Differential Power Analysis

Siva Sai Yerubandi
Department of Electrical & Computer Engineering,
Oregon State University, Corvallis, Oregon 97331 -USA.
E-mail: yerubasi@ece.orst.edu

Abstract— **Most modern cryptographic devices are implemented using semiconductor logic gates, which are constructed out of transistors. They leak information about the operations they process. To attack the cards using Differential Power Analysis requires a high level of technical skill in several fields. Yet this can be performed using a few thousand dollars of standard equipment. It has been seen that using this equipment the pin numbers and secret keys can be often broken in a few hours or less. These attacks can be made automated once a device has been characterized. This paper examines methods for analyzing power consumption to get the secret keys and also discusses the ways for building systems that can operate safely using existing hardware, which leaks information.**
Keywords: **SPA, DPA, Differential Power Analysis, Smart Card Attacks, Power Attacks, Power Analysis.**

I. Background.

There are many ways a to attack a security system. This is because the security depends on many cases. And in most cases there is not much coordination between the software developers and hardware engineers. A correct implementation of a cryptographic algorithm is not sufficient. There are failures due to incorrect computations and information leakages during the secret key operations. Conventional techniques such as differential and linear analysis are most helpful in exploiting the weakness of the cryptographic algorithms that can be represented as mathematical objects. But these are limited to only those algorithms. When it comes to the case of hardware, attackers cannot get the weakness because hardware can be implemented in many different ways and also in real life, problems become more complicated and are much more hard to understand. Eliminating side channel information or preventing this from being used to attack the secure system is a very good area of research.

II. Introduction.

Presently a lot of research is gong on in the field of smart cards to provide security against the attacks. One of the major attacks on the smart cards, which are considered tamper resistant, is because of leakage of side channel information. In most of the cases with smart cards their cryptographic key or the authentication certificate should be kept secret and as well prevent it from being made counterfeit. This is because all the cryptographic devices are implemented using silicon transistors forming logic gates that consume power. Electrons that flow across the silicon substrate when charge is applied to or removed from the gate produce electromagnetic radiations.

Author is a graduate student at the Department of Electrical & Computer Engineering, Oregon State University, Corvallis, Oregon 97331. E-mail: yerubasi@engr.orst.edu

Among the entire side channel information that is leaked the most dangerous one and the most difficult one to control are the power measurements. All the calculations that are performed on the smart cards are based logic 1 and 0. This makes it more vulnerable because the power consumption is way different for 1 and 0. Attacker monitors these differences in power to get the side channel information. Of the power attacks the *Simple Power Analysis*(SPA) and *Differential Power Analysis*(DPA) are the most dangerous. Differential Power Analysis is statistical way to approach the monitoring of the power analysis.

III. Power Dissipation.

Figure 1 shows a simple lumped component that is used to get the power dissipation measurements. The ground pin of the smart card easily monitors power dissipated by a smart card. A small resistor is placed in series with V_{ss} pin of the card and the ground. The current passing though the resistor creates a time varying voltage depending on the operation the smart card reader is performing. This is recorded using a digital oscilloscope. The current flows through the out of the smart card through a bond wire. The values of the L_{bond} and the capacitor will determine the shape of the power signal.[1]

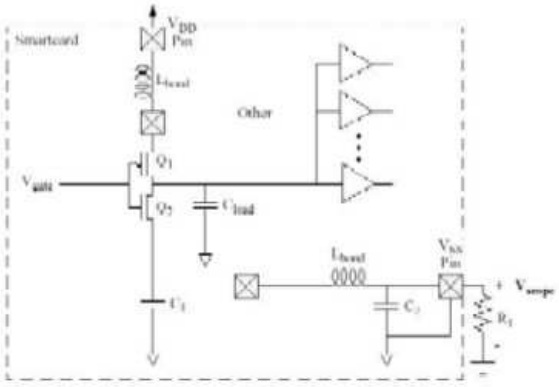

Figure 1: Measuring Power Consumption of Smart Card.

IV. Simple Power Analysis.

In Simple Power Analysis attack, the attacker makes use of the power that is consumed and radiated. This power

consumption and radiation will leak the information on the secret key and plain text because instructions that are executed depend on the values that are being processed. Temporal power variations are caused when the data is being stored and retrieved from the registers or during the computations. The way or the sequence of instructions are executed can be detected by SPA. SPA can be used to break the cryptographic devices for data dependent implementations. SPA can be applied on to any encryption algorithm.

To do all this i.e., to measure the power consumption we need a small resistor, which is inserted in series with the power, or the ground input. The voltage difference across the resistor, divided by the resistance gives current.[2]

A trace refers to a set of power consumption measurements taken across a cryptographic operation. For example, a 1-millisecond operation sampled at 5MHz yields a trace containing 5000 points. The Figure below shows an SPA trace from a smart card performing a DES operation. All the 16 DES rounds are clearly visible in the figure.

Figure 2 gives a more detailed view of the same trace showing the second and third rounds of a DES encryption operation. Many details of the DES operation are visible. For example, the 28-bit DES key registers C and D are rotated once in round 2 (left arrow) and twice in round 3 (right arrows). In this figure, small variations between the rounds just can be perceived. Many of these discernable features are SPA weaknesses caused by conditional jumps based on key bits and computational intermediates [3].

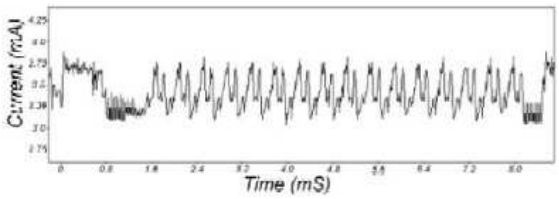

Figure 2: SPA trace of DES Operation.

Figure 3. shows even higher resolution views of the trace showing power consumption through two regions, each of seven clock cycles at 3.5714MHz. The visible variations between clock cycles result primarily from differences in the power consumption of different microprocessor instructions. The upper trace in the below figure shows the execution path through an SPA feature where a jump instruction is performed, and the lower trace shows a case where the jump is not taken. The point of divergence is at clock cycle 6 and is clearly visible. [3].

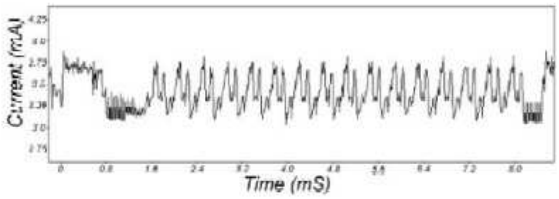

Figure 3: SPA trace of DES Rounds 2 and 3.

SPA can reveal the sequence of instructions executed so it is used to break cryptographic implementations in which the execution path depends on the data being processed.[3]

V. Preventing SPA.

To prevent Simple Power Analysis is also simple and easy to implement. Try to avoid the procedures that are using secret intermediate keys and conditional branches. This will help prevent in masking many SPA characteristics. The attacker to get the Secret Key using SPA also uses constant Execution path. And most (but not all) hard-wired hardware implementations of symmetric cryptographic algorithms have sufficiently small power consumption variations that SPA does not yield key material [3].

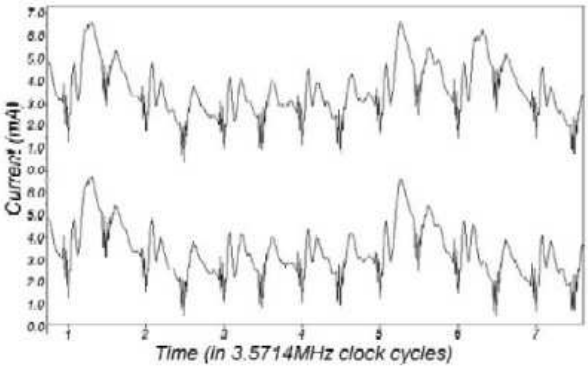

Figure 4: SPA trace individual clock cycles.

VI. Differential power analysis.

Differential Power Analysis is more powerful than SPA. When the attacker uses DPA he does not need any detailed knowledge of how the encryption algorithm was implemented. This technique is also more powerful because it uses statistical analysis to get the side channel information. Differential Power Analysis (DPA) is more effective at grabbing information leaks at low signals than SPA. Unlike SPA more sensitive equipment is required for higher sampling rates. DPA analysis uses power consumption measurements to evaluate correlations between power signatures found when an internal value is correctly determined by the DPA selection function.[4]

The basic principle of DPA is that the probability distribution of the power consumption, given that a certain event occurs, can be distinguished from the average probability distribution.

To start with DPA first we need to get the *Power Consumption Curves*(PCCs)of the device before starting the DPA. This data collected is then used calculate the order of the differential curve. In the case of DES DPA allows to determine the key bits six by six by targeting the output of one S-box at a time. The power consumption curves are then grouped. They are grouped according to the output. The group of the curves are added if the bit is a one and will be subtracted if the bit is a 0. If the six bits used to plot the differential trace are correct a spike is generated or else the wave looks as if it is generated like a noise.[5] The Differential trace is calculated as:

$$\triangle_D[j] = \frac{\Sigma_{i=1}^{N} D(P_i,K_s)T_i[j]}{\Sigma_{i=1}^{N} D(P_i,K_s)} - \frac{\Sigma_{i=1}^{N} (1-D(P_i,K_s))T_i[j]}{\Sigma_{i=1}^{N} (1-D(P_i,K_s))} = \epsilon_1 - \epsilon_0$$

Where K_s are the six unknown key bits, P_i the i-th known plain bit, $D(P_i,K_s)$ the selection function, $T_i[j]$ the j th sample of the PCC and $\triangle_D$[j] the j th element of the differential trace.

The number of PCCs necessary to perform the attack heavily depend upon measurement conditions such as the lower the noise the less number of curves are necessary.

$$\epsilon_1 - \epsilon_0 > \sigma/\sqrt{N}$$

where ϵ represents the noise and N the number of PCCs required.

VII. DPA Attack.

A. Step 1: Data Acquisition Phase

Instruct the cryptographic device to perform a number of cipher computations. For each of these compactions power consumption pattern is P_i is measured and stored together with the pattern of computations i.e., a_i the plain text or cipher text. This is called the data acquisition phase which produces a data set $D = (a_i, P_i) \mid i = 1...z$

B. Step 2: Target Sub Key

In this step specify an event whose occurrence depends either on the value of a number of plain text or cipher text(any one) bits and key bits. The result of an intermediate cipher calculation gives a result, which is at some time present in a CPU register, ALU, bus, or memory cell. This is called target subkey the key bits the specified event depends on.

C. Step 3:

The following check is performed for all the possible values of the target subkey. Assuming that the target subkey s* is correct, Power consumption patterns are divided into two groups: first one for those where the event occurred is $D_1 = (a_i, P_i) \mid f(s^*, a_i) = 1$ and the second, its complementary set $D_0 = (a_i, P_i) \mid f(s^*, a_i) = 0$. f indicates whether the event occurs given the known and hypothesis values[6].

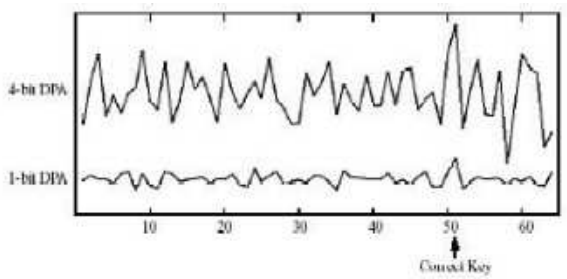

Figure 5: DPA bias for DES key 0123456789abcdef.[1]

The two subsets D_1 and D_0 are assumed to be statistically distinguished for the correct hypothesis. Some distance between the two distributions is defined. The subkey value for which we get the maximum distance is considered correct value. A wrong target subkey value will divide the power consumption patterns in two sets in which the event occurs an average number of times. If the round function has certain algebraic properties, several subkey values, among which the correct one may be suggested. [6]

VIII. Counter Measures

There are very few was we can prevent the DPA attacks. They can be categorized into three groups. First thing we can do is reduce signal size, by using a constant execution path code as well as choosing operations which leak little information during their power consumption or it can also be done by adding extra gates to compensate for the power consumption. But such signal size reduction cannot reduce the signal size to zero so the attacker will still have an infinite number of samples with which he can go ahead with the attack on the signal.

Second thing we can do is to introduce noise into power consumption measurements but still attacker has infinite number of samples and can still be able to do the statistical analysis. In addition to this execution timing and order can be randomized [3]. Designers must approach temporal obfuscation with great caution because many of these techniques will be used to bypass or compensate for these effects.

As a third approach we can start by using non-linear key update procedures. For example, hashing a 160-bit key with SHA should effectively lose all partial information an attacker might have gathered about the key. Similarly, aggressive use of exponent and modulus modification processes in public key schemes can be used to prevent attackers from gathering data across large numbers of operations. Key use counters can prevent attackers from obtaining large numbers of samples.

A. Random Process Interrupts

One of the most common counter measures against DPA can be done by the introduction of *Random Process Interrupts* RPIs. Instead of executing all the operations sequentially the CPU interleaves codes execution with that of dummy instructions so that the corresponding opera-

tion cycles do not match because of the time shifts. This has the effects of smearing the peaks across the differential trace. This is due to desynchronizing effect. It is also called "*incoherent averaging*". The time shifts can be considered as added noise RPIs do no make the attack theoretically infeasible but increases N.

RPIs are assumed to occur at a constant probability p. Though the spike is seen in the differential trace as a result of correct guess, this spike makes the attacker confused because the noise is spread over consecutive cycles.

Usually if spike is to be seen at n cycles a spike will again appear after again n+C_n cycles if RPI occurs where delay $C_n = \sum_{i=1}^{n} C_i$, C_i being the i th cycle, with c_1 =1; Then mean position of the spike is given by the equation

$$\mu =< C_n + n >= \sum_{i=1}^{n} < c_i > +n = np + n,$$

and the variance is given by

$$\nu =< {C_n}^2 > - < C_n >^2 = \sum_{i=1}^{n} Var(c_i) = n(p-1)p \cong np.$$

The spike will be distributed over a range $\pm\delta$ and this is centered around μ. The spike is distributed over a range for k = $2\delta \cong 2np$ consecutive cycles.

So by seeing all this we can say that spike is visible only if

$$\frac{\varepsilon_1 - \varepsilon_0}{k} > \frac{\sigma}{\sqrt{N'}}$$

Comparing the above two equations we see that the number of RPI-protected PCCs necessary to put the DPA back into action is given by[5]

$$\text{N'} = k^2 \text{ N}.$$

IX. MY BIG IDEA

This part of the paper gives the best way of counter measure that can be used to prevent the Differential Power Attacks. The graphs show that the attacker cannot determine anything by looking at the power consumption graphs. In my method I go ahead and combine all the three approaches mentioned earlier. Adding noise as well with the non-linear key update. Here in this method even the condition of spike reconstruction using integration is also not possible. Even the original amplitude, which can be restored by integrating the RPI-protected signal over a given number of cycles is not a success for the attacker. Though the attacker tries both the following steps i.e., obtaining the differential curve and then using high number of PCC's he still can't guess the because there are no spikes in the PCC. As the next approach even if the attacker tries to determine the key by classical DPA just by observing only one out of four the S-box output bits. In this case some output bits leak more amount of information than the others. So attacker tries to perform the DPA and get which bits yield better spikes for the correct key guess. Even by Hamming integration it is not possible for him to calculate the bits for the spike because of the extra noise signal inserted. By the method i designed it is not possible for the attacker to guess the right key in any case. I am showing the power consumption of the smart card using my big idea. But still there are a few changes that are pending.

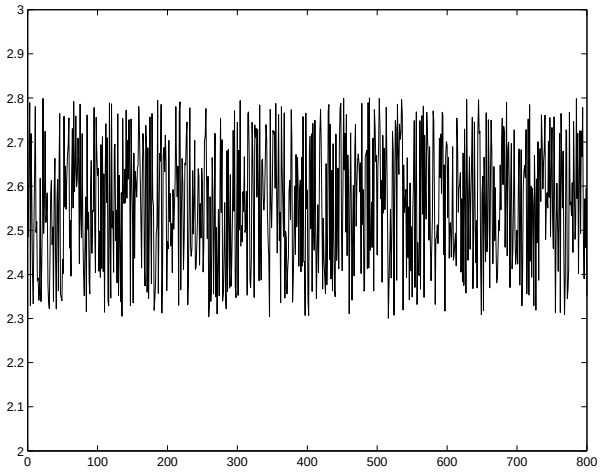

Figure 6: Power Consumption of Smart card After My Big idea Implemanation

References

[1] Ezzy A. Dabbish Thomas S. Messerges and Robert H. Sloan, "Investigations of power analysis attacks on smartcards," in *USENIX Workshop on Smartcard Technology*, May 1999.

[2] Jameco Electronics, "Pc - multiscope(part #142834) febraury 1999 catalog,," p. 103.

[3] Joshua Jaffe Paul C. Kocher and Benjamin Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO'99*, Micheal Wiener, Ed. 1999, pp. 388–397, Springer, Berlin, Germany.

[4] Tom Lash, "A study of power analysis and the advanced encryption standard recomendations for designing power analysis resistant devices," George Mason University, Feb 2002.

[5] Nora Dabbous Christophe Clavier, Jean-Sbastien Coron, "Differential power analysis in the presence of hardware countermeasures," in *Cryptographic Hardware and Embedded Systems — CHES'00*, Ç. K. Koç and C. Paar, Ed. 2000, Lecture Notes in Computer Science, No. 1965, pp. 252–263, Springer, Berlin, Germany.

[6] Michael Peters Joan Daemen and Gilles Van Assche, "Bitslice ciphers and power analysis attacks," in *Fast Software Encryption Workshop 2000*, Bruce Schneier, Ed. 2000, pp. 134–149, Springer, Berlin, Germany.