Differential Power Analysis in the Presence of Hardware Countermeasures

[Published in Ç.K. Koç and C. Paar, Eds., *Cryptographic Hardware and Embedded Systems – CHES 2000*, vol. 1965 of *Lecture Notes in Computer Science*, pp. 252–263, Springer-Verlag, 2000.]

Christophe Clavier[1], Jean-Sébastien Coron[2], and Nora Dabbous[2]

[1] Gemplus Card International
Avenue du Pic de Bretagne, B.P. 100, 13881 Gémenos, France
christophe.clavier@gemplus.com
[2] Gemplus Card International
34 rue Guynemer, 92447 Issy-les-Moulineaux, France
{jean-sebastien.coron, nora.dabbous}@gemplus.com

Abstract. The silicon industry has lately been focusing on side channel attacks, that is attacks that exploit information that leaks from the physical devices. Although different countermeasures to thwart these attacks have been proposed and implemented in general, such protections do not make attacks infeasible, but increase the attacker's experimental (data acquisition) and computational (data processing) workload beyond reasonable limits.
This paper examines different ways to attack devices featuring random process interrupts and noisy power consumption.

Keywords. Power analysis, smart card, hardware countermeasure, random process interrupt.

1 Introduction

In past decades, cryptanalysis focused on exploiting mathematical weaknesses in algorithms to break into the targeted systems. As a result, modern cryptosystems are generally designed to better withstand logical threats and attackers are concentrating on analysis of side channel leakage. Among these, timing attacks, Simple Power Attacks (SPAs), Differential Power Attacks (DPAs) and TEMPEST are certainly best known [7, 8, 1].

Most of the time, the cryptographic kernels of products used are not isolated in perfectly tamper-proof locations. It has long been known that execution time, power consumption, radio frequencies, magnetic field values, etc. could leak some information on sensitive data. After a first glance, cryptographers had concluded that these would only be able to reveal partial information, therefore not causing a real danger. It was only in 1996 that Paul Kocher demonstrated that side

channel attacks were effective enough to recover secret keys in numerous cryptosystems. Differences in execution time were the first to be exploited [8] and in 1999 it was shown that power consumption measurements, if carefully analyzed, could also reveal sensitive information [7]. Now that these pitfalls have been uncovered, analyzed and better understood, different countermeasures are being studied in order to minimize the side channel attacks' impact by reducing the signals that can be exploited to perform these attacks, or making them useless.

Following a more or less uniform reaction pattern, most manufacturers came-up with software and hardware means to hide and randomize sensitive data. This paper focuses on DPA on systems in which hardware countermeasures have been implemented. The experiments described below were successfully carried out on DES, proving that the some countermeasures, initially thought to be heuristicly sufficient, do not guarantee the claimed security level.

Section 2 briefly recalls DPA and explains how to perform the attack on devices featuring *random process interrupts* (RPIs) and noisy power consumption. Section 3 focuses on a first method to eliminate the chip's hardware protection. Section 4 improves this method, as long as the guidelines in section 5 are taken into account.

2 DPA in the Presence of Random Process Interrupts

Power attacks isolate information correlated to operations or manipulated data by examining devices' power consumption. Following Kocher's terminology [7], Simple Power Analysis (SPA) consists in directly analyzing a device's power consumption, whereas Differential Power Analysis (DPA) spots correlation between the data being manipulated and the side channel information.

2.1 Differential Power Attacks

DPA can be easily performed on the first DES round if the plaintext is available or on the last DES round if the ciphertext is known. We will recall the basic DPA attack on DES round one. Given the plaintext and the round subkey, the attacker can calculate the input to the S-box functions and, by table look-up, their output. As is, DPA on DES is performed on one S-box at a time and allows to determine key bits six by six by targeting the output of one S-box. To perform a DPA, different *power consumption curves* (PCCs) of the device must first be collected. In the basic attack, PCCs are grouped according to one among the 4 S-box output bits observed. If the bit is a 1 the power values are added, if it's a 0 they are subtracted to calculate a *differential curve*. An attacker supposedly has no information on the key so, when performing DPA, he must calculate 64 differential traces, one for each of the 2^6 6-bit partial subkey combinations. A spike will appear in the differential curve that was plotted by using the correct subkey bits where the selection function is correlated to the value of the bit being manipulated. The trace will only feature moderate noise (in this model correlations between different key values are neglected) in the

other 63 differential curves obtained with incorrect subkey bits. Theoretically, any bit among the 4 S-box output bits could be analyzed to classify the PCCs. A differential trace obtained by analyzing the other bits could be calculated to confirm the results. More on this will be said in section 4.

As the goal of this research was to test the effectiveness of hardware countermeasures, we executed a DPA plaintext attack. The implementation of a DPA ciphertext attack is straightforward, instead of making assumptions on S-box inputs and analyzing the outputs, the inverse approach would be followed. Attackers most probably have knowledge of the ciphertext rather than the plaintext and therefore would run an attack on DES round 16. All results reported in this paper are valid for DPA ciphertext attacks as well.

As explained in [7], the differential trace is calculated as:

$$\Delta_D[j] = \frac{\sum_{i=1}^{N} D(P_i,K_s)T_i[j]}{\sum_{i=1}^{N} D(P_i,K_s)} - \frac{\sum_{i=1}^{N}(1-D(P_i,K_s))T_i[j]}{\sum_{i=1}^{N}(1-D(P_i,K_s))} = \varepsilon_1 - \varepsilon_0$$

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

The number of PCCs necessary to perform the attack heavily depends on the measurement conditions: the lower the noise, the fewer curves are necessary. We refer the reader to [4] for several useful guidelines. For the spike to be identified

$$\varepsilon_1 - \varepsilon_0 > \sigma/\sqrt{N}, \tag{1}$$

must hold, where σ represents the noise and N the number of necessary PCCs.

Better acquisition equipment and higher sampling rates yield lower noise. Although chip-dependent, table 1 gives a rough idea about the required N as a function of the acquisition experiment's sampling rate S expressed in MHz for a card running at 3.68 Mhz. The card used to obtain such values featured no countermeasures designed to thwart DPA attacks.

N	600	500	120	100
S	50	100	500	1000

Table 1. N as a function of the attacker's equipment sampling rate.

2.2 Random Process Interrupts

One of the most common countermeasures against DPA is the introduction of *random process interrupts* (RPIs). Instead of executing all the operations sequentially, the CPU interleaves the code's execution with that of dummy instructions so that corresponding operation cycles do not match because of time

shifts. This has the effect of smearing the peaks across the differential trace due to a desyncronisation effect, known in digital signal processing under the name of *incoherent averaging* [9]. The time shifts can be considered as added noise. Needless to say, RPIs do not make the attack theoretically infeasible but increase N considerably.

Assuming that RPIs occur with a constant probability p, even if a spike should be seen on the differential trace because the correct key was guessed, the spike might remain confused with the noise because it was spread over consecutive cycles. Due to RPIs, the spike that actually appears follows a gaussian distribution, thoroughly characterized by a mean position μ and a variance v that can be precisely calculated.

Suppose the spike on the differential trace should be seen after n cycles. If RPIs occurred, a spike will appear after $n + C_n$ cycles, where the delay $C_n = \sum_{i=1}^{n} c_i$, c_i being the i-th cycle, with $c_i = 1$ if an RPI occurred and $c_i = 0$ if not.

The mean position for the spike is:

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

and the variance v is:

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

We can thus estimate the standard deviation[1] $\delta \cong \sqrt{np}$, which means that (for all experimental purposes) the spike will be distributed over a $\pm\delta$ range centered around μ. In other words, we consider that the spike was distributed over $k = 2\delta \cong 2\sqrt{np}$ consecutive cycles. The spike will thus be visible if:

$$\frac{(\varepsilon_1 - \varepsilon_0)}{k} > \frac{\sigma}{\sqrt{N'}}, \tag{2}$$

We therefore infer from (2) and (1) that the number of RPI-protected PCCs necessary to put the DPA back on it's feet is:

$$N' = k^2 N.$$

As a characteristic example, if the DPA spike should be seen after $n = 1600$ cycles (which can typically be the case for a spike observed after the first DES round) then $p = 12\%$ yields $k \cong 28$. This means that the number of RPI-protected PCCs necessary to re-run the same attack must be multiplied by a factor of 784. For research purposes this attack was indeed successfully performed, but in reality such an attack is improbable because of the number of PCCs that should be acquired. In the next section, we will describe a method that allows to consistently reduce the number of PCCs necessary to run the attack.

[1] We intentionally use δ instead of the letter σ that we reserve for further use.

3 Spike Re-construction by Integration

The spike's amplitude $(\varepsilon_1 - \varepsilon_0)$ can be re-constructed in order to decrease the number of power consumption curves needed. Because of desynchronization, the spike's amplitude is divided by a value bound by k, but it's original amplitude can be restored by integrating the RPI-protected signal over k consecutive cycles. The new signal value is $k \times \frac{(\varepsilon_1 - \varepsilon_0)}{k}$, the new noise value is $\frac{\sigma}{\sqrt{N''}} \times \sqrt{k}$. Consequently the spike will be visible if

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

Therefore, to restore the same signal to noise ratio as in (1), the following equation must be satisfied:

$$N'' = kN.$$

As this method implies integrating PCC values on k consecutive cycles, we called it *Sliding Window DPA* (SW-DPA). Implementing SW-DPA involves two steps. First of all, a classic (Kocher-style) differential curve must be obtained. Unless a very high number of PCCs is used, even for the correct key guess no spike will appear because of the RPIs. For the spikes to appear, RPI-protected PCCs must be integrated. This step consists of adding points on k consecutive cycles from the differential PCC obtained in step one. To visualize this operation, the reader may imagine a comb with k teeth, each corresponding to a point on the differential PCC created in step one. The distance between two consecutive teeth on the comb must match the number of time samples separating two consecutive cycles. Integration is obtained by adding the power value of the points indicated by the comb.

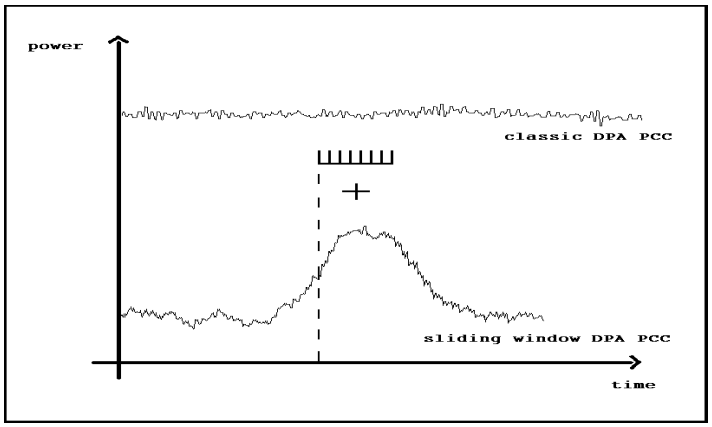

Fig. 1. The integration operation.

If the same figures as before are considered (that is $k = 28$) the attack can be implemented by increasing by a factor of 28 the number of necessary curves. If a DPA can be performed using 120 unprotected power consumption curves acquired at 500 MHz, then only 3360 RPI-protected curves would be necessary.

Figure 2 shows real-life differential curves obtained with an integration window of 30 for a right (upper curve) and a wrong (lower curve) key guess.

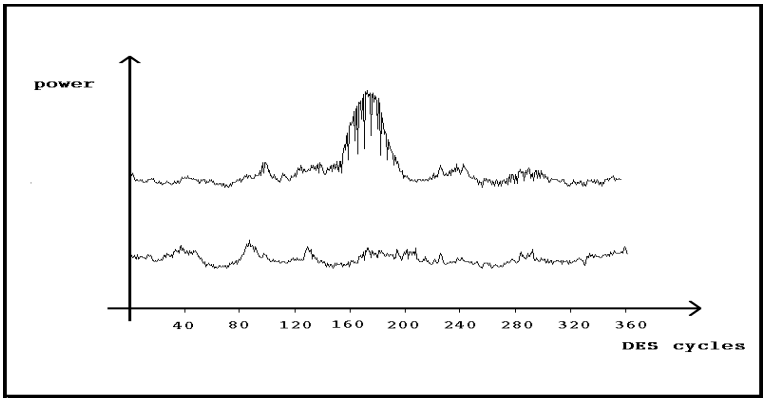

Fig. 2. Differential trace for correct and wrong key guesses.

We'd like to clarify that all cycle number values are reported to show the reader how wide the DPA spike is. They are not intended as an absolute reference from the beginning of the DES execution, as this number greatly depends on the way the function is implemented.

Figure 3 shows a zoomed in view of the differential trace spike for the correct key. It can clearly be seen that what looks like a single spike in the larger view is made up of different "spike portions". This is the integration's effect, which adds up the fractions of all distributed spikes only when accurately centered.

4 The Hamming Integration Variant

When determining the key by classic DPA, PCCs are classified by observing only one out of the four S-box output bits. Experimentally we obtained 4 differential traces per S-box, each one by examining a different S-box output bit, and noticed that some output bits leak more information than others. To successfully perform a DPA, an attacker could predetermine which bits yield better spikes for correct key guesses. Our approach, though, was to take advantage of the information gathered from all 4 S-box output bits simultaneously.

Let us assume that the chip's power consumption is proportional to the output's *Hamming weight*. If only one S-box output bit is observed and PCCs are classified according to this bit's value, the spike's amplitude will be proportional

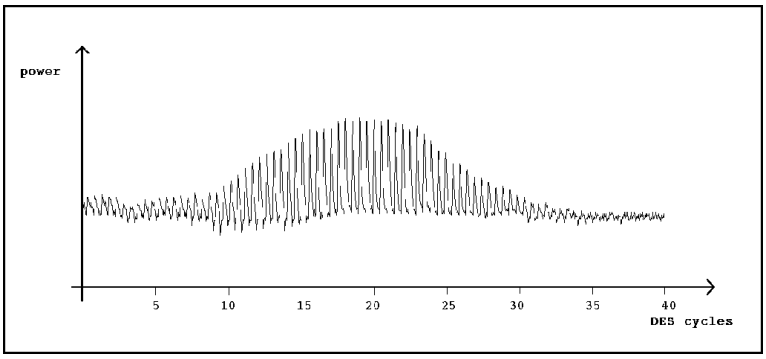

Fig. 3. Enlarged view of the right guess spike.

to:

$$< H >_{s_i=1} - < H >_{s_i=0} = (1 + \frac{3}{2}) - (0 + \frac{3}{2}) = 1,$$

where H is the power consumption at a certain instant and s_i is the i-th S-box output bit. In particular, we set $< H >_{s_i=1} = (1 + \frac{3}{2})$ because we are considering S-box output bits for which one bit is certainly equal to one, whereas the remaining three bits are equal to one with probability one half.

The signal to noise ratio is equal to:

$$\text{SNR} = \frac{1}{\frac{\sigma}{\sqrt{N}}},$$

where σ is the differential curve's standard deviation and N the number of curves considered.

If, instead, all 4 S-box output bits are observed simultaneously, a new PCC classification criterion must be designed. Curves could be classified according to the total S-box Hamming weight, that is curves for which four or three ones appear could be grouped in one class while, on the other hand, curves for which zero or a single one are shown could be grouped in a second class, and curves with two ones and two zeros in the output could be discarded. In this case the spike's amplitude would be proportional to:

$$< H >_{b=4,3} - < H >_{b=0,1} = \frac{16}{5} - \frac{4}{5} = \frac{12}{5} = 2.4, \tag{3}$$

where b is the number of ones. In particular, $< H >_{b=4,3} = \frac{16}{5}$ is the Hamming weight mean when three or four ones appear in all possible combinations of four bit strings.

We call this method the Hamming integration variant. This, combined with an integration in case RPIs had been inserted, yields a much higher spike.

The signal to noise ratio using this method is equal to:

$$\mathrm{SNR} = \frac{\frac{12}{5}}{\frac{\sigma}{\sqrt{\frac{10}{16} \times N}}} = 1.9 \times \frac{1}{\frac{\sigma}{\sqrt{N}}},$$

where σ is the differential curve's standard deviation and $10N/16$ is the number of curves considered. In the above calculation we take into account the fact that $6N/16$ curves are discarded because a PCCs is not processed whenever two ones (and two zeros) appear as S-box output.

The 1.9 SNR between the two different methods is a theoretical result. As stated before, some output bits leak more information than others and the experimental ratio between the observed spikes greatly depends on the targeted output bit. This can clearly be seen on figure 4, for which the curves were obtained by examining two different S-box output bits for the first DES S-box. The upper curve was obtained by SW-DPA on the first S-box output bit, whereas the lower curve was obtained by SW-DPA on the fourth one.

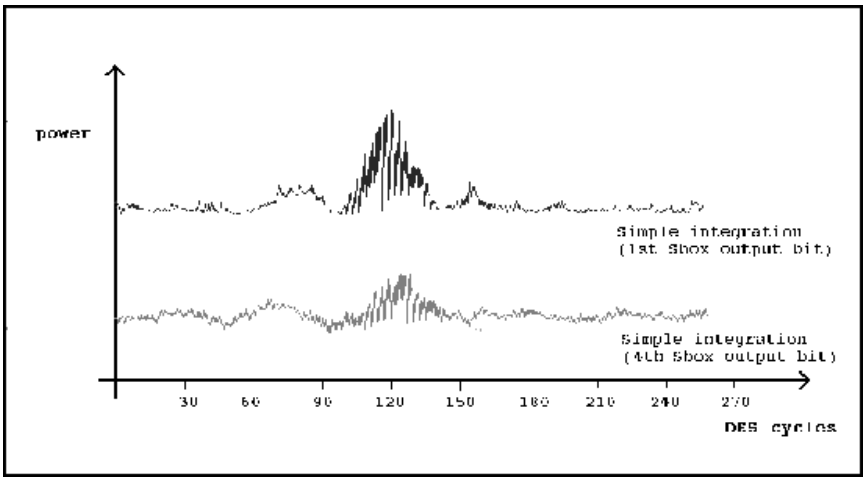

Fig. 4. Differential trace obtained on first S-box applying SW-DPA on the first output bit (upper curve) and applying SW-DPA on the fourth output bit (lower curve).

5 Redefining the Selection Function

A very strong correlation exists between the chip's power consumption and the operation being executed. This value is quite high during data transfer between the CPU and the external RAM, so this operation, performed after an S-box output is determined, is usually targeted for DPA. The assumption made to

create and interpret the differential trace curve is that the power consumption is different when the S-box output bit is a 0 or a 1. Power consumption, though, does not *only* depend on the output value, but also on the transitions that occur on the bus (*c.f.* to [4] for instance). Assuming ordinary CMOS inverter implementation, a high power consumption is to be expected when a 1 is being written onto a bus line previously discharged, or when a 0 is being written onto a bus line previously charged. Values in these two cases are, of course, not the same but, as this difference is not essential for the purpose of this study, it has been neglected hereafter.

The differential trace obtained by classic DPA will show a spike for the correct key even if the bus' status is not taken into account: the two power consumption groups in which curves are classified will still contain the same elements and at most an error on the spike's sign will be made (but this is irrelevant for the attack's purpose). On the other hand, the bus' status must be considered when observing simultaneously all four output bits for otherwise information could be lost.

When applying the Hamming variant, the power consumption of four bus lines is simultaneously analyzed. Values 0 to 15, corresponding to all possible combinations of ones and zeros on the four bus lines, could have been represented previously. To reduce the number of possibilities that must be studied, only values from 0 to 7 can be considered, as in our simplified model we postulate that power consumption due to transitions from 1 to 0 or from 0 to 1 are equivalent.

Let us erroneously classify PCCs according to the S-box output bits, neglecting the bus line's previous state. Two groups will result:

High Hamming Weight	Low Hamming Weight
1111	0000
1110	0001
1101	0010
1011	0100
0111	1000

Table 2. Power consumption curve classification according to S-box output.

For the correct guess, we expect a spike amplitude proportional to 2.4 (3).

The bus lines on which a transition occur are the ones for which $S_i \oplus B_i = 1$, where S_i is the i-th S-box output bit and B_i is the i-th bus line. Therefore, to correctly group curves according to power consumption, they should be classified according to the number of ones that result from $S_i \oplus B_i$.

Let us suppose that the value previously on the bus was 0011. If this information is neglected, the classification will yield what is reported in table 3.

From columns 3 and 6 in the table 3, having neglected the value previously on the bus,we infer that the spike's height will be proportional to:

$$< H >_{b=4,3} - < H >_{b=0,1} = \frac{10}{5} - \frac{10}{5} = 0,$$

High Hamming Weight			Low Hamming Weight		
S-box	bus	S-box $\oplus$ bus	S-box	bus	S-box $\oplus$ bus
1111	0011	1100	0000	0011	0011
1110	0011	1101	0001	0011	0010
1101	0011	1110	0010	0011	0001
1011	0011	1000	0100	0011	0111
0111	0011	0100	1000	0011	1011

Table 3. Power consumption curve classification having neglected the previous value on the bus.

therefore all useful information is lost.

However, in case the value previously present on the bus is 1000, or any other configuration in which three ones, or three zeros, are present, the spike's height would be proportional to:

$$< H >_{b=4,3} - < H >_{b=0,1} = \frac{13}{5} - \frac{7}{5} = \frac{6}{5},$$

therefore some, but not all, useful information is lost.

Only if the value on the bus had previously been 0000, or equivalently 1111, even by neglecting this value the spike's height would be proportional to 2.4.

If the previous bus state is unknown, in order to find the correct key guess by applying the Hamming integration, the attack must be run for all 8 possibilities. For the correct key, the following is observed:

- one differential curve with a high spike for the correct previous value on the bus
- four differential curves with a medium spike for a mistake on one bit (or three bits) on the previous value on the bus
- three flat differential curves for a mistake on two bits on the previous value on the bus

To be able to perform this attack, the state of the bus line before the instruction targeted by DPA must be constant or else a correct power curve classification cannot be performed. This value is constant when the previous operation concerning the bus is an opcode loading, or when data, constant but not dependent on the source code, transits on the bus.

Figure 5 shows a differential trace obtained by Hamming integration knowing the previous value on the bus compared to one resulting from SW-DPA on the S-box output bit that leaks the most.

6 Conclusions

This paper shows that DPA can still be applied to chips on which hardware measures thought to provide DPA resistance had been implemented. The first attack

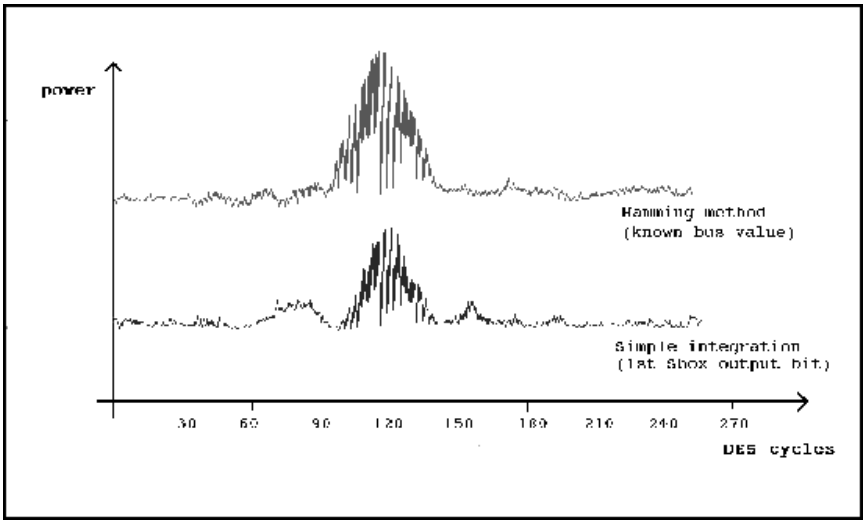

Fig. 5. Differential trace obtained on first S-box applying Hamming integration (upper curve) and applying SW-DPA on the first output bit (lower curve).

proposed consists in applying a sliding window to the classic DPA described in [7]. The loss of synchronization caused by RPIs and the consequent increased number of necessary PCCs to perform the attack is calculated. The Hamming integration variant is slightly more complicated because, since 4 output bits are considered simultaneously, the previous value on the correspondent bus lines must be determined. As this information is usually unknown to the attacker, all possibilities must be examined experimentally. The advantage of the variant is a higher SNR, or sucess of the attack with a reduced number of PCC. As the second method involves a greater computational cost, it could be applied only to a restricted number of probable secret keys when the first method leaves some doubt.

To be secure, cryptographic devices should incorporate both hardware and software countermeasures to decrease the feasibility of side channel attacks. It is also important to prove the validity of the countermeasures implemented, as heuristic assumptions are often not enough.

7 Acknowledgements

We would like to thank David Naccache for all his invaluable suggestions. We would also like to thank Olivier Benoit and Pascal Moitrel for the experimental results. Finally we would like to thank an anonymous referee for his detailed analysis of the paper which helped us provide the necessary improvements.

References

1. R. Anderson, M. Kuhn, *Tamper resistance – a cautionary note*, The second USENIX workshop on electronic commerce, pp. 1-11, 1996.
2. S. Chari, C. S. Jutla, J. R. Rao, P. Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, Springer-Verlag, LNCS 1666, pp. 398–411, Crypto'99, 1999.
3. J.-S Coron, *Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems*, Springer-Verlag, LNCS 1717, pp. 292–302, CHES'99, 1999.
4. J.-S. Coron, P. Kocher, D. Naccache, Statistics and Secret Leakage, FC'00, to appear in the LNCS series, 2000.
5. P. N. Fahn, P. K. Pearson, *IPA: A New Class of Power Attacks*, Springer-Verlag, LNCS 1717, pp. 173–186, CHES'99, 1999.
6. L. Goubin, J. Patarin, *DES and Differential Power Analysis*, Springer-Verlag, LNCS 1717, pp. 158–172, CHES'99, 1999.
7. P. Kocher *Differential Power Analysis*, Springer-Verlag, LNCS 1666, pp. 388–397, Crypto'99, 1999.
8. P. Kocher *Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and Other Systems*, Springer-Verlag, LNCS 1109, pp. 104–113, Crypto'96, 1996.
9. R. G. Lyons, *Understanding Digital Signal Processing*, Addison Wesley Longman, pp. 327–330, 1997.