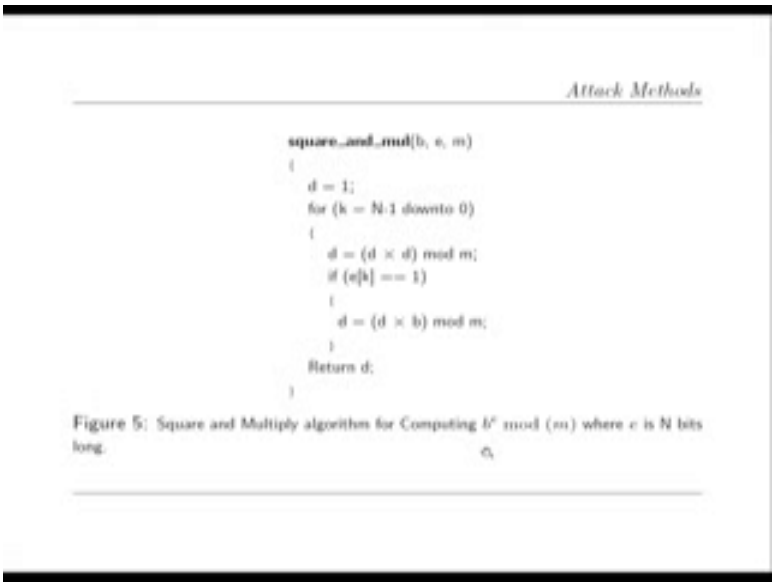
So, this is the attack model.

(Refer Slide Time: 27:31)



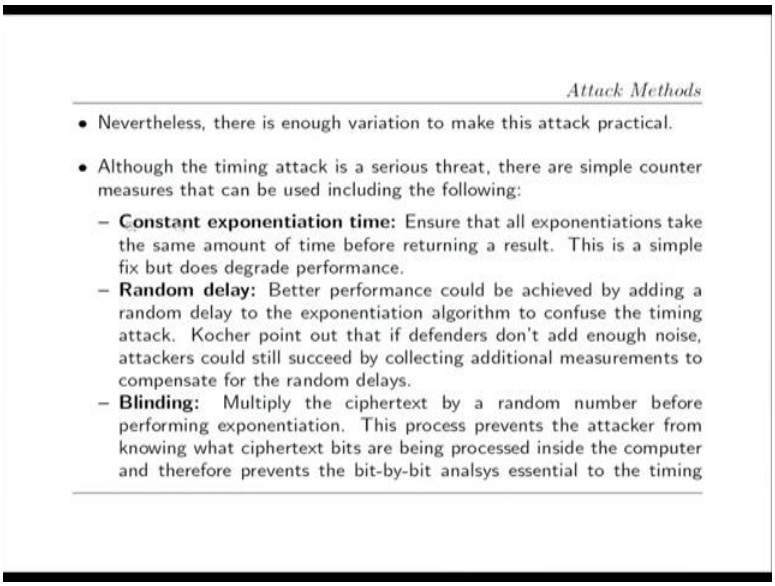
So, now, we will talk about the timing attack on this. This is also part of the implementation attack. So, timing attack is basically tell you suppose we are running a code in the machine. Suppose, let us quickly go to this square and multiply method.

(Refer Slide Time: 27:44)



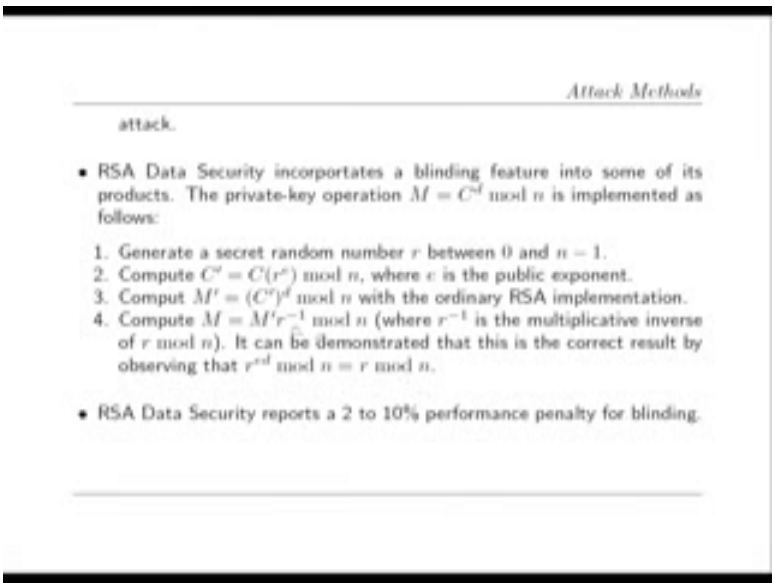
Suppose, we are trying to find out b to the power e mod m. Suppose this is this type of expression will required suppose RSA Alice is sending a message to Bob. So, Bob public key is c secret key is d. So, Alice is computing m to the power e m is the message, so that is the c. The opposition will receiving c, Bob is computing c to the power d now d is the secret key. So, this type of expression we need to calculate. So, for this we know this is the square and multiply method. So, every time we square it and we multiply if there is a 1. So, if we just observe the time in this loop if this is a loop, if we observe this loop is taking more time then we guess that the corresponding bit is 1. So, this is also a side channel attack. So, this is implementation attack. So, somebody is sitting in side of the computer and checking how much time is computer is taking to run this code. Now, if for sometimes, it is taking more, then that means, that corresponding bit is 1; otherwise it is 0. So, this way one can gaze the bits of this.

(Refer Slide Time: 29:07)



So, to prevent this, this is to prevent this, these are the things we need to do constant exponentiation time. So, each, we should take constant some random delay we can produce; if it is not multiplying we can have some random delay. Or blinding means we can multiply some multiply ciphertext by a random number before performing the exponentiation.

(Refer Slide Time: 29:30)



So, this way one can prevent that. So, this is the way we can prevent we have this random the number we can multiply this to prevent this timing attack. So, this is also an example of implementation attack.

Thank you.