Evaluation of Power Constant Dual-Rail Logics Countermeasures against DPA with Design Time Security Metrics

Sylvain Guilley, Laurent Sauvage, Florent Flament, Vinh-Nga Vong, Philippe Hoogvorst, Renaud Pacalet

Evaluation of Power-Constant Dual-Rail Logics Counter-Measures against DPA with Design-Time Security Metrics

Sylvain Guilley, *Member, IEEE,* Laurent Sauvage, Florent Flament, Vinh-Nga Vong, Philippe Hoogvorst, and Renaud Pacalet
E-mail: < sylvain.guilley@TELECOM-ParisTech.fr >

Abstract—Cryptographic circuits are nowadays subject to attacks that no longer focus on the algorithm but rather on its physical implementation. Attacks exploiting information leaked by the hardware implementation are called side-channel attacks (SCA). Amongst those attacks, the differential power analysis (DPA) established by Paul Kocher *et al.* in 1998 represents a serious threat for CMOS VLSI implementations. Different countermeasures that aim at reducing the information leaked by the power consumption have been published. Some of these countermeasures use sophisticated backend-level constraints to increase their strength.
As suggested by some preliminary works (*e.g.* by Huiyun Li from Cambridge University), the prediction of the actual security level of such countermeasures remains an open research area. This article tackles this issue on the example of the AES SubBytes primitive. Thirteen implementations of SubBytes, in unprotected, WDDL and SecLib logic styles with various backend-level arrangements are studied. Based on simulation and experimental results, we observe that static evaluations on extracted netlists are not relevant to classify variants of a counter-measure. Instead, we conclude that the fine-grain timing behavior is the main reason for security weaknesses. In this respect, we prove that SecLib, immune to early-evaluation problems, is much more resistant against DPA than WDDL.

1 Introduction

Side-channel attacks are techniques to extract keys or secret elements from cryptosystems otherwise unbreakable by cryptanalysis or brute force. The instant power dissipation of a device has been studied first because it corresponds to a practical scenario, especially for smartcards. Indeed, those embedded devices receive their power from the outside. A rogue reader can thus supply the card while recording the instant current drawn, typically with a fast acquisition card. Based on these measurements, so-called differential (DPA [1]) or correlation power analyses (CPA [2]), referred to as in the sequel by the same generic term "DPA", can be mounted. DPA exploits the coincidence of two properties that characterize every cryptographic algorithm. On the one hand, it is always possible to exhibit an internal variable dependent on a manageable subset (*i.e.* small, usually 6 or 8 bits) of the key and of the input or output data. On the other hand, in "high threshold voltage" technologies, CMOS gates consume only when toggling. Therefore the power consumption is directly proportional to the circuit's activity. The power consumption due to the internal variable activity can be extracted from the circuit power traces by correlation with a power model. The attacker makes guesses about the unknown key subset and for each of them computes the correlation function. The larger correlation will betray the correct key hypothesis. Any unprotected cryptographic implementation is thus vulnerable to DPA, because any use of the key bits leads to an information leakage in the power dissipation.

One way to protect a device from the DPA is to make its power consumption independent from the input data and key, by making it constant. This is the aim of the dual-rail with precharge logic (DPL [3]). This logic ensures by design a constant toggling rate irrespective of the data manipulated.

In dual-rail, every Boolean variable a is represented by a couple of two wires (a_0, a_1); when a is valid, $a = 0 \Leftrightarrow (a_0, a_1) = (1, 0)$ and $a = 1 \Leftrightarrow (a_0, a_1) = (0, 1)$. The convention to signal that A is not valid is $a_0 = a_1$. Every computation consists in one precharge (where a is invalid) followed by one evaluation (where a is valid). a_0 and a_1 are complementary. Whatever the input and key, exactly one and only one of (a_0, a_1) will toggle. The number of toggles is thus constant, so should be the overall power consumption.

Wave dynamic differential logic (WDDL [4]) and SecLib (Secured Library [5]) are two DPL solutions. WDDL is a DPL logic that makes use of the standard cell library.

- *S. Guilley, L. Sauvage, F. Flament, Ph. Hoogvorst and R. Pacalet are with the Institut TELECOM, TELECOM ParisTech, CNRS LTCI (UMR 5141), Département COMELEC, 46 rue Barrault 75 634 PARIS Cedex 13, FRANCE.*
- *S. Guilley and L. Sauvage are also with the TELECOM ParisTech spin-off Secure-IC.*
- *V.-N. Vong is embedded software engineer at Airbus, and has carried out this work while at TELECOM ParisTech.*
- *R. Pacalet is head of the System-on-Chip laboratory of TELECOM ParisTech located at CICA, BP 193, 2229 route des Crêtes 06 904 Sophia-Antipolis Cedex, FRANCE.*

SecLib is another DPL logic that relies on customized balanced cells set that furthermore synchronize their inputs before evaluating.

In addition to comparing insecure logics with WDDL and SecLib, a second goal of this paper is to study further refinements of DPL logics consisting in balancing the layout. WDDL instances are separable: each gate is made up of two independent halves. Therefore, the two dual instances can be designed to have the same structure. They can also be constrained to be placed side-by-side. All DPL logics can be forced to have a balanced interconnection between them, and, on top of that, the wiring can be shielded. In MDPL [6], it has been suggested an alternative method to balance a netlist with a *deus ex machina* mechanism, consisting in randomly swapping the signification of a_0 and a_1 according to one single-bit mask. However, this protection can be defeated easily by a so-called PDF-attack [7]. This attack becomes all the more difficult as the netlist is already well balanced without any masking. This is the main focus for our work.

In order to compare logics styles and backend countermeasures, a chip called `SubBytes` has been realized. It embeds thirteen versions of the "SubBytes" function, the substitution box used in the AES algorithm [8]. Its purpose is to enable a comparative evaluation of the several implementations of the same combinatorial block.

The rest of the article is organized as follows. Section 2 presents the security features that are implemented in the `SubBytes` circuit. The section 3 gives conclusions about the expected security level using a static evaluation based on the layout study. Next, section 4 is dedicated to the dynamic evaluation based on actual experiments. In this section, the specifications of the ASIC floorplan, programming model and drivers are described and motivated; then, an experimental evaluation of each SubBytes module is carried out. The section 5 is a discussion about the relevance of design-time security metrics, the efficiency of WDDL *versus* SecLib, and the usefulness of backend-level counter-measures. Finally, section 6 draws the conclusions of the paper and opens further research perspectives.

2 Presentation of the Security Features Embedded into the SubBytes Chip

2.1 Thirteen versions of the AES SubBytes Combinatorial Function

For the realization of the `SubBytes` chip, four libraries of cells were assessed:

1) Standard cell (CORE9GPLL library from STMicroelectronics, version 4.1),
2) Read-Only Memory ("ROM", generated by the STMicroelectronics Unicad tool – ugnLib),
3) WDDL [4] and enhanced-WDDL, based upon CORE9GPLL,

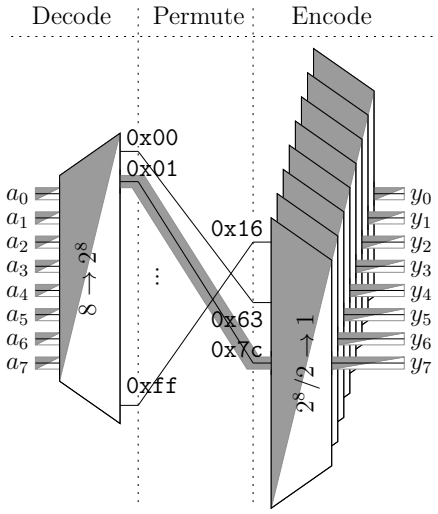


Fig. 1. The decode/permute/encode low-power and unprotected architecture for SubBytes.

4) SecLib [9], a custom secure quasi-delay independent (QDI) logic. SecLib is similar to the quasi-delay insensitive logic presented in [10]: its structure is however optimized and it does not support the errors reporting capability.

The two first libraries (standard cells and ROM) are unprotected, and can thus constitute references for the security evaluation. The `SubBytes` chip embeds four unprotected instances with the following architectures:

1) **Standard cell**, described in VHDL as look-up table [8, p. 16] (called `stdcell_lut`),
2) **Standard cell**, factored in GF$(16)^2$, as suggested by Vincent Rijmen [11]–[13] (called `stdcell_gf`),
3) **Standard cell**, in a decode/permute/encode architecture presented by Guido Bertoni [14], depicted in Fig. 1 (called `stdcell_gb`)[1],
4) **Layout-level generated** low-power contact-programmable ROM.

The references [16] and [17] are comprehensive studies of the different hardware architectures of the AES SBox and will help the reader in having a complete overview of the SBox in any dimension: security, area and power dissipation.

The secured implementations, WDDL and SecLib, embedded in `SubBytes` both resort to DPL. The SecLib cells are part of a full-custom library [5]. The WDDL and SecLib gates we consider in the sequel contain only one- and two-input gates.

WDDL, illustrated on the example of an `AND` gate in Fig. 2, suffers from two identified weaknesses:

1) The two dual standard cells making up the WDDL gate are structurally different. They thus consume

1. The work of Guido Bertoni *et al.* has been extended by Matteo Giaconia *et al.* in [15]. It yields an even more balanced design and thus achieves a still higher DPA resistance. However, at the date of the tape-out of the `SubBytes` circuit, we were not aware of this implementation, which explains it is not included into the ASIC.

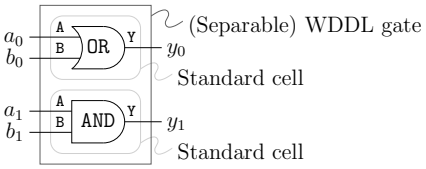


Fig. 2. The WDDL "AND" functionality.

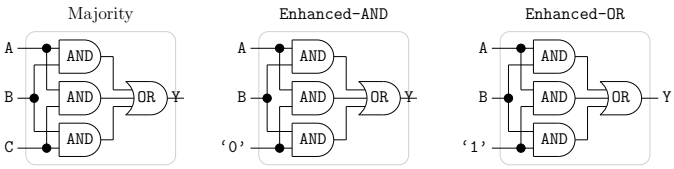


Fig. 3. The majority standard cell, called AO5NLL in STMicroelectronics library, can be specialized as two enhanced-WDDL cells implementing the *true* "AND" and "OR" functionalities.

slightly different amounts of power with different current signature.

2) Depending on the arrival order of its inputs, the gate activity occurs at different instants [18].

The first issue can be fixed, employing what we call "enhanced-WDDL" (or eWDDL for short) cells, based on the 3-input majority standard cell from the STMicroelectronics library: $(\mathtt{A},\mathtt{B},\mathtt{C}) \mapsto \mathtt{A}\cdot\mathtt{B}+\mathtt{B}\cdot\mathtt{C}+\mathtt{C}\cdot\mathtt{A}$. The schematic of the majority and of the two enhanced WDDL derived cells are given in Fig. 3. Those cells use the same architecture as MDPL [6], albeit with a constant hardwired mask.

The second issue of WDDL cannot be solved at the implementation-level, because it is fundamentally logical.

The figure 4 shows that both issues are definitely fixed in SecLib:

1) All evaluations activate the same number of indiscernible logic gates: one C-element [19] and two OR gates. This contrasts with WDDL with which either a AND or a OR gate is activated.
2) The head C-elements synchronize the signals, thus preventing the gate from evaluating early.

The logic underlined in gray in Fig. 4 is activated in the transition from precharge to evaluation and vice-versa.

The implementation-level variations amongst the secured cells are many-fold. They are defined and described in the list below:

- $\mathcal{B}1$: identity of the dual gates,
- $\mathcal{B}2$: differential placement,
- $\mathcal{B}3$: differential routing,
- $\mathcal{B}4$: differential dummies,
- $\mathcal{B}5$: shield by global wires of each dual pair of wires,
- $\mathcal{B}6$: complete module area shield by a top-level metal coating plane.

The first backend feature $\mathcal{B}1$ makes the computational paths to the true and the false outputs indistinguishable.

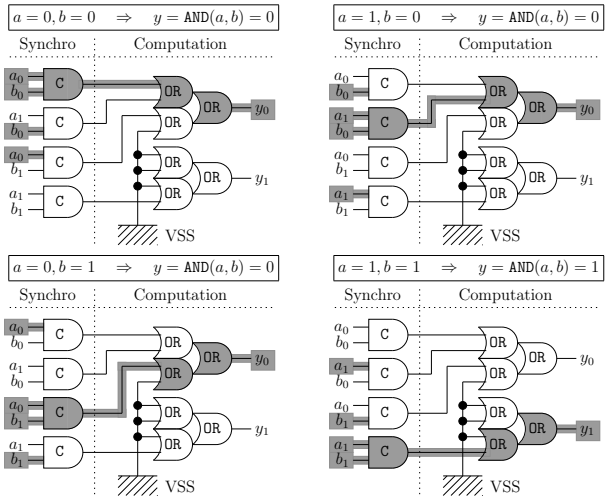


Fig. 4. The SecLib AND function activates always the same number of gates and is guaranteed to be immune from early evaluation thanks to the synchronizing C-elements.

The second item $\mathcal{B}2$ requires the gate to be somehow separable into two halves (refer to [20, appendix A]). Differential placement lessens the risks of unbalancedness due to variations from one location to another across the circuit's die. This helps reduce the disparities in power consumption, but most importantly, the constraint placement tends to make the routing (automatic, *i.e.* not constrained) similar for each gate. This pseudo-differential routing has the beneficial consequence that the load of each gate dual outputs is sensibly the same, so does the power consumption. The third item $\mathcal{B}3$ depends on the second one: differential routing can only be achieved provided the placement is also differential. It ensures a perfectly parallel hence balanced routing. It is performed thanks to the "backend-duplication" method [20]. The fourth item $\mathcal{B}4$ depends in turn on the third one: dummy metal slots can be spread in a differential way only if the routing is differential too. We recall that dummy slots are non-functional pieces of metal scattered "randomly" for the layer to reach a minimal density. This is indeed specified by the design rules manual in order to guarantee the planarity after chemical-mechanical polishing during fabrication. A constant planarity guarantees that the differential wires keep the same thickness along their route. The fifth item $\mathcal{B}5$ protects every dual pair of wires from cross-talk by placing a global (vss or vdd) wire between them. Usually, only the ground vss is used for shielding because it collects and drains all the parasitic currents without injecting the noise of the supply vdd into the integrated circuit's core. However, combining vss and vdd is an option worth considering when the power is not noisy because the shield wires can serve as a pervasive supplying network. They thus keep the voltage drops of the underneath logic cells as low as possible [21]. The sixth item $\mathcal{B}6$ is independent from the

others and consists in coating the SubBytes module with a top-level metal (M6) plane. This shield against electromagnetic analyses (EMA [22]) is not studied in this paper, because we focus exclusively on power analysis.

Thirteen SubBytes modules are designed, combining the various logic styles and implementation-level options. They are detailed in Tab. 1. In the sequel they are either referred to by their number or by their nickname, given in first and second columns respectively.

The unprotected implementations (modules (1) to (4)) do not benefit from any differential feature ($\mathcal{B}1$ to $\mathcal{B}5$), hence the "not applicable" (abbreviated "n/a") indications in the table.

Not surprisingly, secured implementations (modules (5) to (13)) suffer from a large overhead in terms of silicon area. Thus, the most compact architecture amongst the unprotected (namely (1), *aka* `stdcell_gf`) is selected as a reference. The performances of the thirteen modules are given in Tab. 2. This table clearly shows that the synthesizer tries hard to use as many cells as possible from the library for the straightforward LuT architecture (53 unique instances as for (2)), to the detriment of a global optimization (such as the smaller implementation (1), that uses only 22 unique instances).

2.2 Projected Security Level of DPL Versions of SubBytes

The security level of WDDL and SecLib (with the same security features as `wddl_4` and `seclib_4`) has already been studied in simulation in [21], and from experimental measurements done *in silico* in [23]. In contrast, this article explores a trade-off between security features and cost overhead. Thus, we investigate degraded (*i.e.* sub-optimal) backend-level countermeasures with respect to WDDL and SecLib.

The expected security partial order is expressed by the "$\prec$" operator in Fig. 5.

Note 1 Unprotected implementations have (*a priori*) a comparable level of security (no counter-measure.) Secured libraries, based upon either WDDL or SecLib, are expected to be more secure. Differential placement, differential routing and differential dummies are counter-measures that are built on top one of each other to increase the security provided by the cell library. EMA [22] shield is not expected to impact the protection against the DPA [1].

Note 2 Everything being otherwise comparable (differential placement, routing and metal dummies), WDDL is expected to be weaker than SecLib [21], [23]. The reason is that at the "silicon"-level, SecLib is more balanced than WDDL. Further "metal"-level (*i.e.* interconnect) security features will enhance the security, but will most probably not make up for the "silicon"-level (*i.e.* logic) discrepancies.

2.3 Evaluation Methodology for the Simulations & the Experimental Measurements

There are two ways to evaluate the security level of the competing SubBytes modules.

1) *Static* evaluation considers the layout and tries to find dissymmetries in it. Statistics on the nets can be collected from the netlist. The dispersion of the characteristics across nets is considered a measurement of static unbalancedness. This evaluation strategy is called "design-time" because it does not require to have a silicon prototype at disposal. This is the approach carried on in Sec. 3.
2) *Dynamic* evaluation considers the global behavior of each SubBytes module. Statistics are realized by trying all possible input configurations. The approach is thus either a simulation or real-world measurements on a silicon chip. The latter requires a device, and thus costs more because the whole fabrication process must be realized. However, it is also a more accurate than simulation because it places the evaluator in the same shoes as a potential attacker. Section 4 concentrates on this aspect of the evaluation.

2.4 Motivation for Combinatorial Gates Study

Most side-channel attacks are based on correlations with an intermediate variable. In both software and hardware implementations, a variable is stored in a register. From an attacker's standpoint, this is a great opportunity since the register activity is reproducible in time. This means that statistics will coherently correlate with either the register contents or its contents change.

It thus appears that combinatorial logic is seldom studied as an exploitable source of leakage. The main reason is probably that the relationship between the activity of this logic and the data it computes is far from being obvious: combinatorial gates evaluate at data-dependent dates and might even produce non-functional transitions (called glitches) whose impact on the power dissipation is difficult to model. Moreover, while the number and exact location of the registers are quite simple to guess or reverse-engineer, the structure of combinatorial logic is much more difficult to figure out. So, paradoxically, although in some algorithms such as AES the combinatorial logic makes up about 80 % of the implementation area and power dissipation, it happens not to be the most frequent target for a side-channel attack.

One example where the analysis of a combinatorial net has been successful was the attack of a masked sbox, by Stefan Mangard *et al.* at CHES'05 [24]. Amongst the whole netlist, they identified the net that was the less dependent in the mask, and focused the attack on the variable it carried. However, apart from this very special situation, inner nets within combinatorial logic are not the most frequently encountered candidates for a side-channel attack.

TABLE 1
Security features $\mathcal{B}1$ to $\mathcal{B}6$ of the thirteen SubBytes modules.

#	Nickname	$\mathcal{B}1$ (Gate)	$\mathcal{B}2$ (Placement)	$\mathcal{B}3$ (Routing)	$\mathcal{B}4$ (Dummy)	$\mathcal{B}5$ (Shield)	$\mathcal{B}6$ (EMA)
(1)	stdcell_gf	n/a	n/a	n/a	n/a	n/a	no
(2)	stdcell_lut	n/a	n/a	n/a	n/a	n/a	no
(3)	stdcell_gb	n/a	n/a	n/a	n/a	n/a	no
(4)	rom	n/a	n/a	n/a	n/a	n/a	no
(5)	wddl_0	no	no	no	no	no	no
(6)	wddl_1	no	yes	no	no	no	no
(7)	wddl_2	no	yes	yes	yes	no	no
(8)	wddl_4	no	yes	yes	yes	yes	no
(9)	ewddl_4	yes	yes	yes	yes	yes	no
(10)	seclib_1	yes	yes	no	no	no	no
(11)	seclib_2	yes	yes	yes	yes	no	no
(12)	seclib_4	yes	yes	yes	yes	yes	no
(13)	seclib_4ema	yes	yes	yes	yes	yes	yes

$$\left[\begin{array}{ll} (1) = (2) = (3) = (4) \prec \left\{\begin{array}{l}(5) \prec (6) \prec (7) \prec (8) \prec (9)\\ (10) \prec (11) \prec (12) = (13)\end{array}\right. & \text{// See note 1.}\\ (6) \prec (10) & \text{// See note 2.}\\ (7) \prec (11) & \text{// See note 2.}\\ (8) \prec (9) \prec (12) & \text{// See note 2.}\end{array}\right.$$

Fig. 5. Expected security order of the 13 modules embedded into the ASIC `SubBytes`.

TABLE 2
SubBytes blocks physical characteristics.

#	Area [μm^2]	#! instances	# instances	Density
(1)	1 767	22	144	98.6 %
(2)	4 018	53	423	98.1 %
(3)	4 841	53	548	98.6 %
(4)	12 830	n/a	n/a	n/a
(5)	8 981	2	342 × 2	95.8 %
(6)	10 760	3	449 × 2	93.7 %
(7)	10 844	3	449 × 2	93.0 %
(8)	16 097	3	449 × 2	62.5 %
(9)	16 944	3	451 × 2	75.9 %
(10)	23 468	8	166	88.2 %
(11)	25 586	8	166	80.9 %
(12)	25 417	8	166	81.4 %
(13)	25 417	8	166	81.4 %

But in a circuit where the registers are perfectly protected, the only remaining sources of data dependency are the Boolean logic gates. This is the assumption we made in this article and the reason why we focused on combinatorial parts evaluation.

Figure. 6 illustrates this. It shows the voltage drop over a spy resistor monitoring the instant current consumed by an unprotected DES module during two clock cycles. The registers consume current at the clock rising and falling edges of the clock, whereas the combinatorial logic consumes current only after the registers have evaluated, typically a couple of nanoseconds after the rising edge of the clock. It seems easy to balance the registers, because there are not so many of them in an implementation. However, the combinatorial parts are numerous and complex. Both DPA and template attacks could target the variations in the combinatorial parts even if the registers are exactly balanced.

The sbox, for instance, offers room for concrete attack thanks to its mathematical properties. If we denote by S the functionality of the sbox, then a correlation attack basically consists in evaluating an auto-correlation of S (between the measurements and the guessed model). When the correct key is guessed, the auto-correlation is maximal, equal to $(S \otimes S)(0)$. Otherwise, the correlation yields $(S \otimes S)(\epsilon) \leq (S \otimes S)(0)$, where ϵ is the error on the key guess. In case the exclusive-or operation is used to mix the key with the datapath, $\epsilon \doteq k_{\text{actual}} \oplus k_{\text{guessed}}$. The contrast of an auto-correlation is all the higher as the sbox S is non-linear [25]–[28]. For cryptanalytic reasons, the sboxes are chosen as highly non-linear. In this respect, the abstract function of S, rather than its implementation, helps the attacker in her decision for the correct key: mathematical properties of S allow to discriminate efficiently the different key candidates.

3 Static Evaluation of the Security of Nine SubBytes Dual-Rail Modules

The security of the dual-rail modules is assessed statically based on the study of differential routing unbalancedness. In order to reach this goal, the resistance "R" and capacitance "C" for every net of the dual-rail modules have been extracted after completion of all backend steps (*i.e.* placement, routing, dummies insertion). The extraction tool is `rcOut`, provided with Cadence software suite SoC/Encounter version 6.1. Without any surprise, we observe that all the resistances match pair-wise, because this quantity depends only on the geometry of the nets. In contrast, the capacitances are different from one regular net to its dual, since capacitances are cross-coupled with the neighboring nets, and

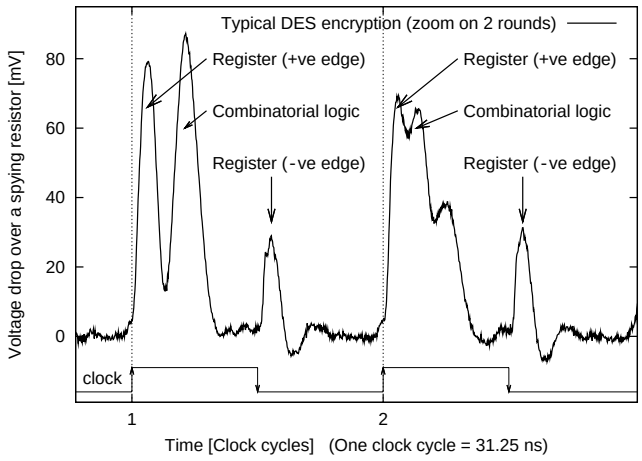


Fig. 6. Typical voltage trace of an unprotected DES module. This quantity is proportional to the instant current consumed by the chip. The sequential (positive & negative clock edges) and combinatorial currents are identified by arrows.

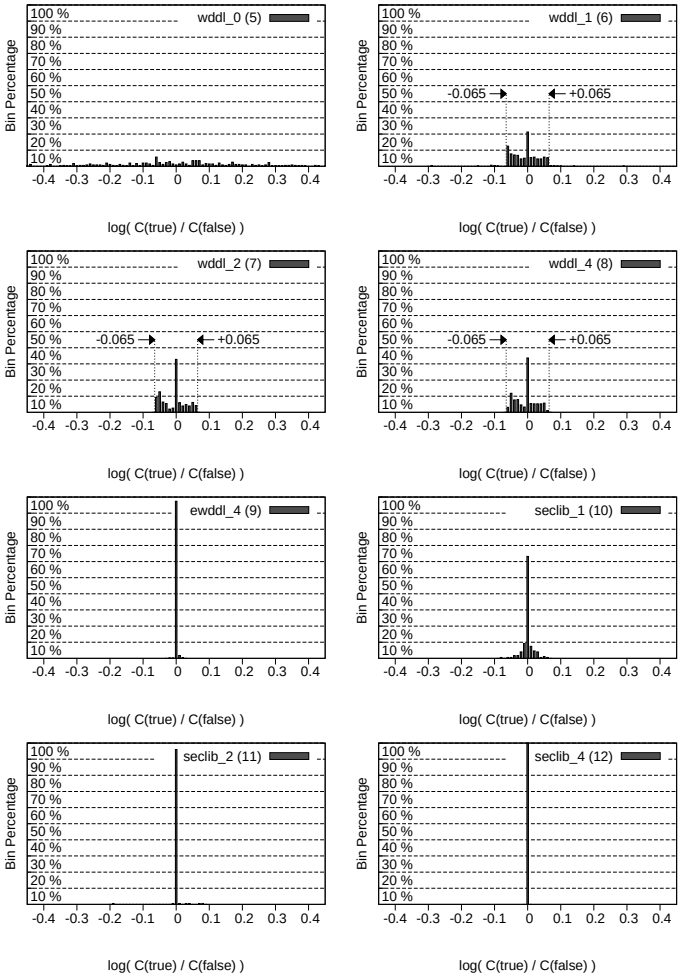


Fig. 7. Distribution of the extracted deviation from the perfectly balanced dual-rail pair (*default extraction mode*.)

the neighborhood of each dual net differs. For each net, the relationship between the parameter "C" extracted from the layout and the instant current drawn by the driver when it switches is linear: $I = VDD \times C$, where VDD is the nominal power supply voltage measured relatively to the ground. Therefore, the ratios between true and false nets capacitances, denoted C_1 and C_0, are computed. Any deviation from 1 is a dissymmetry. Indeed, the observable side-channel amplitude is $|I_1 - I_0| = VDD \times |C_1 - C_0|$, which is non-zero if and only if (iff) $C_1/C_0 \neq 1$. The logarithm of these quantities is plotted in Fig. 7 and 8 to allow for a duality-wise agnosticism:

- if the load of a true net is ε more than its false counterpart, then $\log(\frac{1+\varepsilon}{1}) \approx \overline{\boxed{+\varepsilon}} + \mathcal{O}(\varepsilon)$, whereas
- if the unbalancedness is the opposite, $\log(\frac{1}{1+\varepsilon}) \approx \overline{\boxed{-\varepsilon}} + \mathcal{O}(\varepsilon)$, which is "fair" w.r.t. the true/false duality: the penalty is exactly the opposite at first order, hence the same in absolute value.

Fig. 7 shows dispersion in the so-called "default mode", where capacitances are extracted only w.r.t. the ground (`vss` = 0 volt). In Fig. 8, cross-capacitances between nets are extracted too, in a π-model, also called "detailed mode". Thanks to the usage of the logarithmic scale, the dispersion profiles are centered around 0. One can notice that they are more or less scattered. The dispersion is ideal in the "default mode". The values for module (12), for instance, present the shape of a Dirac peak with the chosen quantification of 1 %. The "detailed mode" better captures the unbalancedness due to the neighborhood dissymmetry: the same module (12) does show an appreciable dispersion in C_1/C_0. The module (13) is not represented because its coupling with the ground is strictly equal to that of (12).

In order to easily compare these dispersions, we compute the standard deviation, also abbreviated "std_dev" in the sequel. Those figures are given in Tab. 3. The module `wddl_0`, that is neither placed nor routed differentially, is — by far — the worst. For the other modules we need to notice that the nets capacitance is made up of two components:

1) The *wire* capacitance C_{wire}. The differential routing, the dummies and the shield are supposed to reduce the dispersion in the wire capacitance.
2) The *gate* input capacitance C_{gate}. The logic style is expected to impact this part of the capacitance dissymmetry: WDDL is not balanced in the gates inputs, because the dual gates are different, whereas eWDDL and SecLib logic are.

The average ratio between the wire capacitance and the total capacitance $C_{total} \doteq C_{wire} + C_{gate}$ is about 50 %, which means that dissymmetries in *wires* and *gate* inputs are to be fought with the same amount of efforts. Behind `wddl_0`, the WDDL modules `wddl_{1,2,4}` come next, due to the unbalancedness of the gates input capaci-

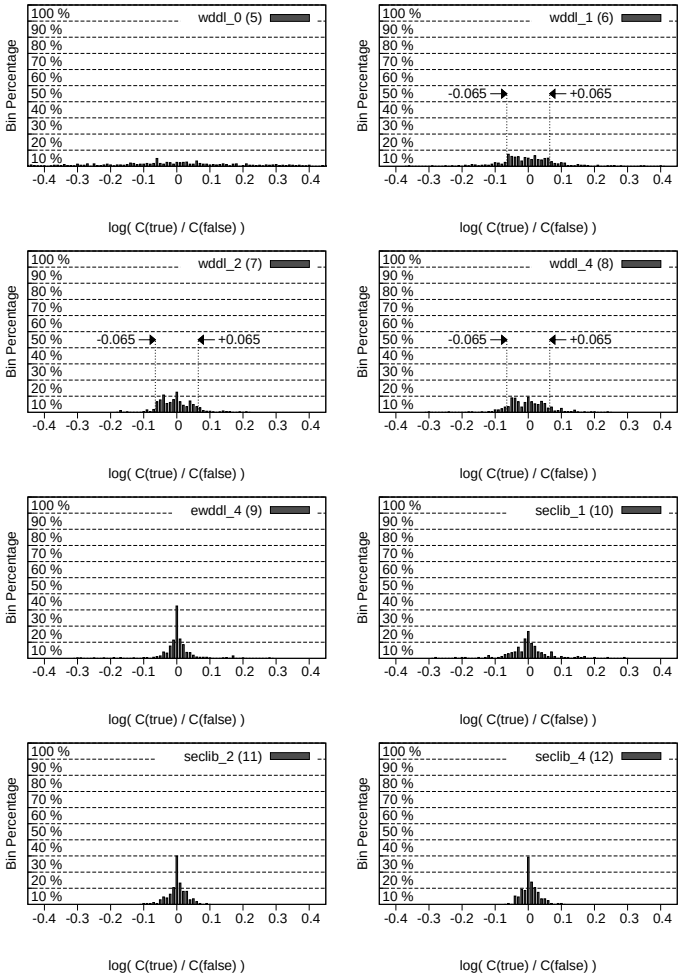


Fig. 8. Distribution of the extracted deviation from the perfectly balanced dual-rail pair (*detailed extraction mode.*)

tances. The dispersion of C_{gate} (of inputs `A` and `B`) in WDDL can be observed in histograms (6), (7) & (8) of Fig. 7. Apart from the inverter cells, their netlists are made up exclusively of `AND` and `OR` standard cells, that happen to have different input capacitances:

$$\log\left(\frac{C_{AND:A}}{C_{OR:A}}\right) = \log\left(\frac{1.63\text{ pF}}{1.53\text{ pF}}\right) = 0.063 \quad \approx$$
$$\log\left(\frac{C_{AND:B}}{C_{OR:B}}\right) = \log\left(\frac{1.43\text{ pF}}{1.34\text{ pF}}\right) = 0.065\,.$$

The other modules (eWDDL and SecLib) have the input gates balanced, and thus feature a smaller dispersion, because only the routing dissymmetry remains. For both WDDL and SecLib, it is clear that the back-end duplication does help (`wddl_1` *vs* `wddl_2` and `seclib_1` *vs* `seclib_2`). Notice that in Fig. 7, SubBytes module `seclib_1` (10) seams at first glance to be more dispersive than module `seclib_2` (11). However, some rare net couples, with $|\log(C(true)/C(false))| \approx 0.2$, are very unbalanced in module `seclib_2`, which explains the results obtained in Fig. 7 default mode: std_dev(11) > std_dev(10). Anyway, we recall that only the detailed mode provides a sufficiently accurate estimation of the nets average unbalancedness, hence of the layout static security.

TABLE 3
SubBytes dual-rail blocks capacitive dispersion, computed from the statistics collected in Fig. 7 and 8.

#	Nickname	Std_dev (default mode)	Std_dev (detailed mode)	$\frac{C_{wire}}{C_{total}}$
(5)	`wddl_0`	68.58 $\times 10^{-3}$	77.71 $\times 10^{-3}$	55 %
(6)	`wddl_1`	2.73 $\times 10^{-3}$	7.62 $\times 10^{-3}$	65 %
(7)	`wddl_2`	1.21 $\times 10^{-3}$	2.67 $\times 10^{-3}$	68 %
(8)	`wddl_4`	0.94 $\times 10^{-3}$	3.56 $\times 10^{-3}$	70 %
(9)	`ewddl_4`	0.00 $\times 10^{-3}$	2.44 $\times 10^{-3}$	52 %
(10)	`seclib_1`	0.26 $\times 10^{-3}$	4.95 $\times 10^{-3}$	52 %
(11)	`seclib_2`	0.30 $\times 10^{-3}$	0.81 $\times 10^{-3}$	57 %
(12)	`seclib_4`	0.00 $\times 10^{-3}$	0.62 $\times 10^{-3}$	55 %

A similar analysis as the one of Sec. 2.2 is carried out in detailed extraction mode, regarding only static evaluators for the routing. The expected level of security is depicted in Fig. 9. This figure shows that the security level of competing designs can be predicted using methods inspired from the two-dimensional chromatography. Notice that, compared to the overall security expectation (taking into account both the *logic gates* and their *interconnect*) discussed in Sec. 2.2, a new relationship is established: (9) is assumed to be of equal quality as (12), because:

- eWDDL and SecLib have balanced C_{gate} (security feature $\mathcal{B}1$), and
- their interconnect is balanced with the same differential features $\mathcal{B}2$ to $\mathcal{B}5$.

If we compare this figure (Fig. 9) and the statistical results obtained in Tab. 3, it appears that the predictions are all valid, but for the effect of the pairs shielding. Indeed, we have predicted $(7) \prec (8)$ and $(11) \prec (12) = (9)$, but we have neither std_dev(7) > std_dev(8) nor std_dev(11) > std_dev(9). The reason might be that the SubBytes modules are too small for the metal lines to have the opportunity to be cross-coupled. The effect of the shield is merely to increase globally the routing length, and thus paradoxically to increase unequally the capacitive parasitics. This agrees with this intuitive observation on the larger modules (11) & (12): they do satisfy $(11) \prec (12)$. Therefore, the two violations $(7) \not\prec (8)$ and $(11) \not\prec (9)$ can safely be considered artifacts that do not scale up for a complete algorithm protection, with many substitution boxes and a complex datapath.

4 Experimental Comparison of the Thirteen SubBytes Modules

4.1 Implementation into a Single-Chip Prototyping ASIC

The thirteen SubBytes modules studied in the previous section have been implemented in an ASIC. Their posi-

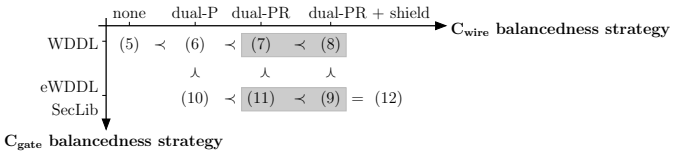


Fig. 9. Expected level of security partial order, based on the sole static criterion. The gray boxes indicate security relationships that are violated by the extraction in "detailed mode" statistics.

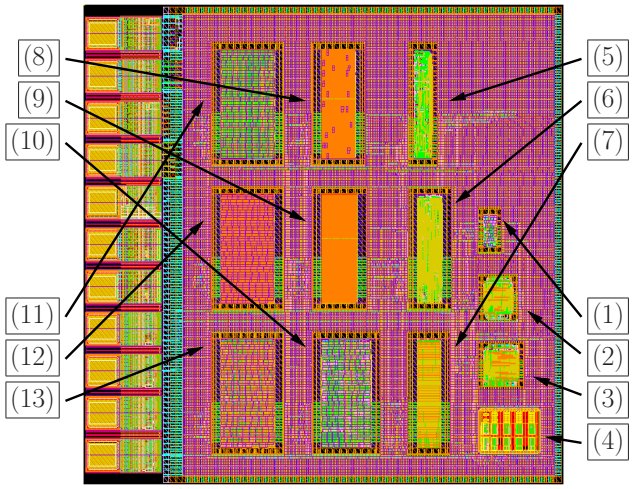


Fig. 10. The `SubBytes` circuit's layout.

tion on the floorplan is indicated in Fig. 10. There are only four functional I/O pads, common to all SubBytes modules: this way, they are all evaluated under the same experimental conditions. The I/O pads are:

1) `clk`: a global clock to synchronize the executions,
2) `data_in`: an input serial line,
3) `data_out`: an output serial line,
4) `enable`: a selection signal deciding whether the circuits loads bits serially from the outside or transfer them in parallel into the substitution boxes.

To reduce noise, pads, core and SubBytes modules are powered from three different sources, all operating under a nominal 1.2 V voltage. The list of all the pads is given in Fig. 11.

The pictures of the ASIC and of its DIL48 (Dual In-Line package with 48 pins) cavity are given in Fig. 12.

4.2 `SubBytes` Programming Model

For the thirteen SubBytes modules to be operated in a unified way, they require a common programming paradigm. The chip architecture is based on a shift-register for serial registers load and flush. Thanks to a two-stage pipeline at the input and one-stage pipeline at the output of the SubBytes blocks, the data are presented in front of all SubBytes modules. To suppress the power consumption of one specific module we freeze its inputs. For example, it can be always loaded the

	Pad name	Nature	#
	VDD_SUBBYTES_1V2	vdd1V2	31
	VSS_SUBBYTES_1V2	vss1V2	32
	VDD_CORE_1V2	vdd1V2	34
	VSS_CORE_1V2	vss1V2	35
	VSS_IOREF_CORE_1V2*	vss1V2	36
	VDD_PAD_3V3	vdd1V2	37
	VSS_PAD_3V3	vss1V2	38
Fonctional I/O pads	data_out	out	39
	data_in	in	40
	enable	in	41
	clk	in	42

*unconnected

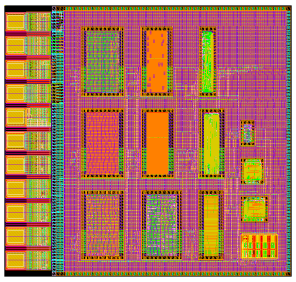


Fig. 11. Datasheet on the `SubBytes` circuit's pads.

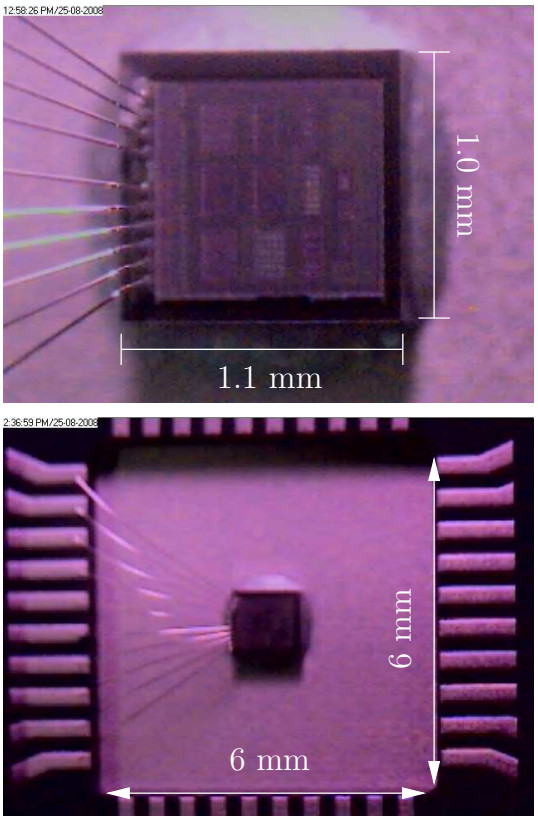


Fig. 12. The `SubBytes` circuit's monographs, as seen from an optical microscope.

same data, say `0x00`. As a result, the circuit simply comprises $3 \times n$ flip-flops (DFFs), where `n` is the total number of inputs of the combinatorial gates. Synthesis and place-and-route were performed with Cadence tools. The synthesizer is `bgx_shell v05.15-s095+1`, used with option `-BGX` for improved results on high-level behavioral VHDL [29] source code. The backend is realized by `First Encounter v04.10-s415_1` and the interconnection routing by `NanoRoute v04.10-s914`. The chip was fabricated through the silicon broker CMP, that prepares the final layout and delegates the actual fabrication to STMicroelectronics' foundries.

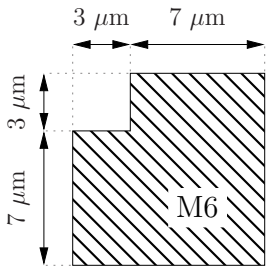


Fig. 13. M6 pattern for EMA-shield using a metal-plane mirror.

TABLE 4
Number of distinct power measurements to realize on the SubBytes instances to fully characterize their signature.

Instance #	Transition count	Description
(1, 2, 3, 4)	$2^{2\times 8} = 65\,536$	$\forall i, f : i \to f$
(5, 6, 7, 8, 9)	$4 \times 2^8 = 1\,024$	$\forall i : 0 \to i, i \to 1, 1 \to i, i \to 0$
(10, 11, 12, 13)	$2 \times 2^8 = 512$	$\forall i : 0 \to i, i \to 0$

The vertical routing direction has been chosen for M3 and M5 and the horizontal for M4 and M6.

As for the top-level metal M6 used to protect the circuit against EMA, it is actually not permitted to use it uniformly, due to stringent design rules about thermal stress. Instead, the so-called "metal-slot" design rules state that 9% of holes must be spread over the plane. The plane is thus a mesh obtained by the replication of the pattern depicted in Fig. 13.

4.3 Experimental Environment

4.3.1 Enumeration of Required Power Traces Measurements for a Comprehensive Evaluation

The power measurements come down to testing the combinatorial functions exercised with all the possible transitions. For unprotected instances, the transitions consist in changes from an initial value $i \in [0, 2^8[$ to a final value $f \in [0, 2^8[$. For secured instances, the protocol consists in transitions between a spacer and a valid state. The WDDL instances can be used both with the $\{00\}^8$ and the $\{11\}^8$ spacers, whereas only the null spacer $\{00\}^8$ is usable (unless making the gate insecure) for the SecLib-based instances. The number of measurements is summarized in Tab. 4.

4.3.2 Acquisition Platform

The acquisition is managed by a central personal computer, that dialogues with:

- the device under test (DUT), namely the `SubBytes` ASIC, driven by an `ACME fox` (`http://www.acmesystems.it/`) development board, and
- a digital oscilloscope, in charge of acquiring traces and storing them in a `postGreSQL` database server.

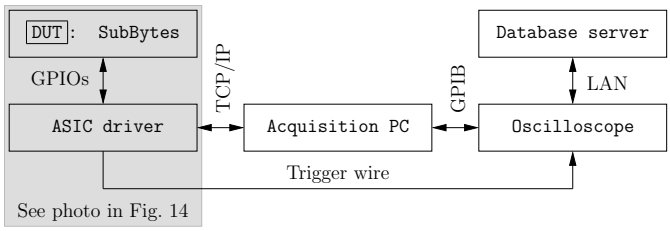


Fig. 14. Acquisition platform for `SubBytes` power traces.

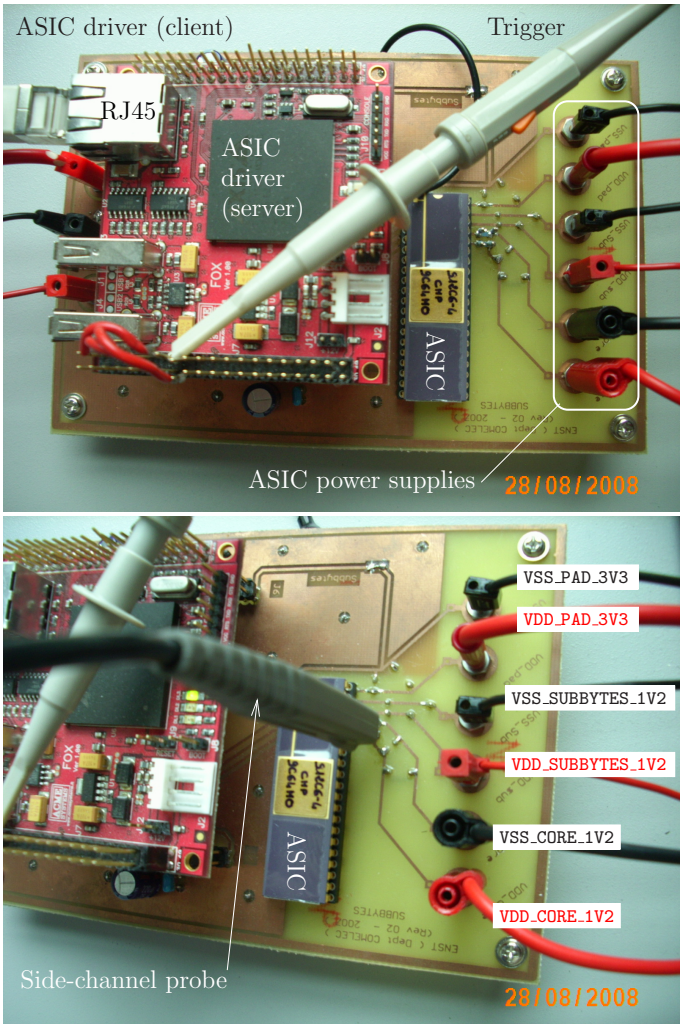


Fig. 15. Control board for `SubBytes` (ASIC under test) power traces.

The acquisition architecture is depicted in Fig. 14. Two photographs of the *in-house* platform driving `SubBytes` are shown in Fig. 15.

4.4 Experimental Evaluation Metrics

4.4.1 Definition of $M1$: Maximum Standard Deviation over a Complete Trace

To compare the diverse implementations, the following metric M_1 is used:

- let $P(x \to y)(t)$ a power trace, acquired by the platform shown in Fig. 15, with x and y in [0x00, 0xff] and t the time in one clock period $[0, T[$,
- let $P(t)$ be the average power trace over all the x (initial value) and y (final value),
- let $\sigma(t)$ be the traces standard deviation: $\sigma(t) \doteq \sqrt{\frac{1}{2^8 \times 2^8} \sum_{x,y} \left(P(x \to y)(t) - P(t)\right)^2}$ (notice that $\sigma(t)$ is also a trace: it has as many points t as the any original trace),
- let M_1 be the maximum value taken by $\sigma(t)$ over all dates t.

M_1 focuses on the highest bias on a clock period, which makes sense in cryptographic applications where any singularity is exploited. It also concurs with the "mono-variate" bias one DPA will identify as the most leaking instant that correlates best with the leakage model.

Two other metrics, called M_2 and M_3, are also considered, as variations.

4.4.2 Definition of $M2$: Mean Standard Deviation over a Complete Trace

M_2 is the integral of the standard deviation over one clock period T, that is to say $\frac{1}{T} \int_{t=0}^{t=T} \sigma(t)\,\mathrm{d}t$. The metric M_1 is meant to be more stringent than M_2. However, $M2$ grasps variations over a full execution of SubBytes. It closely relates to "multi-variate" analyses, such as *templates* with a principal component analysis [30] where the principal direction is a step function over the evaluation clock period.

4.4.3 Definition of $M3$: Standard Deviation of an Averaged Trace

M_3 models a low-cost attack, where the attacker is supposed not to be equipped with a fast oscilloscope. The simulation of this scenario is obtained by first averaging the traces over one entire clock period, resulting in $P(x \to y) \doteq \frac{1}{T} \int_t P(x \to y)(t)\,\mathrm{d}t$. The metric M_3 is defined as the standard deviation of $P(x \to y)$.

4.4.4 Comparison and Analysis of Metrics

Table 5 presents the three metrics calculated from these measurements. Due to a design error, the ROM (module number 4) is not fully functional (some addresses are unavailable). It is thus excluded from the table.

The single-ended modules (1), (2) & (3) are evaluated based on

1) one computation per clock cycle (65 536 averages) and
2) one computation every other clock cycle, with a precharge to zero in-between (256 averages).

It clearly appears that the single-ended modules operated with a throughput of one computation per clock cycle are much less secure than any dual-rail logic (5), (6), $\cdots$, (13). The gain of the dual-rail logic over classic CMOS logic is thus undebatable.

However, it is interesting to notice that some classic logics are affected by the sole use of a precharge. If we consider an interleaved precharge to 0x00, Tab. 5 shows that:

TABLE 5
Metrics for 12 implementations of SubBytes.

#	Nickname	10^3 M1	10^3 M2	10^3 M3
On 65 536 traces ($\forall i, f \in [0\text{x}00, 0\text{x}ff]^2 : i \to f$).				
(1)	stdcell_gf	76.174	21.162	17.651
(2)	stdcell_lut	122.231	29.742	20.123
(3)	stdcell_gb	228.515	23.677	6.290
On 256 traces ($\forall f \in [0\text{x}00, 0\text{x}ff] : 0\text{x}00 \to f$).				
(1)	stdcell_gf	83.903	21.828	19.488
(2)	stdcell_lut	82.038	21.838	17.644
(3)	stdcell_gb	25.087	8.257	5.661
(5)	wddl_0	23.526	5.795	0.907
(6)	wddl_1	29.558	6.084	0.846
(7)	wddl_2	31.392	6.473	0.750
(8)	wddl_4	32.367	6.329	0.800
(9)	ewddl_4	40.250	8.050	1.054
(10)	seclib_1	14.824	4.556	0.766
(11)	seclib_2	13.978	4.889	0.837
(12)	seclib_4	11.897	4.404	0.729
(13)	seclib_4ema	15.593	4.681	0.806

- module (1), stdcell_gf, is not affected by the insertion of the spacer,
- module (2), stdcell_lut, becomes slightly more secure, whereas
- module (3), stdcell_gb, becomes drastically more secure.

It is remarkable that implementation (3) which is based on Guido Bertoni's architecture seems less vulnerable than the two other standard cell based implementations. This could be explained by the architecture. The architecture is in fact divided into three steps decode/permute/encode among which only the last encode step is input-dependent. It is based on a glitch-free 1-out-of-256 decomposition, that signs the same irrespective of the input, unless two consecutive inputs happen to be identical (in which rare case there is no dissipation at all). It demonstrates that a well-balanced architecture can reduce information leakage at a very low-cost in term of silicon area. The throughput is divided by two, which is anyway an overhead that dual-rail logics also have to pay for.

In the sequel, we study the metrics for only the 256 transitions corresponding to all possible 8-bit inputs preceded by a precharge phase to zero. As for dual-rail logic, Table 5 also proves the importance of synchronization as SecLib seems more secure than WDDL (See Appendix A for detailed power trace figures). It is however difficult to evaluate the gain of the differential routing on top of the differential placement. The only noting that holds for sure is that differential routing associated to shielding of dual pairs improves the security: (10) is indeed more dispersive than (12). This applies to SecLib, but not to WDDL, where the dispersion due to logic is the overwhelming source of dispersion: for WDDL, the

more backend counter-measures, the larger the module, hence the more intense the information leakage.

One other remark is related to the metrics for `ewddl_4`. In fact, it was expected that the replacement of `AND` and `OR` gates by the `Enhanced-AND` and `Enhanced-OR` (Figure 3) improves the symmetry of the design. But according to the measurements, this has increased the dispersion. This makes us tend to believe that early evaluation is predominant against technological asymmetry. Indeed, eWDDL, as WDDL, is prone to early evaluation; as eWDDL is based on more complex gates than WDDL (`MAJ` instead of `AND/OR`), the propagation time through the logic is increased[2], which exacerbates the early evaluation because it is cumulative along the combinatorial paths.

4.4.5 Confrontation With an Information Theoretic Metric

The level of robustness of a counter-measure can also be evaluated by the quantity of information it leaks. This approach requires an approximation of the probability distribution function (PDF) for one trace to actually match the correct input used during the acquisition. In our setup, we have a close to perfect estimation of the leakage trace for every possible input. By design, the computation of the substitution box is not disturbed by other unrelated activity and the high averaging rate of the oscilloscope greatly improves the signal's vertical resolution. However, it can be interesting to extrapolate the information available from each SubBytes block when the measurements are noisy, as in operational situations. The noise can, for instance, model the activity of surrounding logic gates, which will happen in practice, since SubBytes is customarily embedded into a complete datapath with other substitution boxes. We thus introduce an artificial noise parameter σ. It is equal to the width of the PDFs, assumed to be Gaussians of identical variance σ^2 for any substitution box input.

Our evaluation is inspired from the one carried out by simulation on single logical gates [31][3]. We replaced the simulations by the real measurements and the logic gates by a complete netlist of combinatorial gates making up the SubBytes instances. The dual-rail with precharge substitution boxes embedded in SubBytes correspond to the **Pre-Charged / not Masked Logic Styles** paragraph in Sec. 3.2 of [31]. Therefore, we compute the mutual information as per Eqn. (1), using notations of [31]:

$$\mathrm{I}(S_g, \mathbf{L}_{S_g}^{q=2^8}) = \mathrm{H}(S_g) - \mathrm{H}(S_g|\mathbf{L}_{S_g}^{q=2^8}) = \tag{1}$$
$$8 - \sum_{s_g=\mathtt{0x00}}^{s_g=\mathtt{0xff}} \Pr(s_g) \int_{\mathbf{l}} \Pr(\mathbf{l}|s_g) \log_2 \frac{\Pr(\mathbf{l}|s_g)}{\sum_s \Pr(\mathbf{l}|s)} \,\mathrm{d}\mathbf{l}\,.$$

We use for the input distribution $\Pr(s_g)$ a uniform law over $[\mathtt{0x00}, \mathtt{0xff}]$ and for $\Pr(\mathbf{l}|s_g)$ a multi-variate Gaussian distribution of mean the measurements and of covariance matrix a multiple of the identity of $]0, +\infty[^{T\times T}$.

The integration over all the samples is simplified by a principal component analysis (PCA) of the curves. Thanks to the pre-processing described in [30], we managed to replace all the initial samples of the curves by one single sample. The number of significative components in the PCA validates the limitation to one single sample; this makes it possible to simplify Eqn. (1) from a multi- to a single-valued integral.

The result is plotted in Fig. 16. In this graph, the lowest curves are the most secure. It can be seen that the conclusions already drawn in Sec. 4.4.4 still hold. The single-ended logics disclose more input bits than WDDL, that in turn is less secure than SecLib. We continue to note that the single-rail architecture of Guido Bertoni *et al.* performs almost as good as WDDL. Also, it appears clear the SecLib has a serious security improvement over WDDL. We also confirm that the eWDDL style does not improve WDDL, but instead makes it worse, certainly due to an exacerbated early evaluation propagation. Finally, some behavior amongst the SecLib modules are difficult to interpret, like for instance `seclib_2` that is less secure than the other SecLib modules, but for a narrow window of noise. It is nonetheless certain that SecLib with all the protections set (but without the M6-shield, namely `seclib_4`) is the most secure implementation. One final observation can be made: the $\mathrm{I}(S_g, \mathbf{L}_{S_g})$ curves for SecLib have a discontinuity when it is equal to 8 bits and the noise increases, whereas the behavior for WDDL, eWDDL and single-ended logics is continuous. This means that WDDL, eWDDL and single-ended logics have homogeneously distributed biases. At the opposite, SecLib traces have very few discrepancies when the inputs change: the discontinuity is probably due to a very small number of particularities for some rare inputs. This analysis shows that, should a designer be able to identify those discrepancies, the security level of SecLib could be easily improved.

5 Design-Time Security Evaluation and Backend-Level Counter-Measures Analysis

This section gathers the lessons learnt from the previous design-time (Sec. 3) and *in silico* (Sec. 4) evaluations. The efficiency of the logic styles and backend refinements is also discussed.

5.1 Reflections About High-Level Security Evaluation

High-level evaluations based on static analyses, such as [33] routing unbalancedness estimation, happen to be irrelevant. Indeed, experimental results show that for logics that do not synchronize the signals, the predominant source of unbalancedness is the relative arrival times of inputs. Depending on them and on the values of

2. In STM HCMOS9GPLL library, the average propagation time through the unload unitary `AND` (*resp.* `MAJ`) gate is 81 ps (*resp.* 146 ps).

3. This work has been extended recently on a four-bit datapath of PRESENT in [32].

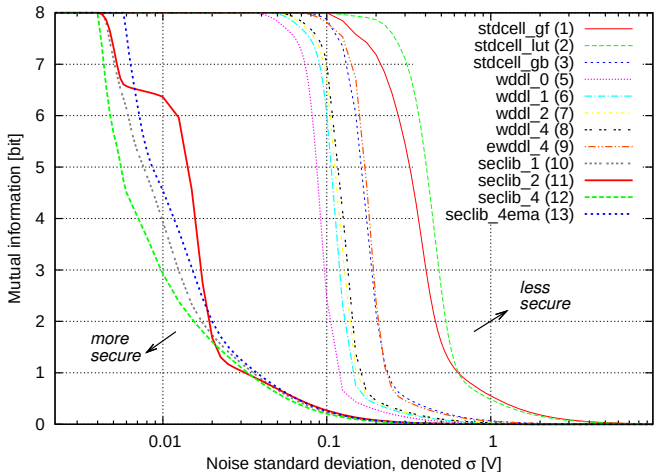


Fig. 16. Mutual information leaked by the implementations of SubBytes using the `0x00` spacer for precharge, in the hypothesis of noise homoscedasticity over all the different inputs.

the inputs, the logic evaluates earlier or later. This early evaluation issue is thus a dynamic problem. It is several orders of magnitude more important than the dispersion of routing characteristics. A correct high-level security evaluation of non-synchronizing logics (such as AND-OR based logics) must thus resort to simulations or to techniques taking the timing behavior into account.

Notice that this remark does not apply to SecLib, since the very structure of this logic makes it possible to decouple the gates from their interconnection. Indeed, static (netlist-level) and dynamic (silicon-level) results agree.

The silicon-level measurements also revealed that amongst unprotected single-rail implementations of SubBytes, some can be almost as secure as WDDL or SecLib. The logic in question is that of Guido Bertoni: as every execution implies a decoding, all inputs activate roughly the same number of gates. Put differently, all execution paths are almost indiscernible: this appears clearly on Fig. 1, where a typical execution path is highlighted. Whatever the input byte, the decoder sets only one bit amongst 256 to '1', that is driven to exactly $8/2$ encoders (because SubBytes is balanced) all having the same structure. Therefore, even if this logic is larger than other unprotected descriptions, it remains smaller than WDDL and much smaller than SecLib circuits, for a comparable security level.

An other interesting point is about the M6-shielded SecLib instance. Eric Peeters already showed in the chapter 5 of his PhD thesis manuscript [34] that:

> "Metallic shield must be tamper resistant as well, because when connecting a differential probe on it, we were able to observe a data-dependent voltage. As a matter of fact, the metallic shield is turned into a very near-field electric probe."

We observe that a metallic shield increases the dissymmetry of an underneath DPL design. A "self-induction" effect might be the cause of such an effect. But for sure, the conclusion is that the usefulness of a top-level metallic shield is far from being obvious.

5.2 Summary About Security-Cost Trade-Offs

The previous analyses have made clear that some *would-be* counter-measures actually both increase the implementation cost and degrade the security level. This is case of eWDDL and the top-level electromagnetic shield. Those two solutions must positively be proscribed.

We note for the time that a non-protected single-rail logic can be made more security simply by interleaving every computation by a precharge to a constant value, such as `0x0`. The impact in terms of silicon area is negligible, but the throughput is divided by two. The other counter-measures, labeled $\mathcal{B}1$ to $\mathcal{B}5$, increase the security level. However, they are actually useful only if the logic is immune to early evaluation. SecLib is in the *silicon-domain* (as opposed to the *wire-domain*), which means that the area of the cells is limiting the density and not the congestions in the interconnect resources. Therefore, in the case of SecLib, The gain they convey by the accumulation of security features is visible in terms of security, and in the meantime also free in hardware, since $\mathcal{B}3$ to $\mathcal{B}5$ complexify the routing, which is not a critical resource.

5.3 Suitability of an Elementary Pattern Circuits for Security Evaluations

The backend-level improvements do not translate into an observable security increase as for WDDL, because we identified that the early evaluation is overwhelmingly the predominant dispersive feature. Nonetheless, we could have expected SecLib to disclose improvements with the backend design care. Paradoxically enough, it is not straightforward to appreciate the impact of backend features on SecLib dispersion. This might be due to the over-simplification of the design; if the SubBytes instances were not insulated (not from the substrate noise but from other noisy instances by a large on-chip spacing), they would be more coupled with extrinsic activity (referred to as "algorithmic noise" in the context of attacks against cryptoprocessors [2], [35]). In this case, we could observe that SubBytes instances with poor backend features would be more influenced by this coupling than full-featured SecLib SubBytes instances. Unfortunately, we cannot verify this hypothesis on the ASIC: do poorly routed and unshielded SecLib instances appear more secure than they really are because of an evaluation artifact?

6 Conclusions and Perspectives

6.1 Conclusions

DPL styles are designed and used to counter-act DPA attacks by making the power consumption constant.

There are several DPL logics such as WDDL and SecLib, respectively based on standard cells and totally customized cells forcing signals synchronization. In this paper we compare these two logics by analyzing the power dispersion of a combinatorial block, the AES substitution box (SubBytes). Our analysis demonstrates that dual-rail logic implementations are indisputably more secured than single-rail logics. We find out that choosing a balanced architecture such as described by Guido Bertoni *et al.* combined with a precharge to zero does reduce the power dispersion impressively, thus increasing the security level against power analysis attacks. We also demonstrate that SecLib is less dispersive than WDDL, confirming experimentally that signals synchronization is important to avoid data-dependent early evaluation and precharge. The security benefits of second-order countermeasures, such as differential placement, routing, dummies and shield against cross-talk are observed on SecLib.

6.2 Perspectives

As static high-level security evaluations are not accurate enough, netlist temporal simulation must be used instead for pre-fabrication validation purposes. This approach has been initiated for instance in [36] with logic simulation (ideal transitions). To further model signals slopes, fast gate-level or transistor-level simulations are mandatory. Efforts in this direction have already been deployed, *e.g.* by Huiyun Li *et al.* [37] or by Giorgio Di Natale *et al.* [38].

We emit the hypothesis that results on SecLib instances of `SubBytes` were evaluated optimistically because of the absence of neighbour logic, and that the impact of coupling cannot be assessed. We suggest to consider FPGAs as prototyping platforms: FPGAs do not exactly behave SCA-wise as ASICs (even at constant technology); nevertheless they allow to better iterate and test more configurations. For instance, the SASEBO boards [39] with the EveSoC environment [40] can be such a commodity.

Appendix A
Traces Showing Power Dispersion for Twelve Implementations of SubBytes

Figures 17 and 18 show the power dispersion measured for the 256 possible inputs ($\forall f \in [\texttt{0x00}, \texttt{0xff}], \texttt{0x00} \rightarrow f$) respectively for standard cell logic, and dual-rail logic — WDDL *versus* SecLib.

The acquisition chain characteristics are listed below:

- The probe's bandwidth is 5 GHz;
- The sampling rate of the acquisition apparatus (Infiniium 54 855A sold by Agilent) is 20 Gsample/s;
- The vertical caliber is 1 mV;
- The curves are averaged 256 times by the oscilloscope, leading to 12-bit vertical resolution;

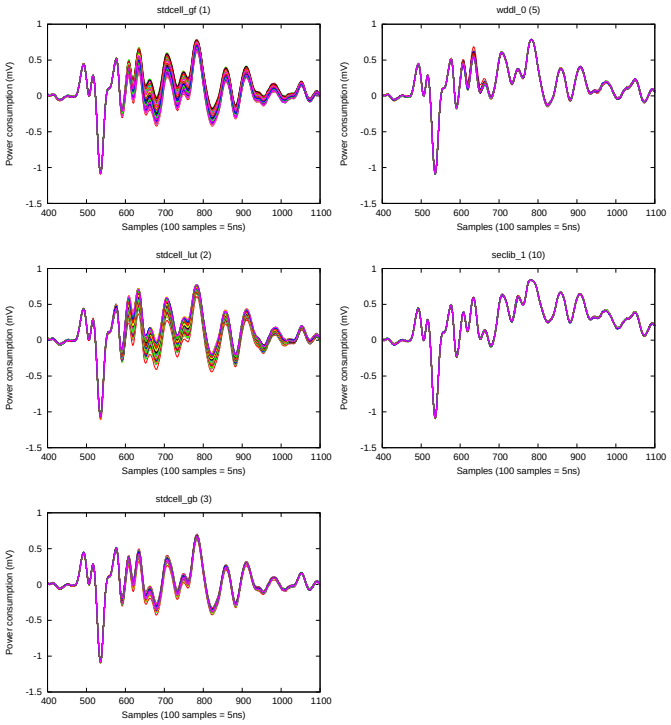


Fig. 17. Power traces for 256 inputs with `0x0` or `0x00` precharge — comparison between standard cell logics and dual-rail logics.

The traces are displayed raw: no post-processing has been done to correct their shape. Compared to a crypto-processor's regular trace (such as the example given in Fig. 6), the average is non-zero after evaluation. This is due to the fact that the SubBytes modules, the power consumption of which is measured, are not electrically insulated from the rest of the `SubBytes` internal logic. Hence a cross-coupling between several parts of the silicon die, that induce a background noise. As the same programmation sequence is employed to test every SubBytes block, the cross-coupling effect is a constant phenomenon that merely adds up to the relevant measurements. Because it is the same irrespectively of the addressed SubBytes module, this "continuous component" can safely be ignored.

Acknowledgments

This work has been partly financed by the french conseil régional "Provence Alpes Côte d'Azur" (Région PACA) and by the SCS (Solutions Communicantes Sécurisées) competitivity cluster via the CALISSON project. We are grateful to STMicroelectronics AST (Advanced System Technology) department for having launched and encouraged this project, to CNFM (Coordination Nationale pour la Formation en Micro et nanoélectronique) for CAD tools licenses and to CMP (Circuits Multi-Projets) for subcontracting the chip fabrication and packaging. We thank Karim Benkalaia, from COMELEC department of TELECOM ParisTech, for the design and the test PCB

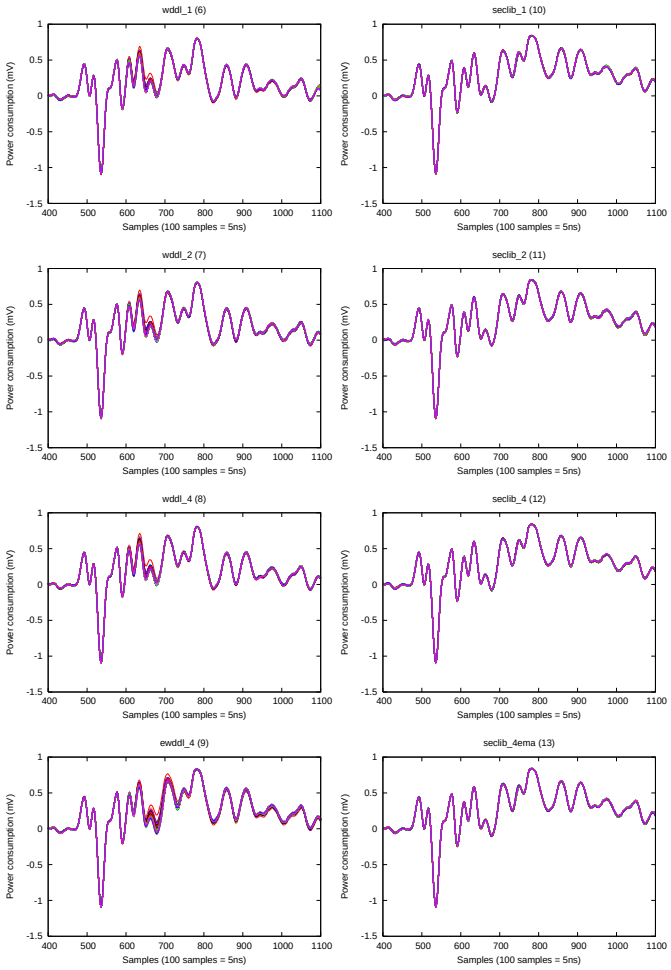


Fig. 18. Power traces for 256 inputs with `0x00` precharge — comparison between WDDL and SecLib.

for deported ICs, such as `SubBytes`. We acknowledge interesting discussions with Guido Bertoni, Jean-Luc Danger and Yves Mathieu, as well as relevent suggestions of improvements from the anonymous reviewers.

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis: Leaking Secrets," in *CRYPTO'99*, ser. LNCS, vol. 1666. Springer, August 1999, pp. 388–397, Santa Barbara, California, USA. Online version: http://www.cryptography.com/resources/whitepapers/DPA.pdf.

[2] É. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *CHES'04*, ser. LNCS, vol. 3156. Springer, August 11–13 2004, pp. 16–29, Cambridge, MA, USA.

[3] J.-L. DANGER, S. GUILLEY, S. BHASIN, and M. NASSAR, "Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors, — New Attacks and Improved Counter-Measures —," in *SCS*, ser. IEEE, November 6–8 2009, Jerba, Tunisia.

[4] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in *DATE'04*. IEEE Computer Society, February 2004, pp. 246–251, Paris, France.

[5] S. Guilley, F. Flament, R. Pacalet, P. Hoogvorst, and Y. Mathieu, "Security Evaluation of a Secured Quasi-Delay Insensitive Library," in *DCIS, full text in HAL, http://hal.archives-ouvertes.fr/hal-00283405/en/*, November 2008, pp. 1–7, DCIS'08, Grenoble, France.

[6] T. Popp and S. Mangard, "Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints," in *Proceedings of CHES'05*, ser. LNCS, LNCS, Ed., vol. 3659. Springer, Sept 2005, pp. 172–186., Edinburgh, Scotland, UK.

[7] P. Schaumont and K. Tiri, "Masking and Dual Rail Logic Don't Add Up," in *CHES*, ser. LNCS, vol. 4727. Springer, 2007, pp. 95–106, Vienna, Austria.

[8] NIST/ITL/CSD, "FIPS PUB 197: Advanced Encryption Standard (AES)," November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[9] S. Guilley, P. Hoogvorst, Y. Mathieu, R. Pacalet, and J. Provost, "CMOS Structures Suitable for Secured Hardware," in *DATE'04*. IEEE Computer Society, February 2004, pp. 1414–1415, Paris, France.

[10] Simon Moore and Ross Anderson and Robert Mullins and George Taylor and Jacques J.A. Fournier, "Balanced Self-Checking Asynchronous Logic for Smart Card Applications," *Journal of Microprocessors and Microsystems*, vol. 27, pp. 421–430, October 2003.

[11] V. Rijmen, "Efficient Implementation of the Rijndael S-box," informal communication.

[12] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient Rijndael Encryption Implementation with Composite Field Arithmetic," in *CHES*, ser. LNCS, vol. 2162. London, UK: Springer-Verlag, May 2001, pp. 171–184.

[13] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES SBoxes," in *CT-RSA*, ser. LNCS, vol. 2271. Springer, 2002, pp. 67–78.

[14] G. Bertoni, M. Macchetti, L. Negri, and P. Fragneto, "Power-Efficient ASIC Synthesis of Cryptographic S-Boxes," in *GLSVLSI '04: Proc. of the 14th ACM Great Lakes symposium on VLSI*. ACM, April 2004, pp. 277–281, Boston, MA, USA.

[15] M. Giaconia, M. Macchetti, F. Regazzoni, and K. Schramm, "Area and Power Efficient Synthesis of DPA-Resistant Cryptographic S-Boxes," in *VLSI Design*. IEEE Computer Society, 6-10 January 2007, pp. 731–737, Bangalore, India.

[16] S. Tillich, M. Feldhofer, and J. Großschädl, "Area, Delay, and Power Characteristics of Standard-Cell Implementations of the AES S-Box," in *SAMOS*, ser. LNCS, vol. 4017. Springer-Verlag, July 17-20 2006, pp. 457–466, Samos, Greece.

[17] S. Tillich, M. Feldhofer, T. Popp, and J. Großschädl, "Area, delay, and power characteristics of standard-cell implementations of the AES S-Box," *J. Signal Process. Syst.*, vol. 50, no. 2, pp. 251–261, 2008.

[18] D. Suzuki and M. Saeki, "Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style," in *CHES'06*, ser. LNCS, vol. 4249. Springer, 2006, pp. 255–269.

[19] M. Shams, J. Ebergen, and M. Elmasry, "Modeling and comparing CMOS implementations of the C-Element," *IEEE Transactions on VLSI Systems*, vol. 6, no. 4, pp. 563–567, December 1998.

[20] S. Guilley, P. Hoogvorst, Y. Mathieu, and R. Pacalet, "The "Back-end Duplication" Method," in *CHES'05*, ser. LNCS, vol. 3659. Springer, August 2005, pp. 383–397, Edinburgh, Scotland, UK.

[21] S. Guilley, F. Flament, R. Pacalet, P. Hoogvorst, and Y. Mathieu, "Secured CAD Back-End Flow for Power-Analysis Resistant Cryptoprocessors," *IEEE Design & Test of Computers, special issue on "Design and Test of ICs for Secure Embedded Computing"*, vol. 24, no. 6, pp. 546–555, November–December 2007.

[22] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *CHES'01*, ser. LNCS, vol. 2162. Springer, May 2001, pp. 251–261.

[23] S. Guilley, S. Chaudhuri, L. Sauvage, P. Hoogvorst, R. Pacalet, and G. M. Bertoni, "Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks," *IEEE Transactions on Computers*, vol. 57, no. 11, pp. 1482–1497, November 2008.

[24] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," in *Proceedings of CHES'05*, ser. LNCS, LNCS, Ed., vol. 3659. Springer, September 2005, pp. 157–171., Edinburgh, Scotland, UK.

[25] S. Guilley, P. Hoogvorst, and R. Pacalet, "Differential Power Analysis Model and some Results," in *CARDIS'04*. Kluwer, August 2004, pp. 127–142, Toulouse, France.

[26] E. Prouff, "DPA Attacks and S-Boxes," in *FSE'05*, ser. LNCS, vol. 3557. Springer-Verlag, February 2005, pp. 424–441, Paris, France.

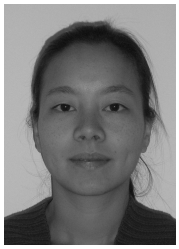
[27] C. Carlet, "On Highly Nonlinear S-Boxes and Their Inability to Thwart DPA Attacks," in *INDOCRYPT'05*, ser. LNCS, vol. 3797. Springer, December 2005, pp. 49–62, Bangalore, India; Complete version on IACR ePrint 2005/387).

[28] S. Guilley, P. Hoogvorst, R. Pacalet, and J. Schmidt, "Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties," in *BFCA*, 2007, pp. 1–25, May 02–04, Paris, France, http://www.liafa.jussieu.fr/bfca/books/BFCA07.pdf.
[29] Institute of Electrical and Electronics Engineers (http://www.ieee.org/), "IEEE Standard VHDL (Very High Speed Integrated Circuits Description Language) Reference Manual," pp. 1–300, ISBN: 0-7381-3247-0 2002.
[30] C. Archambeau, É. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Template Attacks in Principal Subspaces," in *CHES*, ser. Lecture Notes in Computer Science (LNCS), vol. 4249. Springer, 2006, pp. 1–14.
[31] F. Macé, F.-X. Standaert, and J.-J. Quisquater, "Information theoretic evaluation of side-channel resistant logic styles," in *CHES*, ser. LNCS, vol. 4727. Springer, September 2007, pp. 427–442, Vienna, Austria.
[32] F. Regazzoni, A. Cevrero, F.-X. Standaert, S. Badel, T. Kluter, P. Brisk, Y. Leblebici, and P. Ienne, "A Design Flow and Evaluation Framework for DPA-Resistant Instruction Set Extensions," in *CHES*, ser. Lecture Notes in Computer Science, vol. 5747. Springer, 6-9 September 2009, pp. 205–219.
[33] K. Tiri and I. Verbauwhede, "Place and Route for Secure Standard Cell Design," in *Proceedings of WCC / CARDIS*, August 2004, pp. 143–158, Toulouse, France.
[34] É. Peeters, "Towards Security Limits of Embedded Hardware Devices: from Practice to Theory," Ph.D. dissertation, Université catholique de Louvain, Belgium; UCL Crypto Group, Nov 2006.
[35] N. Hanley, R. McEvoy, M. Tunstall, C. Whelan, C. Murphy, and W. P. Marnane, "Correlation Power Analysis of Large Word Sizes," in *ISSC (Irish Signals and System Conference)*. IET, 13-14 Sept 2007, pp. 145–150, Edinburgh, Scotland, UK.
[36] S. Guilley, S. Chaudhuri, L. Sauvage, T. Graba, J.-L. Danger, P. Hoogvorst, V.-N. Vong, and M. Nassar, "Place-and-Route Impact on the Security of DPL Designs in FPGAs," in *HOST*. IEEE Computer Society, 2008, pp. 29–35, june 9, Anaheim, USA. ISBN: 978-1-4244-2401-6.
[37] H. Li, A. Markettos, and S. Moore, "A security evaluation methodology for smart cards against electromagnetic analysis," in *Security Technology, 2005. CCST'05. 39th Annual 2005 International Carnahan Conference on*. IEEE, 11-14 Oct. 2005, pp. 208–211.
[38] G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "An Integrated Validation Environment for Differential Power Analysis," in *DELTA*. Hong Kong: IEEE Computer Society, January 2008, pp. 527–532.
[39] A. Satoh, "Side-channel Attack Standard Evaluation Board, SASEBO," project of the AIST – RCIS (Research Center for Information Security), http://www.rcis.aist.go.jp/special/SASEBO/.
[40] EveSoC software, "A side-channel eavesdropping system-on-chip, http://sourceforge.net/projects/evesoc/."

Sylvain Guilley belongs to the french interministerial body of telecommunication engineers (now "Corps des Mines"). He graduated from the École Polytechnique (X1997) and from the École Nationale Supérieure des Télécommunications (ENST) in 2002. In 2002, he also received the M.S. of quantum physics from the École Normale Supérieure (ENS). He got a PhD *summa cum laude* from TELECOM ParisTech (new brand name of the ENST) in 2007 on the topic of backend countermeasures against side-channel attacks. Since 2002, he is associate professor with the VLSI group at TELECOM ParisTech. His research interests are the security of cryptographic hardware (ASIC and/or FPGA) and the specification of provable trusted computing platforms. Sylvain Guilley is a co-founder of the TELECOM ParisTech spin-off Secure-IC.

Laurent Sauvage received his M.S. in electronics, electrotechnique and cybernetics in 1998 and the "agrégation" (french national competitive exam for high school teachers) of electrical engineering in 2002. He is currently pursuing a PhD in practical side-channel attacks (mainly DPA & EMA). He is responsible for the experimental aspects linked to the physical cryptoanalysis platform of TELECOM ParisTech. Laurent Sauvage is a co-founder of the TELECOM ParisTech spin-off Secure-IC.

Florent Flament received his M.S. from the École Nationale Supérieure des Télécommunications (ENST) in 2005. From 2005 to 2007, he designed cryptographic ASICs (amongst others the 'SecMat' family) within the VLSI group at the ENST. From 2007 to 2008, he was a system engineer in a Hewlett-Packard team, working on the SFR mobile telecommunication system supervision. From 2008 onwards, he has joined again the TELECOM ParisTech VLSI group to work on the security evaluation of embedded cryptoprocessors.



Vinh-Nga Vong received her M.S. from the École Nationale Supérieure des Télécommunications (ENST) in 2005. From 2005 to 2007, she worked on several projects in the telecom (Nortel Networks) and electronics (Thales) industry. From 2007 to 2008, she joined the TELECOM ParisTech VLSI group to contribute to the experimental side-channel analysis platform and to the characterization of design time security metrics. From 2008 onwards, she is with Airbus, at Toulouse, France, working on avionics embedded software.

Philippe Hoogvorst graduated from École Normale Supérieure, Paris. He got his PhD in 1974 for a study on languages without assignments. He was one of the creators of the "Laboratoire d'Informatique Expérimentale of the École Normale Supérieure". He defended a "thèse d'État" in 1983 on the same subject as the PhD. Philippe Hoogvorst is currently researcher at the CNRS and detached to the LTCI/UMR 5141. He is working on innovative ways to attacks on electronic circuits; more specifically, he devises signal and information processing techniques ranging from correlation to template attacks.

Renaud Pacalet received his M.S. from the ENST in 1988. From 1993 to 1995 his worked on various industrial projects as a research engineer at TELECOM ParisTech. From 1996 to 2003 he was responsible for the Integrated Systems group at TELECOM ParisTech. From 2003 on, he created and now leads the Systems-on-Chip laboratory of TELECOM ParisTech at Sophia-Antipolis. His research interests are the flexible architectures for the software defined radio; the methods and tools for the specification, design and validation of integrated systems; the security of embedded systems (shielding against side-channel attacks, privacy and integrity of memory buses, formal proof of critical embedded software).