IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 15, 2021, accepted August 9, 2021, date of publication August 18, 2021, date of current version August 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105635

Analysis of Random Body Bias Application in FDSOI Cryptosystems as a Countermeasure to Leakage-Based Power Analysis Attacks

KENNETH PALMA AND FRANCESC MOLL, (Senior Member, IEEE)
Department of Electronic Engineering, Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain

Corresponding author: Kenneth Palma (kenneth.palma@upc.edu)

The research leading to these results has been supported by the Spanish AEI - Agencia Estatal de Investigación - grant number PID2019-103869RB-C33/AEI/10.13039/501100011033, by Secretaria d'Universitats i Recerca of Generalitat de Catalunya and by the European Social Fund.

ABSTRACT This paper analyses a novel countermeasure to Leakage Power Analysis Attacks based on the application of a random Body Bias voltage level at the beginning of the encryption process. The countermeasure effectiveness is established through the development of a theoretical model of the Pearson correlation coefficient in the presence of a varying body bias under both noiseless assumptions and in the presence of algorithmic noise, and through simulations on a partial cryptosystem implemented in 28 nm FDSOI technology. A study of the effect of averaging power measurements is also developed and contrasted against Monte Carlo simulations of the countermeasure scheme, effectively providing a floor for the increase in required measurements to identify the secret key.

INDEX TERMS Body bias, correlation power analysis, countermeasure, FDSOI, leakage current, power analysis attack, side-channel.

I. INTRODUCTION

As technology nodes progress further into smaller nanometric scales, short-channel effects worsen, one of the most impactful short-channel effects being the increase of leakage currents. Increasing leakage currents implies an increase in static power consumption and this can have a great impact, particularly for portable devices. In the case of cryptographic systems, increasing leakage currents also implies the enhancement of a side-channel with potential to inadvertently leak information to an attacker.

To limit the consequences of increased leakage current, solutions to these effects have been found in the form of transistors with modified structures: FinFETs and Fully Depleted Silicon On Insulator (FDSOI) transistors. FDSOI transistors have been shown to exhibit better electrostatic control of the channel, with smaller variations of the threshold voltage along the channel, smaller parasitic capacitances and, overall, smaller short-channel effects [1], [2].

Among their characteristics, and given their structure, is the capability of having their threshold voltage dynamically modified through its fourth terminal with a much higher dynamic range than in conventional bulk devices. This, among other effects (decreased charging times, increased transconductance, etc.) allow the introduction of varying leakage currents.

Leakage-based side channel attacks, on the other hand, have been shown to convey sufficient information for successful attempts at obtaining the secret key [3], [4]. In [5], the authors performed an analysis of the linear dependence between the Hamming Weight of a register array and their leakage current consumption for a variety of registers, essentially establishing the basic model to perform Correlation or Differential Power Analysis Attacks utilizing leakage current as a Side Channel. In [5], the authors base their study on the application of the Hamming Weight model pertaining to the correlation coefficient initially described in [6].

Since then, several countermeasures have been proposed to hinder the acquisition of the secret key from leakage-based PAA. Some of these countermeasures attempt to reduce the SNR of significant signals by introducing uncorrelated sources of noise, while others attempt to decorrelate the leakage consumption of registers from the data they store [7]–[12].

The associate editor coordinating the review of this manuscript and approving it for publication was Junaid Arshad.

It is, however, understood that Power Attacks that rely on static power consumption typically convey poorer results than their dynamic power counterpart. The need to DC couple static power measurements increases the noise floor of the traces obtained. This factor, coupled with comparatively smaller signals than those obtained through a dynamic power side-channel results in noisier data with smaller signal levels.

Despite this fact, several circumstances can enhance the effectiveness of leakage-based PAA. Extending the clock period allows the acquisition of an increasing number of samples of the signals of interest at closely spaced intervals of time. This ''oversampling'' can be used to obtain averaged measurements, in what is called inter-trace averaging, to reduce overall non-algorithmic noise. This fact alone can reduce the effectiveness of masking countermeasures, which rely on the noisiness of the measurements for their security [13]. In fact, authors on [14] report the need to obtain fewer traces to obtain the secret key in an ASIC implementation of a cryptosystem with masking countermeasure when using static power as a side-channel.

On the other hand, if the attack can be performed in thermally-controlled conditions, inducing higher temperatures can increase the intensity of the signals obtained.

The ongoing research on these topics [4], [15], and the possibility of effectively combining leakage-based with dynamic Power Analysis Attacks (PAA) [16] prompts the development of secure systems that can resist these vulnerabilities.

In this paper we explore the feasibility of utilizing a varying body bias as a countermeasure for Leakage-based PAA in systems implemented in FDSOI technology. A similar approach has recently been presented in [17]. In their paper, the authors perform empirical testing of the countermeasure based on the random application of body bias on a RISC-V microcontroller implemented in FDSOI technology executing an AES-128 encryption process. The authors are able to prove a significant increase in the number of required measurements to disclose the secret key.

In this paper, a mathematical model for the effectiveness of the countermeasure based on technological and countermeasure parameters is developed and contrasted with electrical and numerical simulations. Section II presents a theoretical background of how differing body bias can help to meaningfully decorrelate processed data with consumed leakage power. Section III studies the dependency between register's leakage current and body bias value in FDSOI technologies. In Section IV a potential countermeasure scheme based on the technological properties and concepts introduced in the previous sections is presented, while in Section V a mathematical model for the decorrelation achieved by the countermeasure is derived. Sections VI and VII showcase results obtained from electrical and numerical simulations, including the effect of averaging. In Section VIII the model is expanded to include sources of algorithmic noise, with results from numerical simulations. Finally, Section IX present the conclusions of the paper.

II. CONCEPTUAL OVERVIEW

In the n-bit register slice model introduced in [5], the leakage current is a linear equation of the Hamming Weight of the value stored in the register:

$$I_{leak}(HW) = n \cdot I_0 + \epsilon \cdot HW \tag{1}$$

where $n \cdot I_0$, the y-intercept, is a constant depending on the number of bits of the register under study and the leakage current consumed when a flip-flop is storing a 0. The slope of the function is ϵ, defined as $\epsilon = I_1 - I_0$, the difference in leakage current consumed when a flip-flop stores a 1 and a 0. HW is the Hamming Weight: the total number of 1's stored in the register array.

In a block cryptosystem, the Hamming Weight is the result of a boolean non-linear function implemented by a substitution box (S-box). Based on the attackers knowledge of the encrypting algorithm, this allows the formulation of a power consumption model that can help test secret key hypothesis. If the secret key is correctly guessed, every evaluation of the function $HW(S(x_i \oplus k))$ for every possible plaintext will yield the actual values of the Hamming Weight, which will correlate with measured values of leakage power.

The leakage current consumed by the register array can be considered a random variable, linearly related to the Hamming Weight through the input plaintext, which can also be considered a random variable.

The Pearson Correlation Coefficient (PCC) tests the degree of linear dependency between two random variables. As such, it is a powerful metric to test correct-key hypothesis. The PCC as a metric to perform Power Analysis Attacks was introduced in [6], and is the basis for Correlation Power Analysis attacks (CPA).

The PCC, defined in equation (2), establishes the degree of linear correlation between two random variables X and Y. The candidate secret key that exhibits the highest PCC is generally accepted as the correct secret key, if the attack is performed appropriately.

$$\rho_{Y,X} = \frac{Cov(Y,X)}{\sqrt{Var(Y)}\sqrt{Var(X)}} \tag{2}$$

In these expression Cov denotes the Covariance function and Var the Variance function.

Consider the dummy cryptosystem depicted in Fig. 1, formed by a register array of 8 bits and an equal number of XOR gates.

In this dummy cryptosystem the S-Box has been omitted and the power consumption model is simply based on the XORing between the secret key and input plaintext:

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

where X is a random variable representing the input plaintext and k is the deterministic secret key. Therefore, the Hamming Weight of the content of the register slice, $HW(f,k)$, is also considered a random variable.

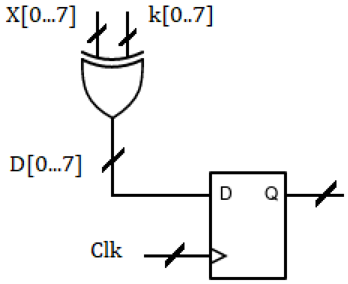

FIGURE 1. Schematic view of the dummy cryptosystem under study, comprising an array of 8 registers and the corresponding number of XOR gates, where D is the vector of bits to be stored in the register array after a clock pulse and Q the output or intermediate output of the dummy cryptosystem.

Under a correct key assumption, the PCC between the leakage current of the power consumption model I_{leak} (Equation (1)) and the Hamming Weight calculated from plaintext X and guessed key k in Equation (3) can be shown to be $\rho_{I_{leak},HW} = 1$. This result holds true if the circuit is considered noiseless. This is the basis for Correlation Power analysis attacks.

Conceptually, this means that, if the secret key is known, every measurement of I_{leak} versus Hamming Weight for different values of plaintext X falls in a straight line, as defined by Equation (1), while an incorrect key would produce values of I_{leak} outside of such curve, reducing the linear correlation between variables (Fig. 2).

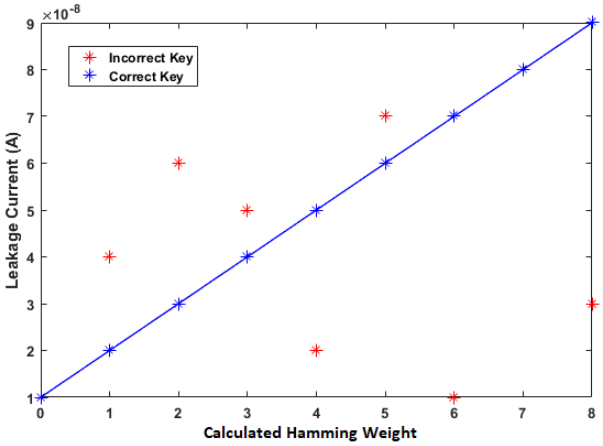

FIGURE 2. Measured leakage current vs hamming weight of different plaintext X in an 8-bit register. The solid line depicts equation (1) for a given register, while the markers represent measured current values for different plaintexts displayed according to the calculated hamming weight with a correct key (blue) and an incorrect key (red).

However, if we can consider the possibility that I_0, so far a constant, can become a random variable, we obtain a collection of curves:

$$\begin{aligned} I_{Leak1}(X,k) &= n \cdot I_{01} + \epsilon \cdot HW \\ I_{Leak2}(X,k) &= n \cdot I_{02} + \epsilon \cdot HW \\ &\cdots \\ I_{Leakn}(X,k) &= n \cdot I_{0n} + \epsilon \cdot HW \end{aligned} \tag{4}$$

where I_0 is now a discrete random variable that can adopt values $\{I_{01}, I_{02}, \ldots, I_{0n}\}$ with probability $P[I_0 = I_{0i}] = p_i$, for $1 \leq i \leq n$.

Provided that I_0 and HW are independent variables and their variances are well defined, the PCC between the register's leakage current and the Hamming Weight now becomes:

$$\rho_{I_{leak},HW} = \frac{\epsilon \cdot \sigma_{HW}}{\sqrt{(n^2 \cdot \sigma_{I0}^2 + \epsilon^2 \cdot \sigma_{HW}^2)}} \tag{5}$$

Assuming that the attacker has no means of accessing the value I_0, this means that for a sufficiently large number of possible values of I_0, having the secret key can be indistinguishable from having an incorrect key, as the values of I_{leak} do not fall in a single curve.

This is exemplified in Fig. 3, where a collection of such curves is shown. Under a correct key assumption, the markers represent the correct evaluation of the Hamming Weight for each possible plaintext. However, it can be seen that much of the linear relation is lost.

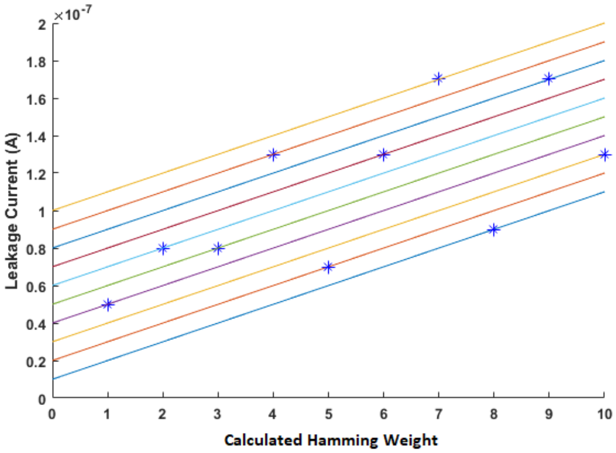

FIGURE 3. Collection of register leakage current vs hamming weight curves. Each line represents a possible realization of equation (1) for different values of I_0. The markers represent measured current values for different plaintexts displayed according to the calculated hamming weight with a correct key.

III. REGISTER LEAKAGE CURRENT AS A FUNCTION OF BODY BIAS IN FDSOI TECHNOLOGY

An exploration on the effect of body bias on the leakage current of FDSOI registers is performed. The registers are implemented in 28 nm, Low Threshold Voltage (LVT), Flipped Well transistors.

These transistors, shown in Fig 4 can be biased through their back gate terminals, noted as BBnmos and BBpmos. As the application of Forward body bias reduces the threshold voltage of transistors, in our study, transistors are symmetrically biased to avoid an imbalance between driving capabilities of N- and PMOS components, meaning that $Vbb_{nmos} = -Vbb_{pmos}$.

Figure 5 shows the variation of the different leakage currents that a register implemented in this technology can incur,

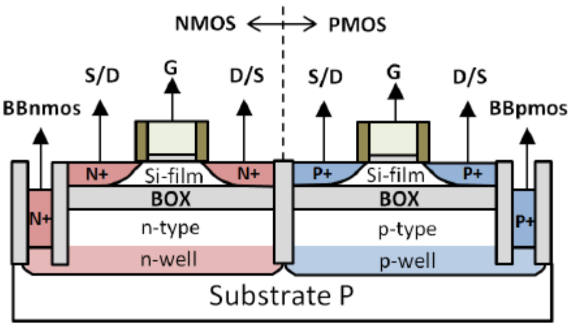

FIGURE 4. FDSOI forward body bias (FBB) transistors' structure.

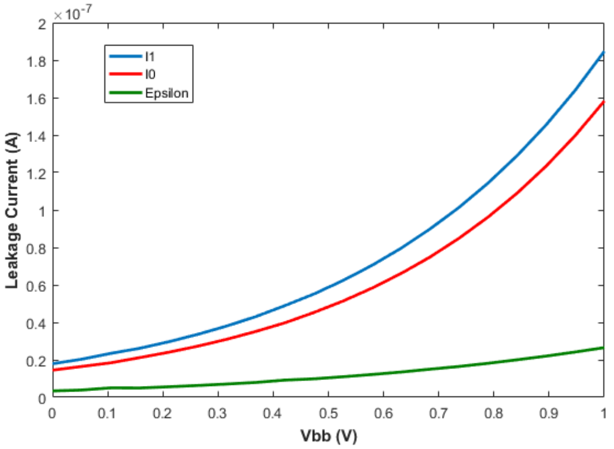

FIGURE 5. Single register leakage currents as a function of body bias. In blue, when the register is storing a 1. In red, when the register stores a 0. In green, ϵ, the difference between the two.

as a function of the body bias. The curves on Fig. 5 are obtained by storing a 1 or a 0 in a single flip-flop and performing a transient simulation. A parametric analysis on the body bias ranging from 0 to $|V_{dd}| = 1$ V, the maximum nominal voltage for the technology, is then performed, and the leakage current value noted after a single clock pulse.

It can be seen that both I_1 and I_0 are monotonically increasing functions, with $I_1(|V_{bb}|) > I_0(|V_{bb}|)$ for all possible values of $|V_{bb}|$. The third curve is obtained by computing $\epsilon(|V_{bb}|) = I_1(|V_{bb}|) - I_0(|V_{bb}|)$. A curve fitting analysis shows that all three curves are of exponential nature and of the form:

$$f(|V_{bb}|) = a \cdot e^{(b \cdot |V_{bb}|)} \tag{6}$$

where a and b are constants.

This result is consistent with simulations and experimental results described in the literature, given the linear dependence of the Threshold Voltage on Forward body bias for both NMOS and PMOS transistors [2].

With these considerations, the leakage current consumed by a register array implemented in FDSOI technology becomes a function of body bias. Equation (1) now becomes a function of two variables:

$$I_{leak}(HW, |V_{bb}|) = n \cdot I_0(|V_{bb}|) + \epsilon(|V_{bb}|) \cdot HW \tag{7}$$

where $\epsilon(|V_{bb}|)$ and $I_0(|V_{bb}|)$ are defined as:

$$\epsilon(|V_{bb}|) = a_1 \cdot e^{b_1 \cdot |V_{bb}|} \tag{8}$$

$$I_0(|V_{bb}|) = a_2 \cdot e^{b_2 \cdot |V_{bb}|} \tag{9}$$

It becomes clear from equation (7) that I_{leak} is represented by an infinite set of curves, as opposed to a discrete one as seen in Fig. 3, given the continuous, analog nature of V_{bb}. Note, also, that if we consider ϵ to be the slope of each curve, the slope also varies with V_{bb}.

IV. PROPOSED COUNTERMEASURE

The proposed countermeasure is based on the notion that information about the secret key is only conveyed when multiple measurements of the leakage power, obtained from different plaintexts, are compared or studied on the whole. That is, a single power measurement with one plaintext yields no information regarding the secret key.

The analysis performed in Section II in combination with Equation (7) allows us to consider a countermeasure based on the random adoption of a body bias value at the beginning of each encryption process. This value of body bias is set before the encryption algorithm begins and is maintained throughout the process. At the beginning of a new encryption process, a new value is chosen, at random and independently. The following analysis describes this principle.

At this point, we drop the $|V_{bb}|$ notation for simplicity. Consider that V_{bb} is a discrete random variable of the form:

$$V_{bb}(S) = V_{bb_Q} + \Delta V_{bb} \cdot S \tag{10}$$

where V_{bb_Q} is an arbitrary constant, ΔV_{bb} is the step size with which the value of the body bias can vary, and S is a random variable which can adopt integer values between $[-s_{max}, s_{max}]$.

The random variable S follows a discrete uniform distribution with $P[s = i] = \frac{1}{2s_{max}+1}$ for all i's, such that $-s_{max} \leq i \leq s_{max}$. It can be seen that the expected value of $E[S] = 0$. This allows the distribution of $V_{bb}(S)$ to be well defined, with expected value equal to $E[V_{bb}(S)] = V_{bb_Q}$.

With a uniform distribution, the entropy of S is maximized. Thus, in the event that S could inadvertently leak information about the system, the least possible amount of information would be conveyed. This distribution also facilitates mathematical derivations of the model described below.

The value of S is set at the beginning of each encryption process, before the algorithm starts execution.

With these considerations, the leakage current consumed by an array of registers is now:

$$I_{leak}(HW, S) = n \cdot I_0(S) \ + \epsilon(S) \ \cdot HW \tag{11}$$

where $\epsilon(S)$ and $I_0(S)$ can now be expressed as:

$$\epsilon(S) = a_1 \cdot e^{b_1 \cdot (V_{bb_Q} + \Delta V_{bb} \cdot S)} \tag{12}$$

$$I_0(S) = a_2 \cdot e^{b_2 \cdot (V_{bb_Q} + \Delta V_{bb} \cdot S)} \tag{13}$$

The values that V_{bb} can adopt are thus discretized and so is the set of curves of I_{leak}, which contains $2s_{max} + 1$ elements.

V. CORRELATION

In this section, the PCC between the leakage current and the Hamming Weight is derived taking into consideration the countermeasure defined in the previous section.

The derivation is made considering a dummy cryptosystem like the one shown in Fig. 1. First, some assumptions are made regarding the probability distribution of the plaintext.

With an n-bit register array, each possible plaintext is a boolean sequence $X : \{0, 1\}^n$. We assume that each element in such sequence follows a uniform probability distribution with equal probability $\frac{1}{2}$ of either being a 1 or a 0.

Thus, the expected value and the variance of the Hamming Weight can be shown to be, respectively, $\mu_{HW} = \frac{n}{2}$ and $\sigma^2_{HW} = \frac{n}{4}$.

It is also assumed that the correct key has been chosen, so that the Hamming Weight is always correct for each possible input plaintext. The system is also assumed to be noiseless.

Using the definition of the PCC, we can calculate how the countermeasure impacts the ability to distinguish the secret key. The PCC between the Hamming Weight of and the Leakage Current consumed by a register slice of n bits can be expressed as:

$$\rho_{HW,I} = \frac{\mu_\epsilon \sigma_{HW}}{\sqrt{Var(I_{leak})}} \tag{14}$$

where we used I_{leak} as shown in Equation (11), with μ_ϵ being the expected value of $\epsilon(S)$.

The expression of the variance of the leakage current, which appears in the denominator of Equation (14) is cumbersome. It can be broken down into three components:

$$Var(I_{leak}) = p1 + p2 + p3 \tag{15}$$

where, taking into account that HW and $\epsilon(S)$ are independent random variables:

$$\begin{aligned} p_1 &= Var(HW \cdot \epsilon(S)) \\ &= \sigma^2_{HW}\sigma^2_\epsilon + \mu^2_\epsilon \sigma^2_{HW} + \mu^2_{HW}\sigma^2_\epsilon \end{aligned} \tag{16}$$

Also:

$$p_2 = Var(n \cdot I_0(S)) = n^2 \cdot \sigma^2_{I0} \tag{17}$$

And finally:

$$\begin{aligned} p_3 &= 2 \cdot Cov(HW \cdot \epsilon(S), n \cdot I_0(S)) \\ &= 2 \cdot n \cdot \mu_{HW} \cdot Cov(\epsilon(S), I_0(S)) \end{aligned} \tag{18}$$

where σ^2_ϵ is the variance of $\epsilon(S)$ and μ_{I0} and σ^2_{I0} the expected value and variance of $I_0(S)$.

To obtain an analytical solution of equation (14) the expected value and variance of both $\epsilon(S)$ and $I_0(S)$ must be calculated. In the following derivation, we adopt a change in nomenclature, with $E[\cdot]$ representing the expected value of a function.

Given that S follows a discrete uniform probability distribution and both $\epsilon(S)$ and $I_0(S)$ are exponential functions as shown in Equations (12) and (13), the expected value of $\epsilon(S)$ can be calculated as:

$$E[\epsilon(S)] = \frac{1}{2s_{max}+1} a_1 e^{b_1 V_{bbQ}} \sum_{i=-s_{max}}^{s_{max}} e^{b_1 \Delta V_{BB} \cdot i} \tag{19}$$

Which defines a geometric series that can be solved as:

$$\begin{aligned} E[\epsilon(S)] = \frac{1}{2s_{max}+1} a_1 \cdot e^{b_1 V_{BBq}} \\ \cdot e^{-b_1 \Delta V_{BB} s_{max}} \left(\frac{1 - e^{b_1 \Delta V_{BB}(2s_{max}+1)}}{1 - e^{b_1 \Delta V_{BB}}}\right) \end{aligned} \tag{20}$$

A similar approach is used to calculate $E[\epsilon(S)^2]$. With these, the variance of $\epsilon(S)$ can be determined using the definition:

$$Var(\epsilon(S)) = E[\epsilon(S)^2] - E[\epsilon(S)]^2 \tag{21}$$

The same procedure is used to determine the expected value and variance of $I_0(S)$.

Note that the Covariance between $\epsilon(S)$ and $I_0(S)$ appears in Equation (18). To compute it, we used the definition:

$$Cov(\epsilon(S), I_0(S)) = E[\epsilon(S) \cdot I_0(S)] - E[\epsilon(S)]E[I_0(S)] \tag{22}$$

Both $E[\epsilon(S)]$ and $[I_0(S)]$ are already well defined. Since each realization of S is independent from each other, the covariance between experiments is zero. Thus, only the intra-experiment covariance is meaningful. In this way, the covariance can be calculated by noting that $E[\epsilon(S) \cdot I_0(S)]$ is simply:

$$E[\epsilon(S) \ \cdot I_0(S)] = \frac{1}{2s_{max}+1} \sum_i \epsilon(s_i) I_0(s_i) \tag{23}$$

Thus, the Pearson Correlation Coefficient between the Hamming Weight and the Leakage current of a register array in the presence of a random body bias countermeasure can be analytically expressed. Equation (14) is a function of technological parameters (the constants in equations (13)), the number n of registers under attack, the quiescent point of the body bias V_{bbQ}, the step size ΔV_{bb} and the number of available steps $2s_{max} + 1$.

VI. RESULTS

A. THEORETICAL RESULTS

In this section, equation (14) is plotted in a variety of conditions.

A D-flip-flop is chosen from the family of available flip-flops in the library. Through a parametric analysis of the body bias and a curve-fitting analysis, the constants that define the exponential functions (Equations (12) and (13)) are extracted.

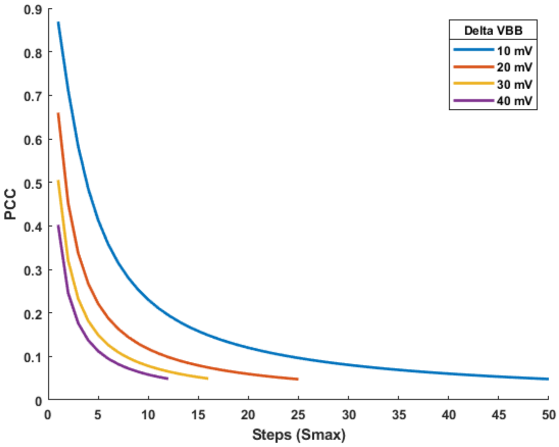

FIGURE 6. PCC between the leakage current of a register array of 8 bits and its hamming weight in the presence of the countermeasure as a function of the number of available steps for different values of step size (ΔV_{bb}).

The nominal dynamic range of the body bias ranges from 0 to V_{dd}, with a maximum $V_{dd} = 1\ V$. Thus, V_{bbQ} is set at $\frac{V_{dd}}{2}$ so that the random body bias can span the entire dynamic range symmetrically.

Figure 6 plots the Pearson Correlation Coefficient (PCC) calculated with Equation (14) between the leakage current and the Hamming Weight as a function of steps (s_{max}) for different values of ΔV_{bb}, as defined in Equation (10).

Equation (14) is undefined for a zero number of steps. However, a limit analysis shows that, in all cases, the PCC tends to 1 as $2s_{max} + 1$ approaches 0, as was expected for the case where I_0 is constant.

It can be seen that the PCC is greatly reduced as the number of steps increases, and with it the dynamic range of body bias. For the maximum range of 0 to 1V, the obtained value of PCC is near 0.048 independently of the body bias step.

Since utilizing the entirety of the body bias' dynamic range provides the greatest decrease in the PCC, it is of interest to explore the effectiveness of the countermeasure with a fixed, maximum dynamic range $DR = V_{dd} - 0$.

In this case, ΔV_{bb} and s_{max} are related (Equation 24), and we can plot the values of the PCC for increasing number of steps (s_{max}). The results are plotted along the entropy exhibited by the random variable S, which can be shown to be $H(S) = \log_2(\frac{1}{2 \cdot s_{max}+1})$.

The results can be seen in Fig 7.

$$\Delta V_{bb} = \frac{DR}{2 \cdot s_{max}} \tag{24}$$

In this graph, a counterintuitive increase of the PCC with respect to the number of steps, ranging from 0.036 for $s_{max} = 1$ to 0.04854 for large number of steps is seen. However, it should be noted that the entropy is also minimum for $s_{max} = 1$, which indicates a potential information leak for low number of steps. As the number of steps increases, the PCC stabilizes and the entropy of S continues to grow.

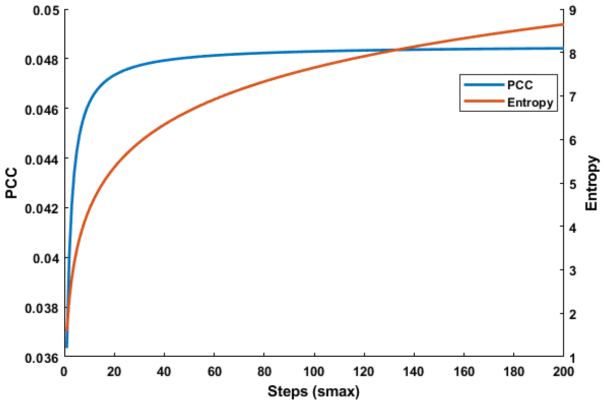

FIGURE 7. Left axis: PCC between the leakage current of a register array of 8 bits and its hamming weight in the presence of the countermeasure as a function of steps with fixed, maximum dynamic range. Right axis: Entropy of S as a function of s_{max}.

B. SIMULATION RESULTS

To test the validity of the theoretical approach, we perform a simulated CPA on a dummy cryptosystem as seen in Fig. 1.

The Cadence Virtuoso ADE is used to perform the simulated attack, using standard logic cells of the 28 nm, FDSOI library.

An 8-bit register array, with the corresponding 8 XOR gates is set. The secret key is set to 10101010_2, or 170 in base ten.

256 transient simulations are performed, where all 256 possible plaintext values are inputted to the XOR gates and stored in the registers after a single clock pulse. The leakage current is measured only for the register array, to simulate noiseless conditions. Each leakage current value is stored along the inputted value of the plaintext.

Two sets of simulations are performed. A set of 256 simulations in which no countermeasure is applied (fixed Body Bias value of 0 V), and a set of 256 simulations where S is set at random at the beginning of each simulation using a pseudo-random number generator function supplied by the Virtuoso function library so that a different value of body bias is applied to each plaintext.

In the set of simulations with applied countermeasure, the Body Bias DC value V_{bbQ} is set at 0.5 V, ΔV_{bb} at 20 mV, with $s \in [-25, 25]$

Once the simulations are complete, a vector with the leakage current values for each inputted plaintext is acquired. The vector is used to compute the correlation between the leakage current and each possible candidate key, for a total of 256 possible keys.

Figure 8 shows the PCC for each possible key when no countermeasure is applied. It can be seen that the secret key, 170, is easily identified, along with its binary complement, 85, as both cases present the maximum value of PCC, which is 1 in this case. This perfect correlation is consistent with the deterministic nature of the simulation under no noise conditions.

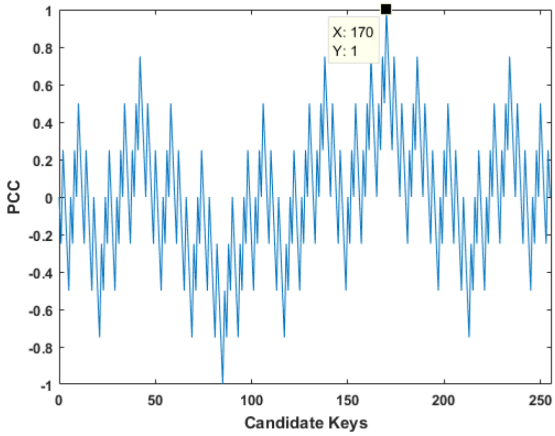

FIGURE 8. PCC between the leakage current and the hamming weight of the register array under attack for each possible candidate key in the absence of the proposed countermeasure. The correct key presents the highest correlation, with a value of 1.

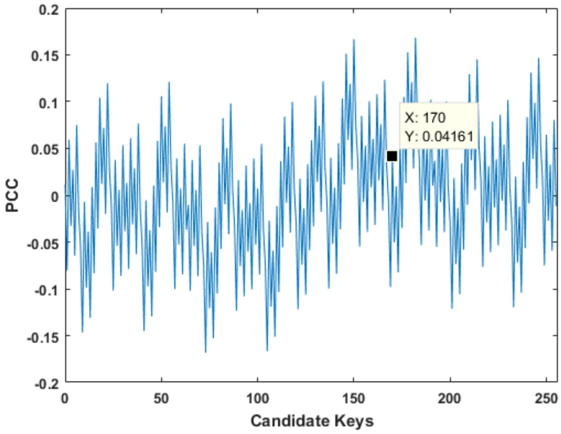

FIGURE 9. PCC between the leakage current and the hamming weight of the register array under attack for each possible candidate key in the presence of the proposed countermeasure. The PCC for the correct key is greatly diminished, as well as the maximum value for the PCC.

Figure 9 shows the CPA for all possible candidate keys when the countermeasure is applied as described above. It can be seen that not only is the secret key not identified but also the maximum correlation is greatly decreased.

Note also that S adopts in this experiment a maximum of $2s_{max} + 1 = 51$ values and that only 256 simulations are performed. Notwithstanding the possible bias of the random generating function provided by Virtuoso, the distribution of S most likely does not resemble a discrete uniform distribution for such a low number of experiments. Thus, Fig. 9 is just one among many possible results.

VII. NUMERICAL SIMULATIONS: EFFECT OF AVERAGING

The main weakness of this countermeasure stems from the fact that the expected value of I_{leak} is well defined even in the presence of countermeasure. Thus, if sufficient measurements are taken and averaged, the $2s_{max} + 1$ straight lines that define the set of I_{leak} curves converge to the expected value $E[I_{leak}]$. In this section we evaluate the effect of leakage current values averaging on the PCC and on CPA attacks when several experiments are performed for each possible plaintext.

Firstly, we evaluate the effect of averaging on Equation (14) by solving it under the assumption of noise averaging. The underlying idea is to run N independent encryption processes for every possible plaintext value, and evaluate the impact on Equation (14), comparing the results to electrical simulations.

However, since performing a set of 256 transient simulations is time consuming even given the simplicity of the dummy cryptosystem under attack (Fig. 1), for large values of N, performing $N \cdot 256$ electrical simulations can become infeasible.

Thus, given the deterministic nature of the leakage current Equation (11) under noiseless conditions, as demonstrated by the perfect correlation seen in Fig. 8, we opt to numerically simulate the analog behavior of the dummy cryptosystem. The numerical simulation of the analog behavior of the system allows us to simulate a CPA attack where each leakage current measurement, for every plaintext, is repeated N times and then averaged. This can be done given that the technological parameters of the registers under study are known (the constants in Equations 12 and 13), and thus these CPA's provide comparable results to the ones that would be obtained through electrical simulations in the Virtuoso ADE.

In these numerical simulations, each input plaintext is XORed with the correct secret key, and the resulting Hamming Weight is evaluated. As per Equation (3):

$$HW_i = f(i \oplus k) \tag{25}$$

for $0 \leq i \leq 255$.

Then, we simulate an encryption process with countermeasure by generating a random value of S. The leakage current function (Equation (11)) is then numerically solved for these values of the Hamming Weight and the realization of $S = s$. This process is repeated N times for every plaintext element, such that a leakage current value is obtained for each one:

$$\begin{gathered} I_{l1,i}(HW_i, s_1) \\ I_{l2,i}(HW_i, s_2) \\ \cdots \\ I_{l_N,i}(HW_i, s_N) \end{gathered} \tag{26}$$

Then, the result is averaged:

$$\hat{I}_{li}(HW_i, \hat{S}) = \frac{1}{N} \sum_{j=1}^{N} I_{l_j,i}(HW_i, s_j) \tag{27}$$

Finally, the PCC between the vector of 256 averaged leakage current values and the calculated Hamming Weight according to the input plaintext for every candidate key is evaluated, thus numerically simulating a CPA attack in the presence of the countermeasure with noise averaging.

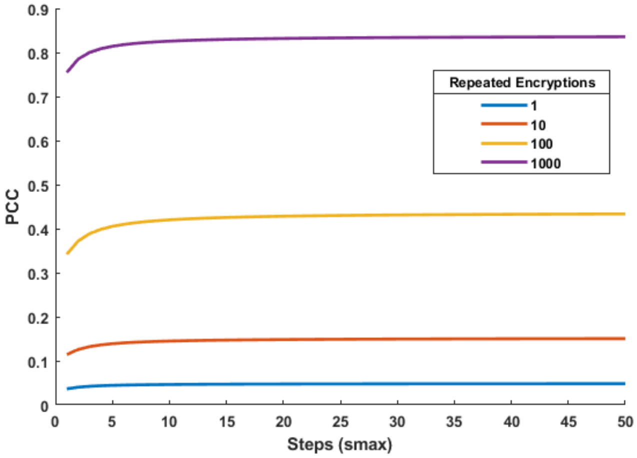

FIGURE 10. Effect of averaging on the PCC between the leakage current and the hamming weight of the register array in the presence of the proposed countermeasure with a fixed, maximum body bias DR of 1 V. It can be seen how averaging undermines the effect of the countermeasure for large number of repeated encryption processes.

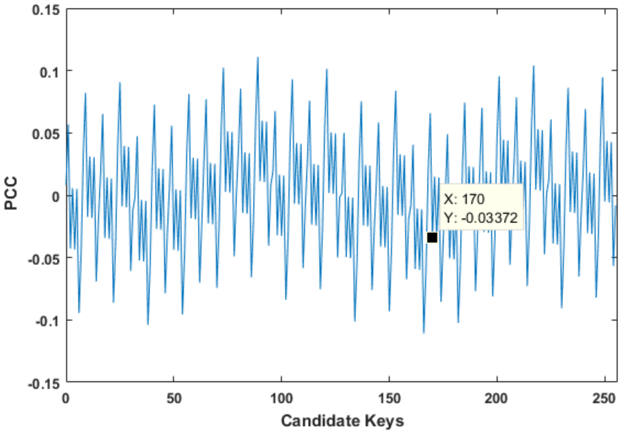

FIGURE 11. PCC between the numerically simulated leakage current and the hamming weight of the theoretical register array under attack for every candidate key in the presence of the proposed countermeasure. The results are obtained with one single encryption process per plaintext.

A. EFFECT OF AVERAGING ON THE PCC

To determine the effect that averaging has on the PCC, Equation (14) must be solved under the assumption of leakage current averaging.

Through averaging, the expected values of the different terms in Equation (14) are expected to remain the same, while the variance introduced by the countermeasure is to be reduced. An approximation can be made regarding the reduction of the variance, such that:

$$\sigma_\epsilon^2 \rightarrow \frac{\sigma_\epsilon^2}{N} \tag{28}$$

$$\sigma_{I_0}^2 \rightarrow \frac{\sigma_{I_0}^2}{N} \tag{29}$$

$$Cov(\epsilon(S), I_0(S)) \rightarrow \frac{Cov(\epsilon(S), I_0(S))}{N} \tag{30}$$

These terms are substituted in Equation (14) and plotted for several numbers of repeated plaintext encryptions (N) as a function of s_{max} for a fixed body bias dynamic range of 1 V.

Figure 10 showcases the increase on the PCC as the number of repeated encryptions increase. Table 2 presents the value of PCC when $s_{max} = 25$ for different number N of repeated encryptions. In both cases, the Dynamic Range is fixed at the maximum 1 V for the technology.

TABLE 1. PCC for maximum dynamic range of the body bias for different number of averaging encryption processes when $s_{max} = 25$.

N	PCC
1	0.0476
10	0.1489
100	0.4300
1000	0.8331

It can be seen that while the PCC increases with repeated number of averaged encryption processes, the behaviour of the PCC as a function of s_{max} for a fixed DR is similar in all cases, regardless of the number of repetitions. That is, a small increase in the PCC with increasing number of steps for small values of s_{max}, followed by a convergence of the PCC to a given value.

B. NUMERICAL SIMULATIONS

Numerical simulations are performed as described in the previous subsection to generate large amounts of I_{leak} values for each plaintext. This allows the exploration of the effect of averaging in CPA attacks.

Four cases are contemplated: no averaging, and 10, 100 and 1000 repeated encryptions per input plaintext. The results can be seen in Figures 11 through 14.

It can be seen that, as the number of repeated encryptions increases, the CPA produces results that more correctly represent the simulated case when no countermeasure is applied (Fig. 8). Note also how the theoretical results derived in the previous subsection, as shown in Table 1, are very close to the results obtained through numerical simulations, as shown in Figures 11 through 14.

To test the effect of increased PCC with averaging in the ability to correctly identify the secret key, a series of 100 distinct runs are then simulated for each collection of N averages. The amount of times the secret key presents the highest PCC is noted. The frequency with which the secret key is disclosed for different number of averaging encryption processes can be seen in Table 2.

Note that 100 repeated encryption processes appear to be enough to identify the secret key in most instances.

VIII. ALGORITHMIC NOISE

So far, the analysis performed to derive the properties and characteristics of the countermeasure has assumed a trivial

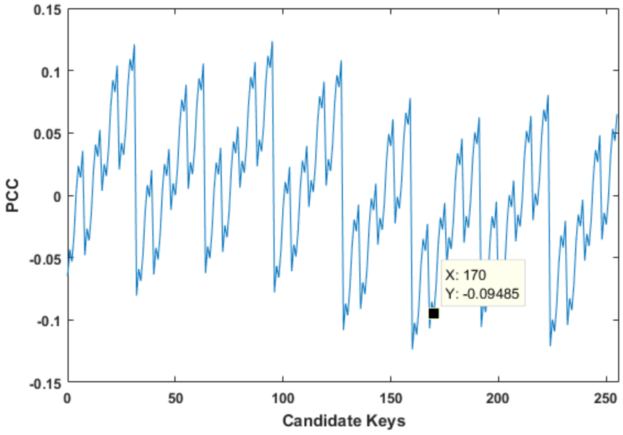

FIGURE 12. PCC of the numerically simulated and averaged leakage current and the hamming weight, with 10 encryption processes per plaintext.

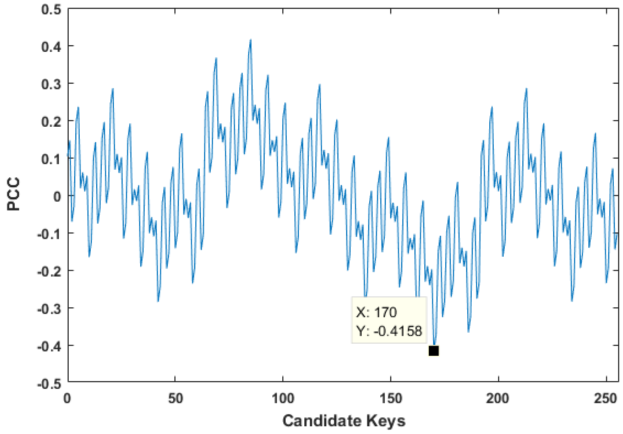

FIGURE 13. PCC of the numerically simulated and averaged leakage current and the hamming weight, with 100 encryption processes per plaintext.

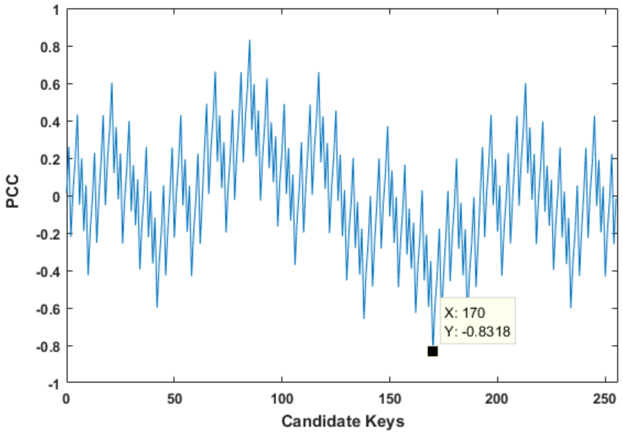

FIGURE 14. PCC of the numerically simulated and averaged leakage current and the hamming weight, with 1000 encryption processes per plaintext.

cryptosystem with only 8 bits. However, a more realistic scenario is that of a cryptosystem that processes several bytes.

TABLE 2. Success ratio of secret key identification in numerical simulations under different number of averaging samples.

N	Success Rate
1	0
10	0.11
100	0.97
1000	1

Consider a dummy cryptosystem that comprises $n+m$ bits of data, with an $(n+m)$-bit key. Similarly to the previous analysis, we are going to ignore the effect of the S-boxes and assume that the encrypting function is the XORing between $n+m$ bits of plaintext and an $(n+m)$-bit key. The processed data is stored in a register array of $n+m$ flip-flops.

The leakage current consumed by a register array of $n+m$ bits can be expressed as:

$$I_{leak} = (n+m)\cdot I_0 + \epsilon \cdot (HW_n + HW_m) \tag{31}$$

We make a distinction between the bits of interest (n) and the bits that introduce algorithmic noise (m). Similarly, we distinguish between HW_n and HW_m.

An attack on n-bits of the secret key is performed similarly to the manner described in previous sections. The attacker inputs all 2^n possible plaintexts and measures the leakage current consumed by the cryptosystem at the time of evaluation. For each possible plaintext, HW_n is calculated for each candidate sub-key, and the correlation between the leakage current measurements and the calculated Hamming Weights is computed.

However, despite performing the analysis without the effect of S-boxes, we attribute to the rest of the bits m the statistical properties that would be expected in a functional encrypting algorithm; namely, that each bit of m is a random variable, independent from the others, with uniform probability of $\frac{1}{2}$ of adopting either of its possible values. As such, HW_m is itself a random variable with expected value and variance, respectively, of $\mu_m = \frac{m}{2}$ and $\sigma_m^2 = \frac{m}{4}$.

Thus, the m bits not directly involved in the attack are a source of noise that decorrelates power consumed with the data being processed.

In fact, the PCC for a cryptosystem with $n+m$ bits, with n key-bits under attack, can be shown to be (without countermeasure, no other sources of noise, and under correct key assumption):

$$\rho_{Ileak,HW_n} = \frac{\epsilon \cdot \sigma_{HW_n}}{\sqrt{\epsilon^2(\sigma_{HW_n}^2 + \sigma_{HW_m}^2)}} = \sqrt{\frac{n}{n+m}} \tag{32}$$

From here, it is straightforward to calculate the PCC between the Hamming Weight of the n bits under attack and the leakage current consumed by a $n+m$ bit cryptosystem in the presence of the countermeasure described in previous sections.

We must simply take into account the changes introduced in the denominator of Equation (14) (the variance of the leakage current consumed by the register array) by the m bits that introduce algorithmic noise. Again, we break down the variance of I_{leak} into three components $Var(I_{leak}) = p1 + p2 + p3$, with:

$$\begin{aligned} p1 &= Var(\epsilon(S) \cdot (HW_n + HW_m)) \\ p1 &= \sigma_\epsilon^2(\sigma_{HW_n}^2 + \sigma_{HW_m}^2) + \mu_\epsilon^2(\sigma_{HW_n}^2 + \sigma_{HW_m}^2) + \cdots \\ &\quad + \sigma_\epsilon^2(\mu_{HW_n}^2 + \mu_{HW_m}^2) + 2\mu_{HW_n}\mu_{HW_m}\sigma_\epsilon^2 \qquad (33) \\ p2 &= Var((n+m) \cdot I_0(S)) \\ p2 &= (n+m)^2 \cdot \sigma_{I_0}^2 \qquad (34) \\ p3 &= 2 \cdot Cov(\epsilon(S) \cdot (HW_n + HW_m), (n+m) \cdot I_0(S)) \\ p3 &= 2 \cdot (n+m) \cdot (\mu_{HW_n} + \mu_{HW_m}) \cdot Cov(\epsilon(S), I_0(S)) \\ &\qquad (35) \end{aligned}$$

A. ALGORITHMIC NOISE: EFFECT OF AVERAGING

The same analysis performed for the countermeasure in the presence of noise averaging can be done under these new conditions. In fact, the same effect is expected for the different variances σ_ϵ^2, $\sigma_{I_0}^2$, and $Cov(\epsilon(S), I_0(S)))$ as in Equations (28), (29), and (30) respectively; namely, their values are reduced by a factor of N, with N being the number of samples.

The same effect can also be expected for the variance of HW_m, which becomes $\sigma_{HW_m}^2/N$. In the particular case with algorithmic noise but no countermeasure applied, it can be shown that the PCC between the leakage current and the Hamming Weight of the bits of interest (Equation (32)) becomes:

$$\rho_{Ileak,HW_n} = \sqrt{\frac{n}{n + \frac{m}{N}}} \qquad (36)$$

Note that in the case of a cryptosystem with $n+m$ bits and an attack on n bits of interest, noise averaging is not obtained by repeating N encryption processes with exactly the same plaintext. Rather, we maintain the plaintext affecting the n bits constant during the N encryption processes, but for each of these encryption processes a random plaintext is chosen for the m remaining bits. With these modifications, numerical simulations of a CPA can be performed as described in section VII.

Figure 15 showcases equation (14) for an 128-bit register array with 8 sub-key bits under attack in the presence of the countermeasure, for different values of N averaging traces per plaintext; that is, that the total number of encryption processes is $N \cdot 256$.

Table 3 presents the results for the case with no countermeasure applied. Equation (36) is evaluated for different values of N (PCC- Theo), along with the PCC obtained through numerical simulations of a CPA (PCC- CPA). The success rate of 100 such CPA's is also noted.

Table 4, on the other hand, presents the results for the case with countermeasure, with a fixed body bias' DR of 1 V, and $s_{max} = 25$. Equation (14) with algorithmic noise is evaluated for different values of N (PCC- Theo), along with the PCC obtained through numerical simulations of a CPA (PCC- CPA).

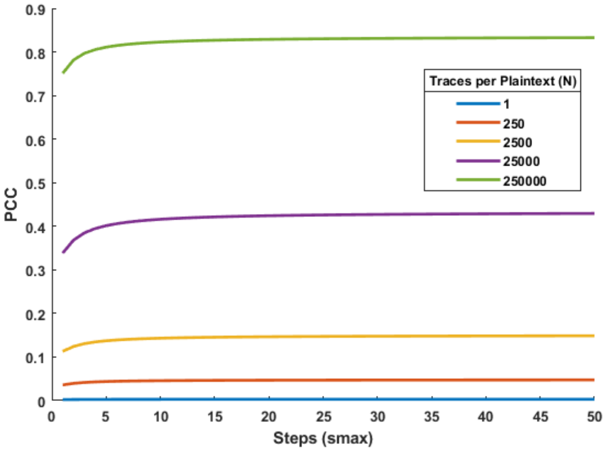

FIGURE 15. Effect of averaging on the PCC between the leakage current and the hamming weight of a 128-bit register array, with 8 bits under attack, in the presence of the proposed countermeasure with a fixed, maximum body bias DR of 1 V and different number *N* of traces per plaintext.

TABLE 3. Theoretical and numerical simulation results of the PCC, along the success ratio for 100 simulated attacks, for a 128 bit register array with 8 bits under attack without countermeasure for different number *N* of averaged traces per plaintext.

No Countermeasure			
	PCC		
N	Theo	CPA	Success Rate
1	0.25	0.17	0.61
300	0.976	0.977	1

TABLE 4. Theoretical and numerical simulation results of the PCC, along the success ratio for 100 simulated attacks, for a 128 bit register array with 8 bits under attack with countermeasure for different number *N* of averaged traces per plaintext.

Countermeasure			
	PCC		
N	Theo	CPA	Success Rate
1	0.0030	0.0091	0
250	0.0470	0.0290	0.05
2,500	0.1472	0.1778	0.14
25,000	0.4328	0.4294	0.98
250,000	0.8301	0.8357	1

Note that in both Tables 3 and 4, the column PCC-CPA is the results of a single CPA attack under each particular value of N. The different values depicted in the columns PCC-CPA are, thus, subjected to noise and can vary between experiments, under the same conditions. However, it can be seen that, as N increases, the values get closer to those obtained from the evaluation of Equations (36) and (14).

On the other hand, both Fig. 15 and Table 4 showcase the need to increase the number of samples by a factor of 250 to obtain the same results as in the case of a 8-bit cryptosystem without algorithmic noise (Fig. 10 and Table 1).

IX. CONCLUSION

In this paper, an analysis of a novel countermeasure to leakage-based Power Analysis Attacks is presented for cryptosystems implemented in FDSOI technology. The countermeasure takes advantage of the large body biasing range of FDSOI technology to modify the static power consumption during the encryption process.

The paper quantifies the impact of the countermeasure on the correlation between the leakage current consumed by the cryptosystem and the data being processed. A theoretical analysis of the electrical behaviours of registers implemented in FDSOI technology in the presence of the countermeasure allows the derivation of a correlation model in terms of the countermeasure parameters.

The paper shows that the random application of body bias can significantly reduce the correlation between power consumed and processed data. This reduction is proportional to the dynamic range of body bias, and the correlation reduction is dependent on both the number of steps and the magnitude of the body bias step.

Numerical simulations based on technological parameters are also used to study the effect of averaging the power measurements on the effectiveness of the countermeasure. These simulations show that in order to obtain the secret key with a high degree of confidence it is necessary to severely increase the number of encryptions: in noiseless conditions, a 100-fold increase in measurements is required to correctly identify the secret key with high certainty. This number is further increased by a factor of 250 in the particular case of a simulated cryptosystem of 128 bits.

While these results are difficult to directly compare with those obtained in [17], given the difference in technology, the trends observed are shared. In both [17] and this article, it can be seen that the wider the Dynamic Range of Body bias applied, the more resilient the system becomes to attacks.

At the same time, in [17], the authors report that the highest increase in needed traces to obtain the secret keys is observed when the values that the body bias can adopt are those closest to the maximum allowable voltage by the technology. In their case, with a DR of 1.2V, between 0.8 V and 2 V. This is consistent with the behaviour observed in fig. 5, where the sensitivity of the different leakage currents to the body bias is highest near the maximum nominal body bias value.

On the other hand, this article limits its study to systems implemented with Low Threshold Voltage (LVT) transistors under Forward Body Bias (FBB) regime. Both these considerations increase the amount of leakage current consumed by the system, as compared to a similar circuit implemented with transistors with higher Threshold Voltage. This, in turn, makes the apparent leakage of information higher. However, the leakage currents of High VT transistors might be less sensitive to variations of their body bias, reducing the capacity of the countermeasure to introduce noise. Thus, there might be a trade-off between the information leaked by a system and the effectiveness of the countermeasure to introduce decorrelated noise depending on technological parameters.

At the same time, there is an inverse relation between the Threshold Voltage and the maximum clock frequency at which a circuit can operate. Since longer clock periods allow the acquisition of more intra-trace samples that can be averaged, higher VT transistors might facilitate the acquisition of samples with lower noise, at the expense of reduced signal amplitude, further pointing to a potential trade-off.

These considerations warrant further exploration in future studies.

While results are promising, further study in more realistic conditions is also required, including the use of more complex cryptosystems to account for additional sources of power consumption, with models that can introduce non-algorithmic noise, and experimentation on real system implementations.

REFERENCES

[1] Y.-K. Choi, K. Asano, N. Lindert, V. Subramanian, T.-J. King, J. Bokor, and C. Hu, ''Ultra-thin body SOI MOSFET for deep-sub-tenth micron era,'' *IEEE Electron Device Lett.*, vol. 21, no. 5, pp. 254–255, May 2000.

[2] S. Clerc, T. Di Gilio, and A. Cathelin, *The 4th Terminal Integrated Circuits and Systems*, Cham, Switzerland: Springer, 2020.

[3] S. M. D. Pozo, F.-X. Standaert, D. Kamel, and A. Moradi, ''Side-channel attacks from static power: When should we care?'' in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2015, pp. 145–150.

[4] T. Moos, A. Moradi, and B. Richter, ''Static power side-channel analysis—An investigation of measurement factors,'' *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 2, pp. 376–389, Feb. 2020.

[5] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, ''Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits,'' *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 2, pp. 355–367, Feb. 2010.

[6] E. Brier, C. Clavier, and F. Olivier, ''Correlation power analysis with a leakage model,'' in *Cryptographic Hardware and Embedded Systems—CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Germany: Springer, 2004, pp. 16–29.

[7] N.-H. Zhu, Y.-J. Zhou, and H.-M. Liu, ''Employing symmetric dual-rail logic to thwart LPA attack,'' *IEEE Embedded Syst. Lett.*, vol. 5, no. 4, pp. 61–64, Dec. 2013.

[8] N. Zhu, Y. Zhou, and H. Liu, ''Counteracting leakage power analysis attack using random ring oscillators,'' in *Proc. SNS PCS*, May 2013, pp. 74–77.

[9] W. Yu and Y. Wen, ''Leakage power analysis (LPA) attack in breakdown mode and countermeasure,'' in *Proc. 31st IEEE Int. Syst. Chip Conf. (SOCC)*, Sep. 2018, pp. 102–105.

[10] W. Yu and S. Köse, ''False key-controlled aggressive voltage scaling: A countermeasure against LPA attacks,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 12, pp. 2149–2153, Mar. 2017.

[11] W. Yu and S. Köse, ''Security-adaptive voltage conversion as a lightweight countermeasure against LPA attacks,'' *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 7, pp. 2183–2187, Jul. 2017.

[12] A. Gornik, A. Moradi, J. Oehm, and C. Paar, ''A hardware-based countermeasure to reduce side-channel leakage: Design, implementation, and evaluation,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 8, pp. 1308–1319, Apr. 2015.

[13] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, ''Towards sound approaches to counteract power-analysis attacks,'' in *Advances in Cryptology*, M. Wiener, ed. Berlin, Germany: Springer, 1999, pp. 398–412.

[14] T. Moos, A. Moradi, and B. Richter, ''Static power side-channel analysis of a threshold implementation prototype chip,'' in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 1324–1329.

[15] D. Bellizia, D. Cellucci, V. Di Stefano, G. Scotti, and A. Trifiletti, ''Novel measurements setup for attacks exploiting static power using DC pico-ammeter,'' in *Proc. Eur. Conf. Circuit Theory Design (ECCTD)*, Sep. 2017, pp. 1–4.

[16] W. Yu and S. Köse, ‘‘Security implications of simultaneous dynamic and leakage power analysis attacks on nanoscale cryptographic circuits,’’ *Electron. Lett.*, vol. 52, no. 6, pp. 466–468, Mar. 2016.

[17] B.-A. Dao, T.-T. Hoang, A.-T. Le, A. Tsukamoto, K. Suzaki, and C.-K. Pham, ‘‘Exploiting the back-gate biasing technique as a countermeasure against power analysis attacks,’’ *IEEE Access*, vol. 9, pp. 24768–24786, 2021.

[18] M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, and A. Trifiletti, ‘‘Effectiveness of leakage power analysis attacks on DPA-resistant logic styles under process variations,’’ *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 429–442, Feb. 2014.

[19] M. Alioto, S. Bongiovanni, G. Scotti, and A. Trifiletti, ‘‘Leakage power analysis attacks against a bit slice implementation of the serpent block cipher,’’ in *Proc. 21st Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, Jun. 2014, pp. 241–246.

KENNETH PALMA graduated in medicine from the University of Barcelona (UB), in 2013. He received the B.Sc. degree in electronic engineering and the M.Sc. degree in electronic engineering with a focus on microelectronics from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in electronic engineering.

FRANCESC MOLL (Senior Member, IEEE) received the M.Sc. degree in physics from the University of the Balearic Islands, Spain, in 1991, and the Ph.D. degree in electronic engineering from the Universitat Politècnica de Catalunya (UPC), in 1995. He has been a Professor with the Department of Electronic Engineering, UPC, since 1997. His research interests include reliability and robustness issues relevant to integrated circuit design, especially in advanced technology nodes, such as signal integrity modeling and its impact, manufacturing variability, and ultra low power and voltage circuits.