Leakage Power Analysis Attacks: A Novel Class of Attacks to Nanometer Cryptographic Circuits

Massimo Alioto, *Senior Member, IEEE*, Luca Giancane, *Student Member, IEEE*, Giuseppe Scotti, and Alessandro Trifiletti

***Abstract*—In this paper, a novel class of power analysis attacks to cryptographic circuits is presented. These attacks aim at recovering the secret key of a cryptographic core from measurements of its static (leakage) power. These attacks exploit the dependence of the leakage current of CMOS integrated circuits on their inputs (including the secret key of the cryptographic algorithm that they implement), as opposite to traditional power analysis attacks that are focused on the dynamic power. For this reason, this novel class of attacks is named "Leakage Power Analysis" (LPA). Since the leakage power increases much faster than the dynamic power at each new technology generation, LPA attacks are a serious threat to the information security of cryptographic circuits in sub-100-nm technologies. For the first time in the literature, a well-defined procedure to perform LPA attacks that is based on a solid theoretical background is presented. Advantages and measurement issues are also analyzed in comparison with traditional power analysis attacks based on dynamic power measurements. Examples are provided for various circuits, and an experimental attack to a register is performed for the first time. An analytical model of the LPA attack result is also provided to better understand the effectiveness of this technique. The impact of technology scaling is explicitly addressed by means of a simple analytical model and Monte Carlo simulations. Simulations on a 65- and 90-nm technology and experimental results are presented to justify the assumptions and validate the leakage power models that are adopted.**

***Index Terms*—Cryptographic circuits, information security, leakage, power analysis, side-channel attacks, Smart Cards.**

I. Introduction

POWER ANALYSIS attacks have been extensively shown to be a major threat to the security of data that are processed and stored in cryptographic devices, such as Smart Cards [1]–[4]. These attacks exploit the dependence of the dynamic power consumption on the inputs of a cryptographic algorithm, i.e., the input ciphertext (plaintext) that is to be decrypted (encrypted) and the secret key. The cost in terms of equipment and computational effort are rather low; hence, these attacks can be easily performed [4].

In power analysis attacks, a known input sequence is applied, and the instantaneous power dissipated during a decryption (encryption) of each input is measured and stored. Then, postprocessing techniques are applied to the power traces to recover the secret key that is internally stored by the cryptographic circuit and is used during the decryption (encryption) phase. Among the existing postprocessing techniques, the Correlation Power Analysis (CPA) is well known to be relatively simple and effective [4], [5]. In CPA, a power model is adopted to estimate the dynamic power consumption required to physically evaluate a signal generated within the crypto chip as a function of the input and the secret key [5]. Then, a portion of the secret key is guessed, and the resulting dynamic power consumption is estimated with this model. Successively, the correlation coefficient between this estimation and the measured power is evaluated. Finally, the correct key is identified by taking the key guess that leads to the highest value of the correlation coefficient [4], [5]: Indeed, the closer is the guessed key to the actual key, the greater is the correlation between the estimated and measured power.

In sub-100-nm technologies, the dynamic power is no longer the dominant contribution to the chip power budget, due to the much faster increase of leakage (i.e., static) power at each technology generation [6]–[9]. For example, at the 65-nm technology node, the leakage power is in the order of half the chip power consumption and is planned to be an even greater fraction in successive technologies by the International Technology Roadmap for Semiconductors [9]. Hence, the leakage power can be easily measured in the same way as the dynamic power is measured in traditional power analysis attacks. Due to the strong leakage dependence on the input of digital circuits [8], leakage can also provide a significant amount of information on the secret key; hence, power analysis attacks based on leakage can be devised. In the following, these attacks will be referred to as "Leakage Power Analysis" (LPA) attacks.

Until now, information-security issues related to the leakage dependence on processed data were first discussed in [10]. These considerations were then applied in [11] to simulate an attack to a simple crypto core by adopting a procedure similar to the CPA principle [5]. Successively, an analogous attack was performed in [12] by means of circuit simulations, where the Differential Power Analysis (DPA) principle was used instead of CPA [12]. However, [10]–[12] do not build any theoretical background on leakage-based attacks and do not discuss the experimental issues that are involved in real attacks. Moreover, these papers present only simulated results, without considering

Manuscript received August 28, 2008; revised December 19, 2008 and nulldate. First published March 27, 2009; current version published February 10, 2010. This paper was recommended by Associate Editor N. K. Jha.

M. Alioto is with theDipartimento di Ingegneria dell'Informazione (DII), Università di Siena, 53100 Siena, Italy, and also with the Berkeley Wireless Research Center, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94704-1302 USA (e-mail: malioto@dii.unisi.it; alioto@eecs.berkeleyedu).

L. Giancane, G. Scotti, and A. Trifiletti are with the Dipartimento di Ingegneria Elettronica (DIE), Università di Roma "La Sapienza," 00184 Rome, Italy (e-mail: giancane@die.uniroma1.it; scotti@die.uniroma1.it; trifiletti@die.uniroma1.it).

Digital Object Identifier 10.1109/TCSI.2009.2019411

physical effects (e.g., process and temperature variations) that are observed in real attacks.

In this paper, for the first time in the literature, LPA attacks are formalized and analyzed from both theoretical and experimental standpoints in a systematic manner. Advantages and practical problems related to the LPA attack are discussed through comparison with traditional attacks targeting the dynamic power. Various examples, simulations, and experimental results are reported to better understand LPA attacks, as well as to validate the underlying assumptions. In particular, a practical LPA attack procedure is presented, and a closed-form model of the LPA attack result is developed to better understand the attack. The impact of technology scaling is also analyzed to evaluate the LPA effectiveness in future technologies.

Analysis shows that LPA attacks are a major threat to information security in sub-100-nm technologies. Moreover, since many countermeasures to power analysis attacks targeting the dynamic power have been proposed until now [4], LPA becomes the weak point in the information security of cryptographic circuits, if not taken into account during their design.

This paper is structured as follows. In Section II, the leakage sources in MOS devices and CMOS logic gates are reviewed, and basic hypotheses are validated with experimental and simulation results. In Section III, a practical procedure to perform LPA attacks is introduced, whereas measurement and practical issues are discussed in Section IV. Section V discusses simple models of LPA attacks, whereas practical attacks to a register and a cryptographic core are discussed in Section VI. The effect of process variations on LPA attacks are dealt with in Section VII, and conclusions are reported in Section VIII. An Appendix is added to improve the readability of this paper.

II. Review of Leakage Sources in Nanometer CMOS Logic Gates

The leakage current conducted by MOS transistors operating in the cutoff region consists of three main sources: Subthreshold, gate tunneling, and inverse junction current [13]. In current technologies, the subthreshold current is the most important leakage contribution in a MOS transistor, considering that the gate leakage is reduced with the adoption of high-κ materials, and the inverse junction current is two orders of magnitude lower than the former [14]. More specifically, the subthreshold current $I_{\text{leak,NMOS}}$ for a single NMOS transistor is given by [13], [14]

$$I_{\text{leak,NMOS}} = I_0 \frac{W}{L} \mathrm{e}^{-V_{\text{TH}}/nkT/q} \tag{1}$$

where V_{TH} is the transistor threshold voltage, I_0 and n are technology-dependent parameters, W/L is the transistor aspect ratio, T is the temperature, whereas k and q are the Boltzmann constant and the electron charge, respectively. An analogous expression holds for the PMOS transistor (with V_{TH} being the threshold-voltage magnitude).

Due to the exponential dependence in (1), the leakage current is very sensitive to temperature variations, as well as to process variations in V_{TH}. The impact of temperature will be discussed in Section II-A, where process variations will be neglected. The process variations will be successively analyzed in Section VII.

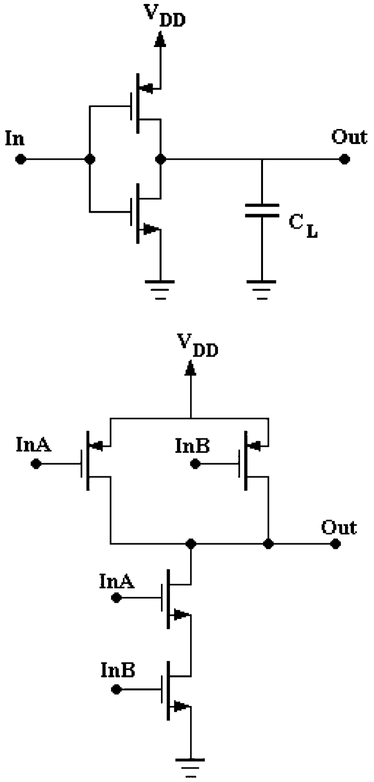

Fig. 1. (a) Schematic of the CMOS inverter. (b) Schematic of the CMOS NAND2 gate.

A. Leakage in Static CMOS Gates

The leakage current of static CMOS logic gates strongly depends on their input [8]. As an example, the leakage of an inverter gate [see Fig. 1(a)] is equal to the leakage of the NMOS (PMOS) transistor when this device is in the cutoff region, i.e., if the input is low (high). Since the threshold voltage of the NMOS and PMOS transistors are significantly different in real CMOS technologies, from (1), the inverter leakage depends on the input value. This significant leakage dependence on the input is confirmed by data in Tables I and II, which report the simulated leakage of a minimum-sized inverter gate in the 90- and 65-nm technologies, respectively. From Table I, the inverter leakage with a low input is greater than that with a high input by a factor of three to five (this is because the NMOS transistor has a lower V_{TH}, compared with the PMOS). From Table II, similar results are obtained in the 65-nm technology for realistic operating temperatures of cryptographic devices (i.e., at room temperature or slightly greater). The strong dependence on the temperature is also confirmed by Table I for the 90-nm technology, whereas this dependence is weaker for the 65-nm technology in Table II, due to the different temperature dependence of the gate leakage.

The aforementioned considerations on the leakage dependence on the input value can be extended to general static CMOS gates, whose pull-up and pull-down networks are made up of series- and parallel-connected transistors. To understand the leakage dependence on the input, it is sufficient to analyze the case of n series-connected transistors, which are always present in either the pull-up or the pull-down network of static

TABLE I
LEAKAGE CURRENT (IN NANOAMPERES) IN VARIOUS CMOS LOGIC GATES (90-nm TECHNOLOGY)

Inverter gate				
In		*T*=0 °C	*T*=25 °C	*T*=50 °C
0		1.36	3.19	6.52
1		0.24	0.73	1.90
NAND2 gate				
InA	*InB*	*T*=0 °C	*T*=25 °C	*T*=50 °C
0	0	0.17	0.47	1.1
0	1	1.36	3.19	6.52
1	0	1.02	2.44	5.09
1	1	0.48	1.47	3.79

TABLE II
LEAKAGE CURRENT (IN NANOAMPERES) IN VARIOUS CMOS LOGIC GATES (65-nm TECHNOLOGY)

Inverter gate				
InA		*T*=0 °C	*T*=25 °C	*T*=50 °C
0		2.67	2.98	3.66
1		0.13	0.47	1.40
NAND2 gate				
InA	*InB*	*T*=0 °C	*T*=25 °C	*T*=50 °C
0	0	2.37	2.45	2.59
0	1	2.65	2.98	3.66
1	0	2.52	2.77	3.29
1	1	0.26	0.94	2.81

logic gates. For simplicity, let us consider the case of the two series-connected NMOS transistors in the NAND2 gate in Fig. 1(b). For example, let us compare the two cases $B = 0$ and $B = 1$, assuming $A = 1$: When $B = 0$, leakage is greater than that with $B = 1$, as was already observed for the inverter gate. Similar considerations can be easily reiterated and extended to a generic number of series transistors and, hence, to generic static CMOS gates. Hence, the leakage current of static CMOS gates tends to exhibit significantly different values depending on the value of each input, once the other inputs are assigned.

Obviously, this result applies to both combinational and sequential logic gates, and in the following, it will be applied to a broad range of circuits.

B. Leakage in Bit-Sliced Logic Circuits

The strong leakage dependence on the input pattern of basic logic gates is a property that can be exploited to understand the overall leakage of more complex circuits. A practical example of complex circuits that is frequently encountered in real circuits is the case of bit-sliced structures, i.e., circuits with m-bit inputs that are made up of m identical replicas of the same building block. Examples of bit-sliced structures are arithmetic logic units, registers, register files, and bus drivers.

In bit-sliced structures, the overall leakage is equal to the sum of the leakage currents of the m-bit slices, each of which is assumed to be equal to the high (low) level I_H (I_L) when the corresponding input bit is high (low), as was discussed in the previous section (the dual case where I_H is associated with a low input is treated in the same way). Since the number of bit slices having a high leakage current I_H is equal to the number of input bits equal to 1, or equivalently, the Hamming weight w of the input word, the overall leakage results to

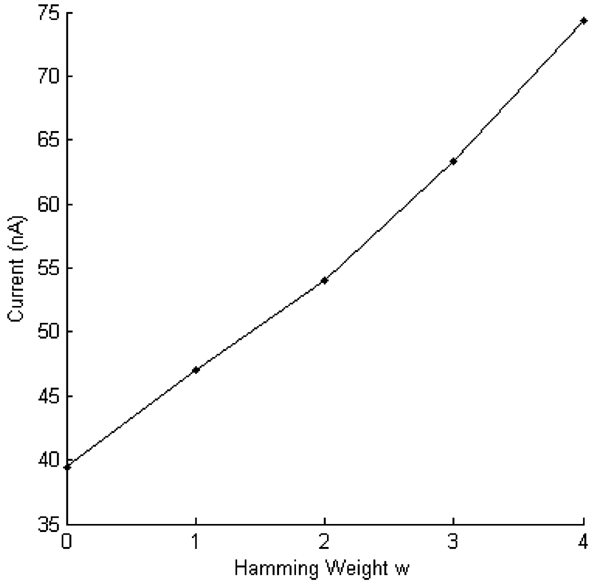

Fig. 2. Simulated leakage versus Hamming weight in 4-bit registers at $T = 27$ °C (65-nm technology).

$$\begin{aligned} I_{\text{leak,TOT}} &= w \cdot I_H + (m - w) \cdot I_L \\ &= w \cdot (I_H - I_L) + m \cdot I_L. \end{aligned} \quad (2)$$

From (2), the overall leakage linearly depends on the Hamming weight w of the input word, rather than the specific value of each bit.

C. Leakage Dependence on the Hamming Weight: Simulation and Experimental Results

The dependence of leakage on the input Hamming weight in (2) is confirmed by simulation results in Table III on a 4-bit register in a 65-nm technology, assuming a temperature of 27 °C (very similar results are obtained for the 90-nm technology; hence, they are omitted in the following). From this table, it is apparent that the register leakage only depends on the weight w of the input data, and this dependence is confirmed to be approximately linear according to Fig. 2, which shows the register leakage current versus w. Parameters I_L and I_H in (2) that fit the curve in Fig. 2 are found to be 9.86 and 18.58 nA, respectively.

To further assess the result in (2), experimental measurements were performed on an off-the-shelf ON Semiconductor 8-bit register of the MC74ACT273N family. With these measurements, the effect of process variations and temperature was also captured. In particular, 100 measurements were performed on each of five different chips at different assigned temperatures. As an example, the measured leakage obtained at $T = 43$ °C is plotted versus the input Hamming weight for the five chips in Fig. 3. To be more specific, the values plotted for each chip

TABLE III
SIMULATED REGISTER LEAKAGE FOR DIFFERENT INPUT DATA VALUES (65-nm TECHNOLOGY, $T = 27$ °C)

input X	Hamming weight w=H(X)	$I_{leak,TOT}$ [nA]
0000	0	39.44
0001	1	47.05
0010		“
0100		“
1000		“
0011	2	54.01
0101		“
0110		“
1001		“
1010		“
1100		“
0111	3	63.39
1011		“
1101		“
1110		“
1111	4	74.30

are the average values among 100 repeated measurements. The estimated standard deviation of these measurements was found to be lower than the average by two orders of magnitude, which confirms that the measurements are reliable and repeatable even in the presence of process variations. According to Fig. 3, the linear trend of leakage approximately holds even under process variations that are seen both within the chip and among the five chips. More comments on process variations will be provided in Section VII, where the slight deviation from the linear trend will be explained. The effect of temperature on (2) was experimentally observed by repeating the aforementioned measurements in a very wide range of temperatures, from 27 °C to 85 °C. For example, the leakage measurement for $T = 85$ °C is shown in Fig. 4, in which the leakage trend is similar to that in Fig. 3, and this was also observed at other temperatures. Hence, the simple model in (2) is confirmed to be valid regardless of the specific operating temperature. This can be justified from (1) by considering that the main impact of temperature on leakage is an equal exponential increase in all transistor leakage currents (or equivalently in I_L and I_H), which preserves the linear dependence in (2). This is also apparent from the comparison of Figs. 3 and 4, in which the trend of $I_{\mathrm{leak,TOT}}$ versus w and the curve slope is almost the same regardless of T and the only difference is the increase of all leakage currents by a factor of about 20.

Summarizing, the leakage current of a bit-sliced structure is directly related to the Hamming weight w of the input data word; thus, it can reveal a significant amount of information on the processed data. This fact is exploited in Section III to propose a novel class of power analysis attacks.

III. LPA ATTACKS

As was discussed in Section II, the leakage current reveals the Hamming weight of the m-bit data X that is processed within a given circuit block. Hence, leakage provides useful information to recover the secret key k of a cryptographic device if the processed data X under attack are a function (or a portion) of k. Hence, it can be conceived that an LPA attack exploits the leakage measurement, as discussed in the following.

In real circuits, the processed data X under attack are generated within a given circuit block, which, in general, is only a part of the entire chip. In practical cases, the power supply node of each block within the chip is not accessible; hence, the adversary can only measure the overall chip leakage, which also includes the contribution of the considered block. Hence, the overall chip leakage $I_{\mathrm{leak,TOT}}$ depends on the Hamming weight $w = H(X)$ of the signal X under attack (where H is the Hamming weight operator), but it also includes many other leakage contributions due to the other blocks within the same chip. As a consequence, when applying random but known input values, the chip leakage $I_{\mathrm{leak,TOT}}$ and $H(X)$ are statistically correlated. This is exactly the premise of the CPA attack [4], [5], which is based on the measurement of dynamic power waveform. As a result, a similar attack procedure can be used in power analysis attacks that

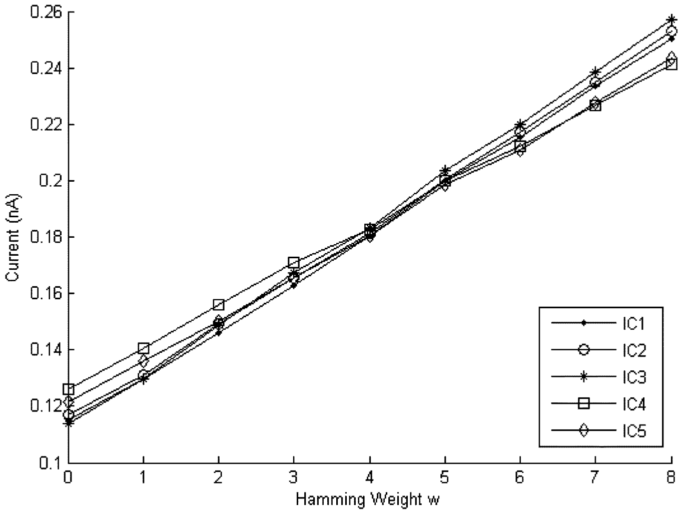

Fig. 3. Measured leakage versus Hamming weight in an ON Semiconductor 8-bit register for five different chips ($T = 43$ °C).

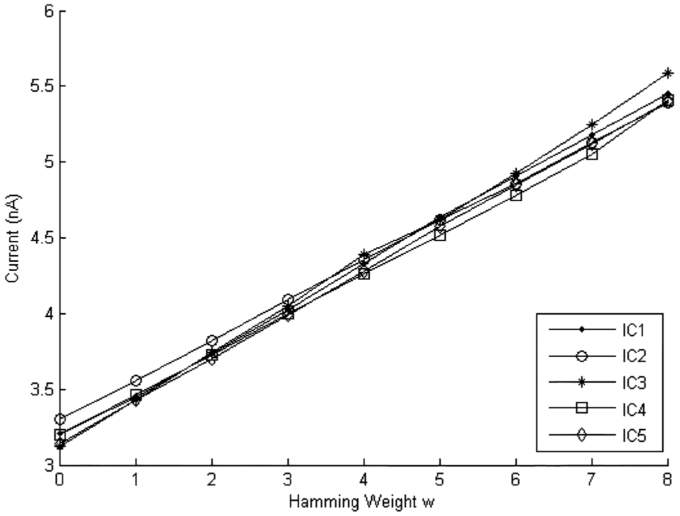

Fig. 4. Measured leakage versus Hamming weight in an ON Semiconductor 8-bit register for five different chips ($T = 85$ °C).

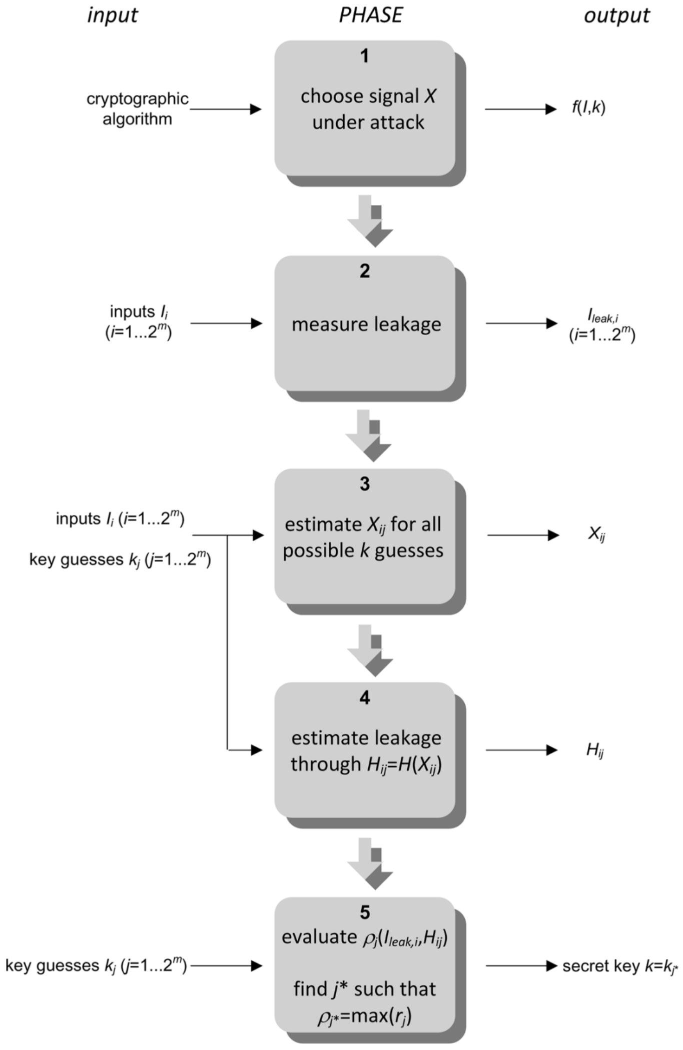
input
PHASE
output
1
choose signal X under attack
cryptographic algorithm
f(I,k)
2
measure leakage
inputs I_i (i=1...2^m)
I_leak,i (i=1...2^m)
3
estimate X_ij for all possible k guesses
inputs I_i (i=1...2^m)
key guesses k_j (j=1...2^m)
X_ij
4
estimate leakage through H_ij=H(X_ij)
H_ij
5
evaluate ρ_j(I_leak,i,H_ij)
find j* such that ρ_j*=max(r_j)
key guesses k_j (j=1...2^m)
secret key k=k_j*

Fig. 5. LPA attack procedure.

are based on leakage measurements, which, in the following, will be referred to as "LPA" attacks.

The following LPA procedure is similar to the CPA presented in [4] and consists of five steps according to Fig. 5. In the first step of LPA attacks, the adversary chooses an internal m-bit signal X that is physically generated within the cryptographic circuit under attack. In general, signal X depends on both the input I and the secret key k of the cryptographic algorithm according to a well-defined function f

$$X = f(I, k) \tag{3}$$

where f is set by the algorithm and, hence, is known by the adversary.

In the second step, the adversary applies 2^m different input values I_i (with $i = 1, \ldots, 2^m$) and measures the corresponding leakage current $I_{\mathrm{leak},i}$ of the cryptographic chip at the point of time in which X is physically evaluated. In principle, this requires the knowledge of the clock period in which X is physically evaluated, as will be assumed in the following for the sake of simplicity. Nevertheless, this assumption can be relaxed, as will be discussed in Section IV. As a result of this step, an array $I_{\mathrm{leak},i}$ with size 2^m is obtained (see Fig. 5).

In the third step, the physical value of X within the chip is estimated for each input I_i according to (3). Since the generic input I_i is applied by the adversary, the only unknown variable in (3) is the secret key k; hence, it must be guessed. For each possible guess k_j of the secret key (with $j = 1, \ldots, 2^m$), the resulting value of $X_{ij} = f(I_i, k_j)$ under the generic input I_i is found according to (3). As a result of this step, a 2-D array X_{ij} is found.

In the fourth step, the leakage current of the block generating X is estimated. In particular, owing to the linear relationship between the leakage current within the block generating X and the Hamming weight $H(X)$ in (2), the current leakage is estimated by $H(X)$. In other words, $H(X)$ differs from the measured leakage only by an unknown multiplicative constant. The output of this step is a 2-D array $H_{ij} = H(X_{ij})$ with $i = 1, \ldots, 2^m$ and $j = 1, \ldots, 2^m$, which contains the Hamming weight of X for all applied inputs and key guesses.

In the fifth step, the measured leakage $I_{\mathrm{leak},i}$ and the estimated leakage H_{ij} are compared. For a given key guess k_j, the sequences $I_{\mathrm{leak},i}$ and H_{ij} associated with the random (but known) sequence of inputs I_i (with $i = 1, \ldots, 2^m$) can be thought of as random variables. When the key guess is correct (i.e., $k_j = k$), the estimated and measured leakages are maximally correlated. Theoretically, if the linear dependence of $I_{\mathrm{leak},i}$ on $H(X)$ in (2) were exact and there were no other leakage contributions, the correlation coefficient $\rho(I_{\mathrm{leak},i}, H_{ij})$ between $I_{\mathrm{leak},i}$ and H_{ij} with $k_j = k$ would be exactly equal to 1 from basic statistics [16]. On the other hand, if the key guess is wrong (i.e., $k_j \neq k$), the measured leakage is no longer linearly related[1] to the estimated $H(X)$; hence, the measured leakage and $H(X)$ are loosely correlated and the correlation coefficient $\rho(I_{\mathrm{leak},i}, H_{ij})$ is lower than unity. This means that the correct guess of k (i.e., the secret key) is that leading to the highest value of $\rho(I_{\mathrm{leak},i}, H_{ij})$ among all possible guesses k_j. Hence, the adversary must evaluate the correlation coefficients $\rho_j = \rho(I_{\mathrm{leak},i}, H_{ij})$ between the measured leakage $I_{\mathrm{leak},i}$ and the Hamming weights H_{ij} for $j = 1, \ldots, 2^m$ and identify the value j^* of j that maximizes ρ_j as in

$$\rho_{j^*} = \max_j \rho_j \tag{4}$$

and the secret key is simply equal to

$$k = k_{j^*}. \tag{5}$$

This is in accordance with the final step of CPA attacks, although they are based on dynamic power measurements [4], [5], [17], [18].

IV. Practical Considerations on LPA Attacks

In the following, practical issues and examples for each of the steps of the aforementioned LPA attack are discussed for cryptographic devices based on a microprocessor and an embedded core.

[1] Obviously, leakage is still linearly dependent on the Hamming weight of the signal X that is physically evaluated within the cryptographic circuit. However, the wrong guess of k leads to an incorrect estimation of X and, hence, of $H(X)$; thus, (2) no longer holds.

TABLE IV
LEAKAGE-CURRENT SETTLING TIME FOR EACH INPUT TRANSITION IN A TWO-INPUT NAND GATE (65-nm TECHNOLOGY, $T = 27\ ^\circ\mathrm{C}$)

From		To		Settling time
A	*B*	*A*	*B*	
0	0	1	0	35.34 ns
0	0	0	1	0.24 ns
0	1	1	1	1.45 ns
0	1	0	0	0.52 ns
1	0	0	0	0.098 ns
1	0	1	1	1.51 ns
1	1	0	1	0.26 ns
1	1	1	0	89.89 ns

In regard to the first step in Fig. 5, the signal X under attack can be easily chosen from the knowledge of the algorithm. To reduce the attack effort, X must be chosen as an internal signal whose dependence $f(I, k)$ on I and k is as simple as possible. For example, in microprocessor-based implementations of the Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithms, the subkeys k are loaded into registers to perform the XOR with the ciphertext I (plaintext) during the decryption (encryption) phase [1], [2]. In these cases, the signal X under attack can be chosen as the input of a register in the register file that stores k (or the XOR of k and the input, i.e., $I \oplus k$, which is successively evaluated); hence, the simple function $f(I, k) = k$ (or $f(I, k) = I \oplus k$) can be adopted in (2). In crypto devices based on an embedded core, similar considerations can be made, although the register under consideration is not in the register file, but divides two pipeline stages. An alternative choice for the signal X is the input (or output) of the bus drivers, since most processed data are transferred through buses to the memory and their leakage is a significant fraction of the overall chip leakage [14] (hence, it can be easily measured [19], [20]).

As far as the second step in Fig. 5 is concerned, the leakage measurements must be carefully performed. Indeed, once the input is applied to a logic gate, its leakage current is well known to have a transient variation and finally settles to the steady-state value after a period ranging from less than 1 ns to a few tens of nanoseconds [14]. As an example, Table IV reports the leakage settling time of a two-input CMOS NAND gate in a 65-nm technology (very similar results were obtained in the 90-nm technology; hence, they are omitted). From these considerations, the clock period is generally comparable or greater than the period required to observe the steady-state leakage, particularly for nanometer technologies. As a consequence, usually, the adversary is not required to stop the clock to measure $I_{\mathrm{leak},i}$ in the period in which X is physically evaluated.

It is worth noting that the exact clock period in which X is evaluated within the chip can be found with a moderate effort, if the adversary has sufficient knowledge of the circuit implementation of the algorithm. For example, in microprocessor-based implementations, the correct clock period can be identified by counting the number of memory accesses that are required prior to the evaluation of X. In embedded core implementations, the identification of the clock period in which X is physically evaluated is even easier, as the circuit implementation usually consists of a low number of pipeline stages [21]; hence, the number of clock cycles to consider is rather low (a few units, typically).

Even in the case where the adversary does not exactly know this clock cycle but knows that it is among a limited number l of clock cycles, he/she can reiterate the procedure in Section III for each period and then evaluate the maximum correlation coefficients ρ_j for each of the l periods under analysis. Obviously, the maximum ρ_j is expected to be achieved in the correct period; hence, the secret key is easily found by selecting the guess of k and the period that maximizes ρ_j. Since this increases the number of guesses by a factor of l (and, hence, the attack effort), the adversary must have at least a rough idea on the clock period in which the signal under attack X is physically evaluated.

Regarding the leakage measurement setup, it should be observed that leakage measurements are, in principle, simpler to carry out, compared with dynamic power measurements in traditional DPA/CPA attacks. In fact, the latter ones require the acquisition of power-consumption waveforms with a high-bandwidth measurement setup and a high-sample-rate digital oscilloscope. Leakage measurements do not require either high bandwidth or high-sample-rate oscilloscopes, as they are simply dc measurements that can be carried out using a simple ammeter. Typically, ammeters with picoampere accuracy can be found at very low prices (in the range of $500–1000); hence, the costs involved in the attack are extremely low (even lower than those in DPA/CPA attacks). As a further advantage of LPA, leakage measurements are rather insensitive to ac additive noise since they are based on measurement averaging over a sufficiently long period of time, as any other dc measurement.

A distinctive feature of LPA attacks is the high sensitivity to temperature, as pointed out in Section II, which deserves some attention during measurements. Indeed, in Section II-C, it was shown that the leakage model in (2) holds regardless of the operating temperature. Nevertheless, it should be observed that this is true only if the chip temperature is not time varying; otherwise, the leakage current may significantly increase (decrease) despite of a reduction (increase) in the Hamming weight in (2), due to an unexpected temperature increase (decrease). For this reason, it is important that the chip temperature is kept constant during LPA attacks. In our experimental results, temperature was set by means of a Peltier cell driven by a proper constant voltage.

The third and fourth steps of LPA attacks do not exhibit particular problems from an experimental point of view, but the choice of the bit width m of the signal X under attack is crucial in terms of computational effort. Indeed, the number of guesses exponentially grows as 2^m; hence, m must be at most seven to eight for computationally feasible attacks. It is worth noting that this does not limit the effectiveness of LPA attacks, as the overall key can be recovered by reiterating the attack for the remaining m-bit portions of the key. This is feasible in practical cases because the intermediate results of a cryptographic device consist of small words that depend on a few bits of the key and not on the entire key (for example, in microprocessor-based devices, the width of the portion of the key that is involved in a single encryption step is no greater than the processor word length).

Like any cryptographic operation, these intermediate results are evaluated by combinational logic and stored in a register at a certain time during the computation of the algorithm.

In regard to the last step in Fig. 5, the correlation coefficient $\rho(I_{\text{leak},i}, H_{ij})$ cannot be exactly evaluated since a finite number of samples of $I_{\text{leak},i}$ and H_{ij} are considered. In practical cases, $\rho(I_{\text{leak},i}, H_{ij})$ is estimated by the sample correlation coefficient (often called "Pearson's correlation coefficient") in (6), where the number of applied inputs was assumed to be 2^m for simplicity [16]

$$r_j = \frac{\sum_{i=1}^{2^m} I_{\text{leak},i} H_{ij} - 2^m \overline{I_{\text{leak},\imath}} \cdot \overline{H_{\imath j}}}{(2^m - 1) S_{I_{\text{leak},i}} S_{H_{ij}}} \tag{6}$$

where the sample mean $\overline{I_{\text{leak},\imath}}$ and $\overline{H_{\imath j}}$ are defined as

$$\overline{I_{\text{leak},\imath}} = \frac{1}{2^m} \sum_{i=1}^{2^m} I_{\text{leak},i} \tag{7}$$

$$\overline{H_{\imath j}} = \frac{1}{2^m} \sum_{i=1}^{2^m} H_{ij} \tag{8}$$

and the sample standard deviation $S_{I_{\text{leak},i}}$ and $S_{H_{ij}}$ are

$$S_{I_{\text{leak},i}} = \sqrt{\frac{1}{2^m - 1} \sum_{i=1}^{2^m} (I_{\text{leak},i} - \overline{I_{\text{leak},\imath}})^2} \tag{9}$$

$$S_{H_{ij}} = \sqrt{\frac{1}{2^m - 1} \sum_{i=1}^{2^m} (H_{ij} - \overline{H_{\imath j}})^2}. \tag{10}$$

Finally, it is useful to observe that the overall chip leakage contribution includes the leakage current of many other blocks that do not physically evaluate X, as what occurs with the dynamic power consumption in traditional CPA attacks [5]. These contributions affect both $I_{\text{leak},i}$ and $\overline{I_{\text{leak},\imath}}$ in (6) and tend to reduce the sample correlation coefficient with respect to the ideal value of 1, as was observed in [5] for CPA attacks. In particular, in the case where the leakage contributions of the other blocks are perfectly constant, they do not effect r_j at all, as is apparent from (6). The same result approximately holds also when these contributions randomly vary with a reasonably uniform distribution when applying the inputs I_i, as they tend to average out [5]. This intuitively justifies why LPA attacks can be performed even in the presence of significant leakage contributions that add to the leakage current of the block under attack. This issue will be further discussed through an example in Section VI.

V. Analytical Model of the Correlation Coefficient in LPA Attacks

LPA attacks rely on the assumption that the correlation coefficient ρ_{j^*} associated with the correct key can be discriminated from that of wrong guesses. This requires that the value of ρ_{j^*} under the correct key (i.e., $\rho_{j^*} = 1$) is sufficiently greater than those under the other guesses. For this reason, in the following, the correlation coefficient under wrong guesses is analytically evaluated according to the approach in [22].

For the sake of simplicity, let us assume, with no loss of generality, that $I_L = 0$ and $I_H = 1$ in (2), that the signal under attack X is the XOR of the input I and the key portion k (i.e., $f(I, k) = I \oplus k$, for the reasons discussed in Section IV), and that the key is $k = 00, \ldots, 00$ (so that $f(I, k) = I$), and let us apply all 2^m possible m-bit input values I_i to a bit-sliced circuit. Accordingly, the leakage $I_{\text{leak},i}$ associated with the generic value of X is linearly related to its Hamming weight $H(X)$ from (2). Hence, the correlation coefficient in (2) is equal to 1 when $H(X)$ is correctly predicted, i.e., when the correct key guess has been chosen. On the other hand, when a wrong guess is made, the Hamming weight $H(X)$ is incorrectly predicted. Moreover, the higher is the number of wrong bits in X, the lower is the correlation coefficient, and hence, the easier is the discrimination from the correct guess (since, in this case, the correlation coefficient is far from the unit value obtained with the correct key). Hence, the most difficult cases to discriminate from the correct guess are those with a minimum number of wrong bits in X, i.e., wrong by just 1 bit. In this case, after performing the simple calculations in the Appendix, it is found that the correlation coefficient ρ_{wrong} obtained when only 1 bit is wrong during the guess of X is

$$\rho_{\text{wrong}} = 1 - \frac{2}{m}. \tag{11}$$

From (11), it is apparent that ρ_{wrong} is easily discriminated from the correlation coefficient value associated with the correct guess (i.e., 1) if m is a few units. On the other hand, if m is in the order of eight to ten, ρ_{wrong} gets very close to unity; hence, it is very hard to distinguish the case of a wrong guess from the correct key. This constraint on practical values of m adds to the existing requirement $m \leq 8$ that resulted from considerations on the number of guesses in Section IV. For both reasons, LPA attacks can be successful only if the number of bits under attack m is not greater than about eight.

VI. Examples of LPA Attacks

Let us apply the LPA attack procedure explained in Section III to three different circuits. In the first one, the 4-bit register in the 65-nm technology discussed in Section II is attacked in simulation, applying a sequence of all possible 4-bit input values in the first column in Table III. The measured leakage $I_{\text{leak},i}$ and the Hamming weight $H(X)$ for each input value are reported in the second and third columns of Table III. The sample correlation coefficient r between the measured leakage vector $I_{\text{leak},i}$ (with $i = 1, \ldots, 16$) and the Hamming weight vector $H(X)$ is then computed according to (6). Using data in Table III, r results to one for the correct logic vector as expected, while r results to 0.41 if only one wrong bit is considered. The latter value reasonably agrees with the theoretical value of 0.5 (within 20%), which is obtained from (11) after setting $m = 4$. Hence, the LPA attack allows for identifying the correct guess, as the corresponding correlation coefficient is the highest. Moreover, it can be easily distinguished from the correlation coefficient obtained with a key guess that is wrong by 1 bit.

In the aforementioned attack, a bit-slice circuit (i.e., a register) was considered. The register is an ideal candidate as a block to attack with LPA for the previously explained reasons. Nevertheless, the aforementioned considerations and attack procedure can also be applied to circuits that are not bit sliced and involve nonlinear transformations. Indeed, even in this case,

TABLE V
S-BOX TRUTH LEAKAGE CURRENT (65-nm TECHNOLOGY, $T = 25\ ^\circ\mathrm{C}$ AND $100\ ^\circ\mathrm{C}$)

IN	*OUT*	$I_{leak,i}$ (nA) @T=25 °C	$I_{leak,i}$ (nA) @T=100 °C
0100	1111	114.6	941.6
1001	1101	115.0	950.0
0000	0011	117.1	950.2
0001	1110	117.9	959.7
1000	1000	118.8	964.0
0011	0111	121.9	999.8
1100	0001	122.9	1007.8
0010	1010	123.4	1016.2
1011	0000	124.0	1024.0
1010	0110	124.8	1026.3
1111	1100	125.4	1031.6
0101	0100	127.1	1031.9
1101	0010	128.2	1032.7
0110	0101	129.6	1051.9
1110	1011	130.7	1052.8
0111	1001	131.6	1071.7

leakage has a strong correlation with the Hamming weight of the input data, although this dependence is no longer linear due to the nonlinear transformation. In the following, this is shown by attacking the well-known Serpent 4 × 4 S-Box transformation, whose truth table is reported in Table V[23].

As a second example of LPA attack, the simple crypto core in Fig. 6 based on the Serpent S-Box transformation was considered. This crypto core was synthesized in Cadence environment using a 65-nm CMOS cell library. As what occurs in many cryptographic algorithms, in Fig. 6, the plain word and the secret key are mixed in advance by XOR gates, and the result is ciphered by S-Box (a similar structure is observed in many other ciphers, such as DES and AES). Simulations on this crypto core were carried out by exploring all possible combinations of plain words and keys. The results are shown in Table V, where leakage-current values are sorted in an increasing order. Interestingly, by sorting the input combinations in an increasing order according to the leakage, the order is the same regardless of the temperature, as was observed over an extremely large range of temperatures (much wider than realistic operating temperatures). Again, the LPA attack effectiveness is expected to be independent of the operating temperature, provided that it is kept constant during the attack. As an example, the cases with $T = 25\ ^\circ\mathrm{C}$ and $100\ ^\circ\mathrm{C}$ are shown in Table V.

The resulting correlation coefficients in (6) of several attacks (i.e., for several keys) for $T = 25\ ^\circ\mathrm{C}$ are shown in Fig. 7, where the "$+$" symbol denotes the right key hypothesis, whereas the "$*$" symbol denotes a wrong key hypothesis. Very similar results were obtained for $T = 100\ ^\circ\mathrm{C}$. From Fig. 7, the correlation coefficient turns out to be lower than unity, since the circuit is not bit sliced. The same figure reveals that all correct keys are clearly distinguishable from wrong key guesses, as the former ones always lead to greater values of the correlation coefficient. More specifically, the minimum value of the correlation coefficient under a correct key is 0.11, whereas the maximum value under a wrong-by-1-bit key guess is 0.10, and in other cases, the difference is much larger.

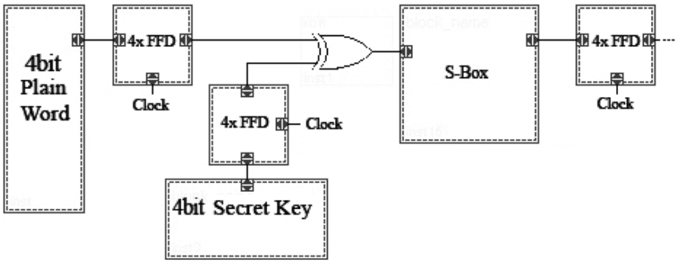

Fig. 6. Crypto core based on Serpent S-Box.

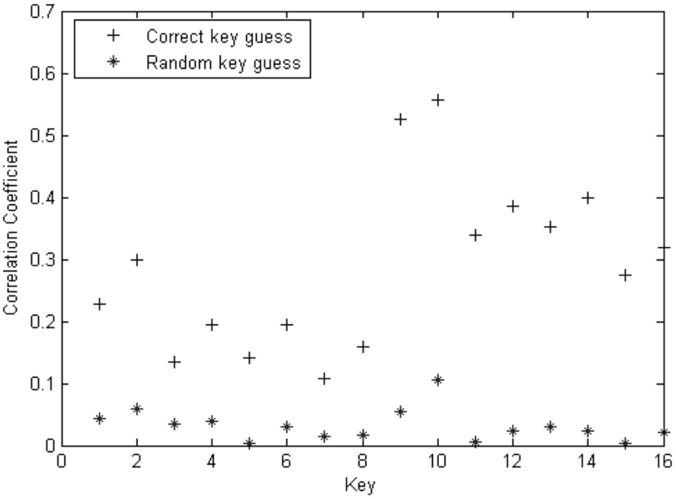

Fig. 7. Correlation coefficient in a simulated attack.

The third example of LPA attack has been conducted on a real circuit. In particular, a portion of the crypto core in Fig. 6 was implemented in the 8-bit architecture in Fig. 8, employing an off-the-shelf ON Semiconductor 8-bit register of the MC74ACT273N family. In particular, 256 leakage measurements have been collected for all possible 8-bit input values with a fixed key. Then, the correlation coefficient between the measured leakage and the predicted Hamming weight $H(X)$ was evaluated for each of the 256 key hypotheses. The resulting magnitude of the correlation coefficient normalized to the maximum value is shown in Fig. 9 for all key guesses. From this figure, the highest value of the correlation coefficient is obtained under the correct key $(01000010)_2 = (66)_{10}$, as expected. There is also another equal peak with an opposite sign that is obtained under the symmetric key $(10111101)_2 = (189)_{10}$, i.e., by complementing all bits of the correct key. This is easily explained by considering that the XOR function is symmetric; hence, by complementing the key bits, the XOR with the input is also bitwise complemented.

For the sake of completeness, a lower number of bits m was also considered in the attack of the circuit in Fig. 8. In particular, the four least significant bits (i.e., $m = 4$) were attacked instead of the eight output bits of the core, as a representative case in

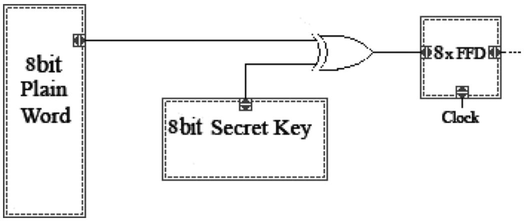

Fig. 8. Cryptographic circuit under experimental attack.

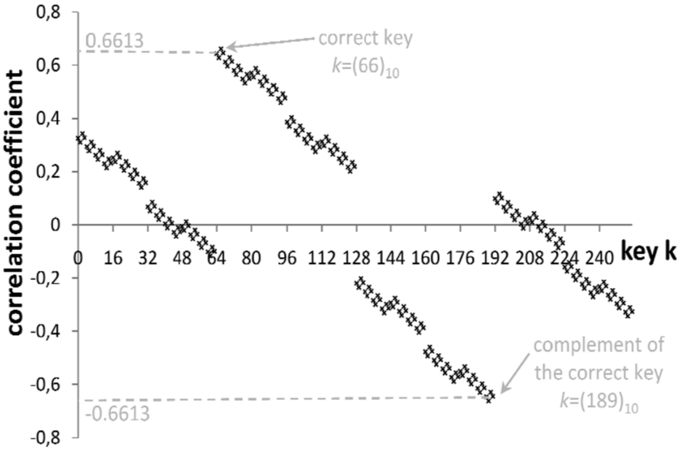

Fig. 9. Correlation coefficient ρ_j for all possible key guesses (LPA attack).

which m is lower than the datapath width n. The highest value of the correlation coefficient was again obtained under the correct key $(66)_{10}$ and results to 0.63, which is close to the previously obtained value of 0.66. Several other attacks with different keys and at various temperatures were repeated, and results showed that they were always successful, as the correct key was always associated with the greatest value of ρ. This confirms the effectiveness of LPA attacks on real hardware and in real conditions.

Finally, we also considered the attack presented in [12], which is based on the DPA procedure to identify the key, instead of the CPA adopted in this paper. In contrast to [12], in which the attack was only simulated with predictive models, we performed an experimental attack for the first time in the literature, at the best of our knowledge. In particular, the attack presented in [12] was performed on the 8-bit crypto core in Fig. 8. The highest DPA peak $(DPA = 0.52)$ was obtained for the right key $(66)_{10}$, as shown in Fig. 10. This means that the attack in [12] is also able to identify the secret key. However, this attack cannot easily distinguish the peak associated with the correct key from those under wrong keys, as clearly shown in Fig. 10, where it is apparent that various peaks are very close to the highest one. This was quantitatively shown by evaluating the number of keys nk whose peak differs by less than a given percentage, compared to the correct key. In general, it was found that nk in the attack in [12] is approximately twice that obtained with LPA. For example, the number nk of peaks within 10% of the correct key peak was found to be 15 for the attack in [12] and 7 for the LPA. This means that, in the attack in [12], it is harder to distinguish the correct key, particularly if we consider that there is always an additive noise in measurements. This fact can be easily understood by recalling that the DPA procedure to identify the correct key is less effective than the CPA procedure adopted in this paper, as is well known from previous works on DPA and CPA attacks based on the dynamic energy analysis [4], [5]. This justifies the superiority of the attack procedure adopted in this paper, compared with that in [12].

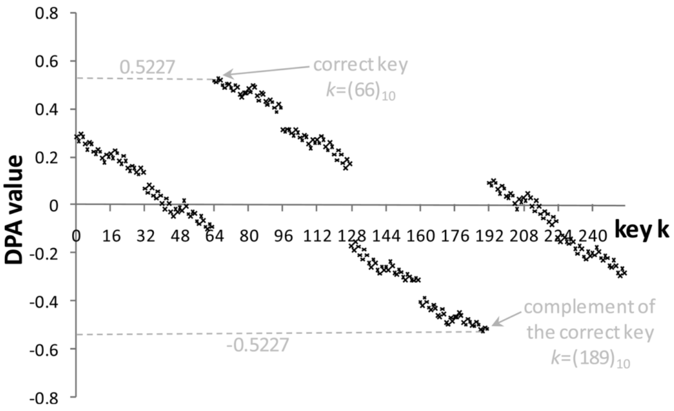

Fig. 10. DPA peak for all possible key guesses (attack in [12]).

VII. Considerations on Process Variations

In the previous sections, LPA attacks were analyzed without explicitly accounting for process variations. In the following, their effect of the attack effectiveness is discussed, since their impact tends to increase in nanometer CMOS technologies.

Process variations are usually classified into interdie and intradie. The former ones impact all devices in the same chip in the same manner [24]; hence, they induce a leakage variation that is equal for all transistors, according to (1)[25]. Thus, the leakage contributions of all gates scale by the same factor, which is similar to the effect of temperature variations previously discussed. As a consequence, the linear dependence in (2) is preserved, and the sample correlation coefficient (6) does not change. Hence, interdie variations do not affect the result of LPA attacks.

On the other hand, intradie variations affect different transistors in the same chip in a different way [15]; hence, two flip-flops give a different leakage contribution even when they have the same input. As a consequence, two different input patterns with the same weight w lead to different register leakage currents. Hence, in this case, the leakage depends on the specific applied input (not only on w), i.e., a slight deviation from the linear trend in (2) is observed, which agrees with the experimental results in Figs. 3 and 4. Hence, intradie variations can potentially reduce the effectiveness of LPA attacks.

To understand the effect of intradie variations, it is useful to observe that they induce a statistical variation in the correlation coefficient r in (6). Accordingly, the correlation coefficient r for a given key guess can be seen as an uncertainty range centered on its nominal value, as shown in Fig. 11(a). The attack is feasible even in the presence of intradie variations if the value of r under the correct guess can be distinguished from that under a wrong guess, i.e., if we are reasonably sure (within a given confidence level, e.g., 99.9%) that the two uncertainty ranges do not overlap, as in Fig. 11(a). On the other hand, if the two

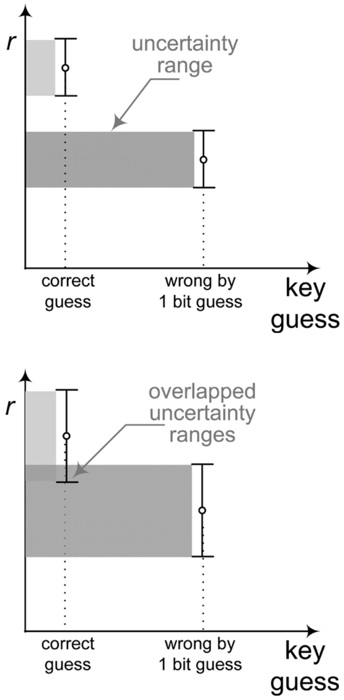

Fig. 11. (a) Correlation-coefficient statistical distribution under moderate intradie variations (successful attack). (b) Correlation-coefficient statistical distribution under high intradie variations (unsuccessful attack).

ranges overlap [see Fig. 11(b)], the attack may be unsuccessful (see the following for further details).

To evaluate the amplitude of the uncertainty range, we first performed numerical simulations by adopting the simplified leakage model in (1) for each bit slice and introducing[2] random intradie variations ΔV_{TH} in V_{TH}. Variations ΔV_{TH} were randomly generated with normal distribution, zero mean, and standard deviation σ_{VTH}. Obviously, for low (high) values of σ_{VTH}, the uncertainty width in Fig. 11 is small (large) and the uncertainty ranges do not (do) overlap, as shown in Fig. 11(a) and (b). Accordingly, we evaluated the maximum standard deviation $\sigma_{\mathrm{VTH,max}}$ of V_{TH} above which the uncertainty ranges overlap as in Fig. 11(b), with a 99.9% confidence level.

Results show that $\sigma_{\mathrm{VTH,max}}/(nkT/q) \approx 1$ regardless of the value of m if we consider the uncertainty range of the correct key and of the key guesses that are wrong by 1 bit. Under the typical value $n = 1.5$ and at room temperature, $\sigma_{\mathrm{VTH,max}}$ results to 37 mV, which, unfortunately, is close to typical values of σ_{VTH} in 65- and 45-nm CMOS technologies [14], [26]. Hence, in current technologies, the variations of V_{TH} may lead to an overlap between the uncertainty range of the correct key and that of a key guess that is wrong by 1 bit. As a consequence, the attack may be successful in some cases (in a probabilistic sense), whereas in some others, the result of the attack may be incorrect (i.e., a key guess that is wrong by 1 bit is mistaken for the correct key). Even in the latter case, the attack is still successful under the assumption that the correct key and the found key differ by at most 1 bit (its validity is discussed hereafter): In this case, the adversary has to search the key in an extremely narrow space with $m+1$ elements (i.e., the key found in the attack or one of the m keys that differ from it by only 1 bit), compared with the exhaustive search among the 2^m possible key guesses.

Now, let us justify the aforementioned assumption that the correct key and the key found in the attack differ by at most 1 bit. The previously discussed numerical simulations showed that $\sigma_{\mathrm{VTH,max}}/(nkT/q) \approx 1.5$ when we consider the key guesses that are wrong by 2 bits (and even more if the number of wrong bits is higher): In this case, $\sigma_{\mathrm{VTH,max}} \sim 56\ \mathrm{mV}$ is significantly greater than σ_{VTH} both for current and future technologies (for example, in 32-nm technologies, σ_{VTH} is projected to be about 40 mV [25], [26]). This means that the uncertainty range of the key guesses with two or more wrong bits never overlaps with the range of the correct key [as in Fig. 11(a)], neither in current nor in future CMOS technologies.

The aforementioned numerical results are in good agreement with Monte Carlo circuit simulations, which were performed on 1000 trials of the same 65-nm register circuit affected by process variations. By analyzing the statistical distribution of the resulting 1000 values of the correlation coefficient, it was found that, sometimes, the correct key may be mistaken for key guesses with one wrong bit, but it is never confused with guesses with two or more wrong bits. This agrees well with the aforementioned considerations found with the simplified leakage model. In particular, it was found that the attack identifies the correct key in 96% of the cases. Hence, in most cases, the attack provides the correct result, and in the other few cases, the found key has at most one wrong bit, which again agrees well with previous numerical analyses.

From the aforementioned considerations, LPA attacks are expected to be successful even in the presence of intradie variations, both in current technologies and in the next technology nodes.

VIII. Conclusion

For the first time in the literature, this paper has introduced, described, and analyzed LPA attacks to cryptographic circuits by providing a theoretical background, discussing experimental issues related to real attacks, and presenting results of experimental attacks.

A simple attack procedure based on the correlation coefficient evaluation has been introduced for the first time, and various LPA attacks to simple circuits have been performed in simulations and experiments. Results confirm the validity of the underlying assumption and the effectiveness of this kind of attacks. Experimental issues have been discussed, including the effect of temperature and process variations, which strongly affect the leakage current of sub-100-nm VLSI circuits. In particular, it is shown that the temperature of the cryptographic chip must be kept constant during the LPA attack, in order to recognize the secret key. Analysis shows that LPA attacks are successful even in the presence of process variations, and this is shown to be valid even in future technologies. An analytical expression of the correlation coefficient to predict the result of the LPA attack is derived. As a practical rule of thumb, it is also shown that

[2] Intradie variations of other parameters (e.g., $W, L, \ldots$) can always be described as an equivalent variation in V_{TH}[23]. Hence, this assumption does not limit the generality of the subsequent considerations.

LPA attacks are feasible if the number of bits under attack is no greater than seven to eight.

Although this paper clarifies various aspects of LPA attacks, it has to be considered a starting point for further investigation, as many issues are not fully understood. For example, the effect of temperature on attacks is not clear, whereas simple criteria to properly set the chip temperature to maximize the attack effectiveness would be desirable, in order to test cryptographic circuits under worst conditions. For the same reason, criteria to build measurement setups for fast and effective attacks are required. Moreover, experimental attacks on more complex circuits should be performed to better understand the robustness against LPA attacks versus their complexity. Furthermore, criteria to consciously select the number of measurements to have a reasonably accurate estimation of the correlation coefficient should be identified. Finally, much work will be required in the future to devise countermeasures that are able to improve the robustness of cryptographic circuits against LPA attacks.

Appendix

Let us evaluate the correlation coefficient under a wrong-by-only-1-bit guess, assuming that, with no loss of generality, the wrong bit in X is the most significant bit (MSB) X_{MSB}. The predicted Hamming weight $H_{\text{wrong}}(X)$ differs from the correct $H(X)$ by exactly one, due to the wrongly predicted contribution of X_{MSB}. Indeed, if the predicted X_{MSB} is zero, it is actually one due to the wrong guess; hence, the predicted Hamming weight $H_{\text{wrong}}(X)$ is equal to $H(X)-1$; analogously, if X_{MSB} is predicted to be one, $H_{\text{wrong}}(X)$ is equal to $H(X)+1$; hence

$$H_{\text{wrong}}(X) = \begin{cases} H(X)-1, & \text{if } X_{\text{MSB}} = 0 \\ H(X)+1, & \text{if } X_{\text{MSB}} = 1. \end{cases} \tag{A.1}$$

Now, let us consider the case where all input values X_i are equally likely and are progressively applied, as in Table VI from $i = 1$ (corresponding to value $00,\ldots,00$) to $i = 2^m$ (corresponding to value $11,\ldots,11$). Hence, by definition, the correlation coefficient ρ_{wrong} between $H(X)$ and the leakage $I_{\text{leak},i}$ corresponding to the same input X_i is expressed as shown at the bottom of the page, where $\overline{H_{\text{wrong}}(X_\imath)}$ and $\overline{I_{\text{leak},\imath}}$ are the average values of the Hamming weight and leakage [defined according to (7) and (8)].

TABLE VI
ORDER OF APPLIED INPUTS X_i

i	X_i	i	X_i
1	000…00	2^{m-1}+1	100…00
2	000…01	2^{m-1}+2	100…01
3	000…10	2^{m-1}+3	100…10
…	…	…	…
2^{m-1}	011…11	2^m	111…11

Assuming that, for simplicity, $I_L = 0$ and $I_H = 1$ in (2), the leakage $I_{\text{leak},i}$ in (A.2) associated with the generic value of X is simply equal to its Hamming weight $w = H(X)$ from (2). It is useful to observe that $H_{\text{wrong}}(X_i)$ and $H(X_i)$ have the same average, since they are evaluated over the same set of values X_i (the only difference is the order, as $H_{\text{wrong}}(X_i)$ is evaluated by complementing the MSB of X_i, as discussed before). For the same reason, $\sum_{i=1}^{2^m}(H_{\text{wrong}}(X_i) - \overline{H_{\text{wrong}}(X_\imath)})^2$ is equal to $\sum_{i=1}^{2^m}(H(X_i) - \overline{H(X_\imath)})^2$. Hence, (A.2) can be written as

$$\begin{aligned}\rho_{\text{wrong}} &= \frac{\sum_{i=1}^{2^m}\left(H_{\text{wrong}}(X_i) - \overline{H(X_\imath)}\right)\left(H(X_i) - \overline{H(X_\imath)}\right)}{\sum_{i=1}^{2^m}\left(H(X_i) - \overline{H(X_\imath)}\right)^2} \\ &= \frac{\sum_{i=1}^{2^m}\left(H_{\text{wrong}}(X_i) - m/2\right)\left(H(X_i) - m/2\right)}{2^m\frac{m}{4}}\end{aligned} \tag{A.3}$$

where the average $\overline{H(X_\imath)}$ and variance $(1/2^m)\sum_{i=1}^{2^m}(H(X_i) - \overline{H(X_\imath)})^2$ of the Hamming weight of uniformly distributed m-bit symbols are well known to be $m/2$ and $m/4$, respectively [22].

From Table VI, for $i = 1,\ldots,2^{m-1}$, we have $X_{\text{MSB}} = 0$; hence, $H_{\text{wrong}}(X) = H(X) - 1$ from (A.1). Analogously, for $i = 2^{m-1}+1,\ldots,2^m$, we have $X_{\text{MSB}} = 1$; hence, $H_{\text{wrong}}(X) = H(X) + 1$. Thus, (A.3) simplifies into

$$\begin{aligned}\rho_{\text{wrong}} = &\frac{\sum_{i=1}^{2^{m-1}}\left(H(X_i) - 1 - m/2\right)\left(H(X_i) - m/2\right)}{2^m\frac{m}{4}} \\ &+ \frac{\sum_{i=2^{m-1}+1}^{2^m}\left(H(X_i) + 1 - m/2\right)\left(H(X_i) - m/2\right)}{2^m\frac{m}{4}}\end{aligned} \tag{A.4}$$

which, after a few simple manipulations, becomes

$$\rho_{\text{wrong}} = 1 - \frac{\sum_{i=2^{m-1}+1}^{2^m} H(X_i) - \sum_{i=1}^{2^{m-1}} H(X_i)}{2^m\frac{m}{4}}. \tag{A.5}$$

From Table VI, for $i = 1,\ldots,2^{m-1}(i = 2^{m-1}+1,\ldots,2^m)$, we have $X_{\text{MSB}} = 0$, whereas the other digits of $X_{i+2^{m-1}}$ are equal to those of X_i. As a consequence, $H(X_{i+2^{m-1}}) = H(X_i) + 1$ in (A.5), thereby yielding

$$\rho_{\text{wrong}} = 1 - \frac{2^{m-1}}{2^m\frac{m}{4}} \tag{A.6}$$

which demonstrates (11). Q.E.D

$$\rho_{\text{wrong}} = \frac{\sum_{i=1}^{2^m}\left(H_{\text{wrong}}(X_i) - \overline{H_{\text{wrong}}(X_\imath)}\right)\left(I_{\text{leak},i} - \overline{I_{\text{leak},\imath}}\right)}{\sqrt{\sum_{i=1}^{2^m}\left(H_{\text{wrong}}(X_i) - \overline{H_{\text{wrong}}(X_\imath)}\right)^2 \sum_{i=1}^{2^m}\left(I_{\text{leak},i} - \overline{I_{\text{leak},\imath}}\right)^2}} \tag{A.2}$$

REFERENCES

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO*, 1999, pp. 388–397.

[2] *AES—Federal Information Processing Standards Publication (FIPS PUB)* , 197 [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[3] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[4] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Berlin, Germany: Springer-Verlag, 2007.

[5] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. CHES*, Boston, MA, Aug. 2004, vol. 3156, Lecture Notes in Computer Science, pp. 16–29.

[6] Z. Chen, L. Wei, M. Johnson, and K. Roy, "Estimation of standby leakage power in CMOS circuits considering accurate modelling of transistor stacks," in *Proc. ISLPED*, Monterey, CA, Aug. 1998, pp. 239–244.

[7] S. Narendra, S. Borkar, V. De, D. Antoniadis, and A. Chandrakasan, "Scaling of stack effect and its application for leakage reduction," in *Proc. Int. Symp. Low Power Electron. Des.*, 2001, pp. 195–200.

[8] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 2, pp. 140–154, Feb. 2004.

[9] International Technology Roadmap for Semiconductors 2006. [Online]. Available: http://public.itrs.net

[10] J. Giorgetti, G. Scotti, A. Simonetti, and A. Trifiletti, "Analysis of data dependence of leakage current in CMOS cryptographic hardware," in *Proc. GSLVLSI*, Stresa, Italy, Mar. 11, 2007, pp. 78–83.

[11] L. Giancane, M. Jovanovich, G. Scotti, and A. Trifiletti, "Leakage power analysis of cryptographic devices implemented in nanometer CMOS technologies," in *Konferencija 9-a 07: Konferenciju za Elektroniku, Telekomunikacije, Racunarstvo, Automatiku I Nuklearnu Tehniku, Herceg Novi*, Igalu, Montenegro, Jun. 4–8, 2007.

[12] L. Lin and W. Burleson, "Leakage-based differential power analysis (LDPA) on sub-90 nm CMOS cryptosystems," in *Proc. ISCAS*, Seattle, WA, May 2008, pp. 252–255.

[13] Y. Tsividis, *Operation and Modeling of the Transistor MOS*, 2nd ed. London, U.K.: Oxford Univ. Press, 2003.

[14] S. G. Narendra and A. Chrakasan, *Leakage in Nanometer CMOS Technologies*. Berlin, Germany: Springer-Verlag, 2006.

[15] S. R. Nassif, "Modeling and forecasting of manufacturing variations," in *Proc. ASP-DAC*, 2001, pp. 145–150.

[16] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*. Englewood Cliffs, NJ: Prentice-Hall, 2006.

[17] S. Aumonier, "Generalized Correlation Power Analysis in the workshop ECRYPT," in Tools for Cryptanalysis. Krakow, Poland, 2007.

[18] F. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, "An overview of power analysis attacks against field programmable gate arrays," *Proc. IEEE*, vol. 94, no. 2, pp. 383–394, Feb. 2006.

[19] M. Alioto, M. Poli, S. Rocchi, and V. Vignoli, "Power modeling of precharged address bus and application to multi-bit DPA attacks to DES algorithm," in *Proc. PATMOS*, Montpellier, France, Sep. 2006, pp. 593–602.

[20] M. Alioto, M. Poli, and S. Rocchi, "Differential power analysis attacks to precharged busses: A general analysis for symmetric-key cryptographic algorithms," *IEEE Trans. Depend. Secure Comput.*, to be published.

[21] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 9, pp. 957–967, Sep. 2004.

[22] M. Alioto, M. Poli, and S. Rocchi, "A general power model of differential power analysis attacks to static logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to be published.

[23] R. Anderson, E. Biham, and L. Kundsen, "A proposal for the advanced encryption standard," in AES Proposal 1998. [Online]. Available: http://www.cl.cam.ac.uk/~rja14/serpent.html

[24] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf, "The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 4, pp. 360–368, Dec. 1997.

[25] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. Berlin, Germany: Springer-Verlag, 2005.

[26] H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki, "Challenge: Variability characterization and modeling for 65- to 90-nm processes," in *Proc. CICC*, Sep. 2005, pp. 593–599.

[27] F. RegazzoniS. BadelT. EisenbarthJ. GroßschädlA. PoschmannZ. ToprakM. MacchettiL. PozziC. PaarY. LeblebiciP. Ienne, "A simulation-based methodology for evaluating the DPA-resistance of cryptographic functional units with application to CMOS and MCML technologies," in *Proc. IC_SAMOS*, Jul. 2007, pp. 209–214.

Massimo Alioto (M'01–SM'07) was born in Brescia, Italy, in 1972. He received the Laurea degree in electronics engineering and the Ph.D. degree in electrical engineering from the University of Catania, Catania, Italy, in 1997 and 2001, respectively.

In 2002, he joined the Dipartimento di Ingegneria dell'Informazione (DII), Università di Siena, Siena, Italy, as a Research Associate and, in the same year, as an Assistant Professor. In 2005, he was appointed Associate Professor of electronics and was engaged in the same faculty in 2006. In the summer of 2007, he was a Visiting Professor with Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland. In 2009–2010, he is a Visiting Professor with Berkeley Wireless Research Center, University of California, Berkeley, investigating on ultra-low power circuits and wireless sensor nodes. Since 2001, he has been teaching undergraduate and graduate courses on advanced VLSI digital design, mircoelectronics and basic electronics. He has authored or coauthored more than 140 publications on journals (50+, mostly IEEE Transactions) and conference proceedings. Two of them are among the 25 most downloaded TVLSI papers in 2007 (respectively, 10th and 13th). He is the coauthor of the book *Model and Design of Bipolar and MOS Current-Mode Logic: CML, ECL and SCL Digital Circuits* (Springer, 2005). His primary research interests include the modeling and the optimized design of CMOS high-performance, low-power, and ultra low-power digital circuits, aritmetic and cryptographic circuits, interconnect modeling, design/modeling for variability-tolerant and low-leakage VLSI circuits, circuit techniques for emerging technologies. He is the Director of the Electronics Laboratory, University of Siena (site of Arezzo).

Prof. Alioto is a member of the HiPEAC Network of Excellance. He is the chair elect of the "VLSI Systems and Applications" Technical Committee of the IEEE Circuits and Systems Society, for which he is also a Distinguished Lecturer. He is regularly invited to give talks to academic institutions, conferences, and companies throughout the world. He has served as a member of various conference technical program committees (ISCAS, PATMOS, ICM, ICCD, CSIE) and Track Chair (ICECS, ISCAS, ICM, ICCD). He serves as an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, as well as on the *Microelectronics Journal*, the *Integration–The VLSI Journal*, and the *Journal of Circuits, Systems, and Computers*. He is Guest Editor of the Special Issue "Advances in Oscillator Analysis and Design" of the *Journal of Circuits, Systems, and Computers* (2009).

Luca Giancane received the M.S. (*summa cum laude*) degree in electronic engineering from the Università di Roma "La Sapienza," Rome, Italy, in 2006. He is currently working toward the Ph.D. degree in the Dipartimento di Ingegneria Elettronica, Università di Roma "La Sapienza," Rome.

In 2003, he was with the Selex SI, Rome, as a Digital Designer. In 2005, he was with the Security and Chipcard ICs Department, Infineon Technologies AG, Graz, Austria. His research interests include the analysis and design of mixed-signal architectures to counteract power analysis attacks and statistical modeling of logic gates.

Dr. Giancane is the recipient of degree prizes such as best Italian thesis on security topics from COPIT Onlus and from ACCENTURE SPA, Rome, Italy, in 2007.

Giuseppe Scotti was born in Cagliari, Italy, on April 14, 1975. He received the M.S. and Ph.D. degrees in electronic engineering from the Università di Roma "La Sapienza," Rome, Italy, in 1999 and 2003, respectively.

He is currently doing postdoctoral work with the Dipartimento di Ingegneria Elettronica, Università di Roma "La Sapienza." His research interests include the design methodologies of high-yield analog and digital integrated circuits, the design techniques of high-speed circuits for optical communication systems, and the design of cryptographic hardware.

Alessandro Trifiletti was born in Rome, Italy, in 1959. He received the bachelor's degree in electronic engineering from the Università di Roma "La Sapienza", Rome, Italy

In 1991, he joined the Dipartimento di Ingegneria Elettronica, Università di Roma "La Sapienza," as a Research Assistant and is currently an Associate Professor. His research interests include high-speed circuit design techniques and III–V device modeling.