Countering Early Evaluation: An Approach Towards Robust Dual-Rail Precharge Logic

Shivam Bhasin, Sylvain Guilley, Florent Flament, Nidhal Selmane, Jean-Luc Danger
Institut TELECOM / TELECOM ParisTech, CNRS LTCI (UMR 5141)
Departement COMELEC, 46 rue Barrault,
75634 PARIS Cedex 13, FRANCE.
firstname.lastname@telecom-paristech.fr

ABSTRACT

Wave Dynamic Differential Logic (WDDL) is a hiding countermeasure to thrawt side channel attacks (SCA). It suffers from a vulnerability called Early Evaluation, **i.e.** calculating output before all inputs are valid. This causes delay biases in WDDL even when synthesized with positive gates. s a consequence, the design can be attacked, although with extra effort, through side channel. However, WDDL is an appealing logic since it has already been reported to natively resist against multiple asymmetric faults. In this article, we suggest a Dual Rail Precharge Logic (DPL), similar to WDDL, free from early evaluation by design. We demonstrate practically that the early evaluation accounts for major part of the leakage. We also provide basic guidelines for designing such a DPL. This DPL can resist against side channel attacks and fault attacks at the same time. In line with the current security evaluation methodology, we use differential power analysis and mutual information to compare the modified WDDL with the traditional WDDL. To compare robustness w.r.t security, we conduct a proof-of-concept experiment that compares the two logics with identical implementations (P&R) apart from the logic style. The sensitive side channel leakage is reduced by half in the DPL without the early evaluation flaw.

General Terms

Design, Security

Keywords

Side-channel attack, WDDL, DPL, early evaluation, mutual information metric.

Categories and Subject Descriptors

B.7.1 [**Logic Design**]: Types and Design Styles- Secure Logic for FPGA; C.3 [**Special-purpose and Application-based Systems**]: Real-time and Embedded Systems; E.3 [**Data Encryption**]: Advanced Encryption Standard (AES); Side Channel Attacks

WESS'10, October 24, 2010, Scottsdale, AZ,USA

1. INTRODUCTION

Cryptographic modules are often used in complex commercial systems to encrypt sensitive data. These cryptographic modules consume a lot of resources as they involve complex mathematical operations needed to prevent illegitimate users from spying, impersonating or altering communication. As modern cryptographic algorithms are robust against mathematical cryptanalysis, logical security provided by cryptography can be jeopardized by observing or perturbing the operations on the physical layer. Attacks based on observing the cryptographic device are known as side-channel attacks (SCAs, such as the DPA [11]) and the attacks based on perturbation are known as fault injection attacks (FIA, such as the DFA [3, 17]).

SCAs rely on the physical leakage of the device to gain information about its internal data. This leakage could be observed in various forms like power consumption, electromagnetic radiation, timing, sound, *etc.* On the one hand, such observation can be done without or with very few changes in the device environment. On the other hand, DFAs work when a device produces faulty outputs due to altered state of the device, that together with nominal outputs, can disclose relationships within the secret bits normally concealed into the hardware. Thus SCA's obvious advantage is its furtivity as it is hard to detect. Since they are virtually impossible to detect, an adequate countermeasure must be vigilant each time the cryptographic engine is in use. SCAs are even more effective in FPGAs as the routing is automatic and thus imbalanced. Since FPGAs are being widely deployed, it is important to secure the implementations on FPGAs.

On the contrary, the first prerequisite for a DFA to be successful is to actually modify the device's state. A detection strategy can thus be enforced to check for the device operations' integrity. Hence a device can be claimed to be tamper-resistant only if it is protected, at least to some extent, against both SCA and DFA simultaneously.

Generally countermeasures against SCAs and DFAs are implemented differently. DFA countermeasures act at the algorithmic level, usually introducing space or time redundancy in data representation and processing. However, the effective protection against SCA is more subtle. There is a need for removal of any source of leakage which could provide some sensitive information through physical side-channels. Therefore, a widespread methodology is to use balanced logic gates along with *ad hoc* backend steps. As we know how to resist against DFA before the logic synthesis and to resist against SCA after synthesis, it is implicitly con-

sidered obvious that the protection against DFA and SCA should be built one on top of the other.

Dual-rail precharge logic (DPL) style is a static countermeasure being widely used against SCAs. The basic idea behind DPL is to make the activity of a device constant at all times thus revealing no information in the side channel leakage. Wave dynamic differential logic (WDDL [26]) is a kind of DPL style which is compact and separable into true & false parts. Due to its data hiding property, WDDL provides evident resistance against SCAs which makes it well suited for FPGA based designs. In [23, 24], authors introduce a vulnerability in WDDL known as "early evaluation" (EE). It is because of imbalance caused by early evaluation that SCAs are possible against WDDL, although with great difficulty, as demonstrated in [25, 10, 20, 15]. In [21], authors prove that WDDL is natively protected against multiple asymmetric faults, typically caused by global perturbations. This means WDDL can resist, to some extent, both observation and perturbation attacks. Therefore, it is likely that if WDDL can be made free from early evaluation, it can be safely used as a common countermeasure against SCAs and DFAs. In ASICs the designer has full control over the design and thus its easier to secure the design. However on FPGAs the designer has limited control over the routing of the design and thus needs extra logic to secure the design. We propose DPL-noEE as the logic to secure FPGA based designs. Nevertheless, minor optimization of this logic can make it applicable to ASICs as well.

In another work [7], it is shown that WDDL is weaker than another EE-free logic (namely SecLib [8]). However, in these papers, it is hard to tell WDDL is less secure than EE-free styles because of EE or because of implementation differences. In this article, we present a method to implement a variant of WDDL which is free from early evaluation. We call it as DPL without Early Evaluation (DPL-noEE) in the text that follows. DPL-noEE is inspired from BCDL, another DPL with synchronization (to remove EE). In BCDL, there is a global signal to acheive synchronization where as in DPL-noEE it is in built into the truth tables. The rationale for DPL-noEE is detailed in section 2.3. We then compare this logic with a DPL logic similar in construction but prone to early evaluation. Thus keeping everything else the same we can compare the influence of early evaluation on a DPL logic. We call the DPL suffering from early evaluation flaw is as DPL with Early Evaluation (DPL-EE).

Another traditional countermeasure consists of the introduction of randomness in the computation. However, such a strategy has many vulnerabilities:

- exploitable non-uniformity [5] in multiplicative masking in the AES [1];
- glitches occurring on transiently unmasked combinational internal signals [13];
- trivial mask removal [16] in MDPL [18].

We realize a proof-of-concept experiment where targeted logics are implemented in a similar manner. The two implementations differ only in the logic gate description, having exactly the same backend as optimised by the tool. In FPGA, our chosen technology target, the configuration files remain the same, but for the LUT masks. Thus, by subsuming the individual issues of robustness against SCA and DFA into a unique problem, we arrive at an original solution that is economic in resources because of its duality w.r.t. both the SCA and the DFA threats. The main objective of this is study is to demonstrate the effect of early evaluation, which to our knowledge, has only been studied theoretically before.

The rest of the article is organized as follows. Section 2 presents a detailed description of DPL style with and without EE. In section 3, we describe the implementation details of the two countermeasures on an AES coprocessor. Section 4 details the practical evaluation of these countermeasures using differential power analysis and mutual information metric. Finally, conclusions are drawn in section 5.

2. RATIONALE OF THE PROPOSED LOGIC

2.1 DPL with Early Evaluation

DPL uses true and false representations of each signal (I/O of each cell). For a signal a, couple $(a_\mathrm{T}, a_\mathrm{F})$ alternates between two values:

1. $(0,0)$ or $(1,1)$, called NULL0 or NULL1, and designated as a NULL token, playing the role of spacer, and
2. $(1,0)$ or $(0,1)$, called VALID0 or VALID1, and designated as a VALID token, carrying the value of a.

To be secure against SCAs, DPL gate should compute alternately NULL and VALID tokens, with the remarkable property that only one bit toggles with every transition. A pair of gates $(f_\mathrm{F}, f_\mathrm{T})$ respects this convention if:

- It propagates the NULL values, *i.e.*, if all the inputs are NULL, then $(f_\mathrm{F}, f_\mathrm{T})$ is also NULL.
- It propagates the VALID values, *i.e.*, if all the inputs are VALID, then $(f_\mathrm{F}, f_\mathrm{T})$ is also VALID.

Table 1: Look-up-Table (LuT) masks encoding for 4-input LuTs implementing the AND function in DPL with early evaluation

DPL-EE				AND_T	AND_F
a_T	a_F	b_T	b_F	CCOO	FAFA
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	1	1

Few DPLs like WDDL have a remarkable property to be separable, meaning that f_F (*resp.* f_T) depends only on the false (*resp.* the true) input's half. In a WDDL cell [26],

one transition per cycle is observed, which is favourable for a DPA resistant logic style, provided only positive logic is used (i.e only AND & OR gates). Figure 1 shows the timing diagram of WDDL AND gate with NULL (precharge phase) and VALID (evaluation phase) tokens in alternate cycles.

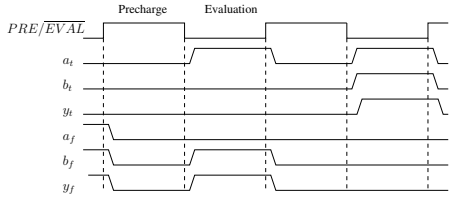

Figure 1: Timing diagram for a WDDL AND gate.

As shown in figure 2, a WDDL AND gate consists of an AND gate (G) and a complementary OR gate (G^*, satisfying $G^*(x) \doteq \overline{G(\overline{x})}$). For sequential circuits, each flip-flop is replaced by two pairs of flip-flop. These four flip-flops allow the precharge wave to propagate through the whole design as all the gates are positive. In addition, this "trivial" architecture is definitely resistant to the attack on the flip-flips (FF) presented in [14] when DPL FFs are shared. It has to be noted that inverters in WDDL are implemented by crossing the true and false signals of the same variable. In [21], authors provide a detailed account of CAD flow to convert a single-rail design in WDDL.

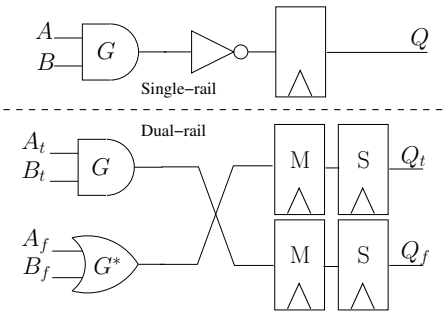

Figure 2: WDDL building block.

However, several weaknesses have been noticed in WDDL since it has been introduced. Firstly, when the circuit switches between two phases, on the way from all NULL to all VALID values, glitches can occur if the functions ($f_{\mathrm{F}}, f_{\mathrm{T}}$) are not positive [10]. Afterwards, many authors notice concomitantly that the evaluation time depends on the inputs values [12, 19]. This problem was called as "early propogation effect" or popularly as "early evaluation". We design our DPL-EE on the same grounds as WDDL i.e. it is separable and propagates NULL & VALID tokens in alternate cycles. Its truth table is shown in table 1. The encoding mask for an AND is (`CC00, FAFA`). In an FPGA, look-up table (LUTs) are the basic elements. LUTs are ROMs where inputs serve as addresses. So for a 4-input LUT we have 16 addresses and the values at these addresses can be determined by a encoding mask. Most FPGAs LUT have single output. So the 4-bit input calculates the 1-bit output depending on the mask.

2.2 Early Evaluation

When a DPL-EE gate enters evaluation phase, input signals which were forced to '0' acquire their original value. At this point one of the two complementary gates evaluates to '1'. Even though the gates are balanced, valid input signals can arrive at different times due to difference in logical path. Since the transition probability of the gate is unity, the gate will evaluate without waiting for all the signal to be valid. This causes skew in the circuit further leading to data-dependant leakage on the power traces. This phenomena is known as early evaluation [23, 12]. Now let us closely observe table 1. The table shows that for some inputs (a_T, a_F, b_T, b_F) values like (0,0,0,1), (0,0,1,1) , (0,1,0,0) the AND_F gate evaluate to '1'. For these values AND_F has evaluated despite the fact that either of a or b is not VALID. This is indeed logical as an OR gate outputs '1' when at least one of the inputs is '1'. Thus the DPL AND gate evaluates early before a and b have acquired a VALID state. Such phenomena can result in a data dependant leakage.

Another problem is the balancing within the pair, not the various paths. The imbalance is caused by automatic place and route. In FPGAs, the delay between the inputs of a gates is increased when the tool places different logic optimally. Recently some methods have been suggested to remove the bias due to place and route [27, 9]. Applying these techniques, the designer can get rid of the bias caused by place and route, but the bias caused by early evaluation exists. If this bias is removed from DPL, it may emerge as a strong countermeasure against SCAs and DFAs. In the next section, we propose a method to remove the early evaluation effect from a DPL cell.

2.3 DPL without Early Evaluation

As stated earlier, if DPL is made early evaluation free then it can be used as a common countermeasure against both SCAs and DFAs. To cure DPL of early evaluation, we implement the truth tables of DPL logic such that it propagates a VALID output only if all the inputs are VALID. This behavior can be achieved by a purely combinatorial gate, as depicted in Tab. 2 for an AND gate where the encoding mask of an AND gate is changed from (`CC00, FAFA`) to (`FC80, FAE0`). Similarly for a DPL OR (`T, F`) gate the encoding mask is changed from (`FFCC, A0A0`) to (`FEC0, F8A0`). This implies that an AND_T gate which was previously implemented by an encoding mask `CC00` (see table 1) will now be implemented with an encoding mask `FC80`(see table 2). By changing the encoding mask we ensure the overall 4-input gate is positive.

By changing the encoding mask we ensure the following:

- The gate outputs NULL when the inputs are NULL or transitional from this value.
- The gate outputs VALID only when all the inputs are VALID.
- In case of inconsistent values w.r.t. DPL convention, the gate outputs an arbitrary NULL value.
- The overall 4-input gate is positive.

This logic does not evaluate early by design, and propagates errors: if any input is stuck to NULL or if the input is out of specifications, then the output always remain to NULL too. An advantage of DPL-noEE over WDDL is that

Table 2: Look-up-Table (LuT) masks encoding for 4-input LuTs implementing the AND function in DPL without early evaluation

DPL-noEE				AND_T		AND_F		Input state in the DPL protocol
a_T	a_F	b_T	b_F	FC80		FAE0		
0	0	0	0	0		0		All NULL0
0	0	0	1	0	0	0	0	Transitional from NULL0
0	0	1	0	0		0		Transitional from NULL0
0	0	1	1	0		0		Faulty
0	1	0	0	0		0		Transitional from NULL0
0	1	0	1	0	8	1	E	All VALID: $(a, b) = (0, 0)$
0	1	1	0	0		1		All VALID: $(a, b) = (0, 1)$
0	1	1	1	1		1		Transitional from NULL1
1	0	0	0	0		0		Transitional from NULL0
1	0	0	1	0	C	1	A	All VALID: $(a, b) = (1, 0)$
1	0	1	0	1		0		All VALID: $(a, b) = (1, 1)$
1	0	1	1	1		1		Transitional from NULL1
1	1	0	0	1		1		Faulty
1	1	0	1	1	F	1	F	Transitional from NULL1
1	1	1	0	1		1		Transitional from NULL1
1	1	1	1	1		1		All NULL1

all kind of logic can be used instead of only positive logic since output is calculated if all inputs are VALID and **does not generate glitches**.

Table 3: LuT4 masks for the $2 \rightarrow 1$ function names in DPL w/ and w/o EE styles.

$f(a,b)$, when $(a,b)=$				DPL-EE		DPL-noEE		Function
(1, 1)	(1, 0)	(0, 1)	(0, 0)	True	False	True	False	Name
0	0	0	0	0000	FFFF	F880	FEE0	0
0	0	0	1	A0A0	FFCC	F8A0	FEC0	$\overline{a} \cdot \overline{b}$
0	0	1	0	C0C0	FFAA	F8C0	FEA0	$\overline{a} \cdot b$
0	0	1	1	F0F0	FF00	F8E0	FE80	$\overline{a}$
0	1	0	0	AA00	FCFC	FA80	FCE0	$a \cdot \overline{b}$
0	1	0	1	AAAA	CCCC	FAA0	FCC0	$\overline{b}$
0	1	1	0	EAC0	FCA8	FAC0	FCA0	$a \oplus b$
0	1	1	1	FAFA	CC00	FAE0	FC80	$\overline{a} + \overline{b}$
1	0	0	0	CC00	FAFA	FC80	FAE0	$a \cdot b$
1	0	0	1	ECA0	FAC8	FCA0	FAC0	$\overline{a \oplus b}$
1	0	1	0	CCCC	AAAA	FCC0	FAA0	b
1	0	1	1	FCFC	AA00	FCE0	FA80	$\overline{a} + b$
1	1	0	0	FF00	F0F0	FE80	F8E0	a
1	1	0	1	FFAA	C0C0	FEA0	F8C0	$a + \overline{b}$
1	1	1	0	FFCC	A0A0	FEC0	F8A0	$a + b$
1	1	1	1	FFFF	0000	FEE0	F880	1

For a complex circuit, we need a set of various logic functions in order to achieve an optimised synthesis. Tab. 3, provides the encoding mask for all non-trivial two input functions synthesized on 4-input LUTs. The table list encoding masks when a particular function is implemented in DPL-EE and its equivalent when the same function is implemented in DPL-noEE. The rows printed in gray correspond either to trivial functions (those that depend on only one input variable), that can be implemented as routing in DPL. The constant functions 0 and 1 are also ruled out because they do not propagate the precharge, and because they are simplified out by the synthesizer anyway: they should not exist in the same netlist. These logics as presented are specific to FPGAs directly targeting the 4 input LUTs rather than basic gates. In this article we focus on implementation and evaluation of this logic on FPGAs only.

3. IMPLEMENTATION

Figure 3 shows the architecture of a simple, non protected AES co-processor. The AES co-processor is designed to have a parallel architecture. It performs a round of AES in each clock cycle. The four sub-rounds are SubBytes, ShiftRows, MixColumns and AddRoundKey. These sub-rounds along with some multiplexers and key scheduler comprise the datapath. The key scheduler or expander calculates a key for each round which is then used in the datapath. The SubBytes & Key Schedule use 16 & 4 Sboxes respectively.

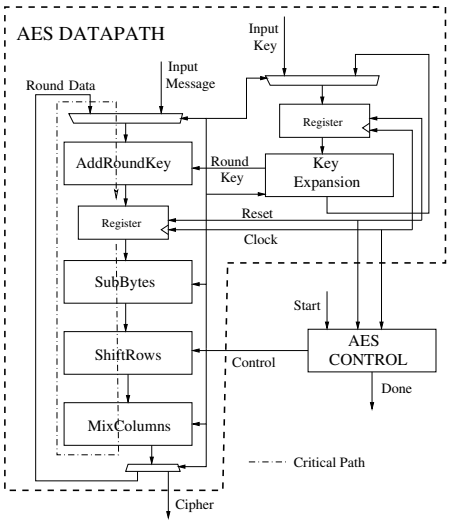

Figure 3: AES architecture.

We implement an AES coprocessor in the two DPL styles for evaluation. We start with a RTL model of single-rail AES coprocessor synthesized using QUARTUS for EP1S25B672C7 device from Altera Stratix FPGAs as a reference design. For DPL logic, this coprocessor is synthesized using a library of all non-trivial two input logic functions. An ASIC synthesizer is used as it can accept custom libraries unlike FPGA synthesizers. The output netlist is then treated by our custom tool VDUPLICATE to convert single-rail netlist into dual-rail netlist, using the LUT masks abiding with the DPL-noEE logic. The DPL design is then connected with the peripherals and synthesized using QUARTUS. Once the synthesis is complete, the routing constraint file (rcf) is extracted from the design. To obtain a design using DPL-EE logic as in 2.1 , we change the LUT masks in the previously obtained dual-rail netlist as in table 3. Again this netlist is connected to peripherals and synthesized using QUARTUS. The routing is constrained by the previously extracted **rcf**. We are able to conserve 99.96% of the routing from one DPL logic to other. Authors would like to specify that no effort has been put in placing and routing the design. The placing and routing is automatically done by the FPGA vendor's tool. Considering the cost of the design, it is evident that a design in original WDDL will be smaller than a design in DPL-noEE. However, in our implementations we force the DPL-EE design which is also based on the principles of WDDL to be placed and routed exactly as the DPL-noEE design. We agree that this is not an optimal implementation for DPL-EE but can be interesting for observing the effect of early evaluation. Thus the AES which we implement on

Table 4: DPA results on SBOX 0 for two DPL variants of AES.

Implementation	No. of Traces							
SBOX BIT index	0	1	2	3	4	5	6	7
DPL-EE	8314	X[1]	1340	X	7000	X	X	X
DPL-noEE	X	X	1150	X	X	X	X	X

Altera Stratix EP1S25B672C7 consumes 14,574 logic elements. In total, the design containing the AES and other peripherals consumes 16,451 logic elements i.e. 64% of the FPGA fabric. The maximal operating frequency achieved is 19.70 MHz. Finally we have two identical netlist, implementing the same circuit and having the same placement and routing, differing only in LUT masks. Thus one netlist has LUT masks which suffer from early evaluation while the other netlist resists. Such a setup will allow us to demonstrate the effect of early evaluation.

4. EVALUATION OF THE PROPOSED COUNTERMEASURE

4.1 Using Differential Power Analysis

We analysed the two DPL variants against differential power analysis. We took power consumption measurements (*traces*), using an *electromagnetic probe* capturing the field of a leaking capacitor on the back-side of the FPGA core with a *54855 Infiniium oscilloscope* from Agilent Technologies. In order to reduce acquisition noise, each trace was averaged 64 times. We performed a mono-bit DPA on the first SBOX of the two WDDL implementations. We could only perform a mono-bit DPA using the hamming weight model. The power consumption difference can be exploited on one bit but cannot be added for multiple bits as in correlation power analysis on unprotected circuits; indeed, the residual biases between activity of (0,1) and (1,0) is not consistent from bit to bit; it is not straightforward which method to combine them would be suitable.

The performed attack was partially successful. The results are shown in table 4. We were able to find the right key of the first SBOX for three bits in DPL-EE implementation and only one bit in DPL-noEE. Given this data, it can be said that DPL-noEE implementation provides higher robustness than DPL-EE. Thus keeping everything constant, countering early evaluation has improved the robustness of the design. In the next section, we try to measure the side channel leakage due to early evaluation, using mutual information metric.

4.2 Using Mutual Information

The Mutual Information Metric [4] can be used as a metric to evaluate the amount of information leaked by some observations [22]. The paper [28] (and more precisely its third contribution) indeed explains that when two circuits are evaluated with a relevant partitioning, the MIA is indeed suitable to compare them on a fair basis. The principle idea of this technique is to compute the mutual information $I(S;O)$ between two random variables S and O, where S represents set of sensitive variables used by the target device and O are the observations measured during computations of such device. If the value of $I(S;O)$ is high and depends on a small part of a secret key k^0 which does not change between consecutive encryptions, we should theoretically be able to guess k^0 with few observations. The correct hypothesis on the key corresponds to that with largest value of $I(S;O)$. Similarly, with a near zero mutual information value, it should be almost impossible to tell the secret k^0 from the other candidates $k \neq k^0$ by analyzing the observations O. In such way, this mutual information value is a way to quantify the intrinsic robustness of a cryptographic device.

The main advantage of mutual information metric lies in the value of $I(S;O)$ which can be used to compare different implementations without internal knowledge of the implementation. For two similar implementations, the one leaking more information from the side channel during a particular time instant will generate a higher $I(S;O)$, computed knowing the correct key $k = k^0$ than the one leaking less. We use this technique to compare the two DPL variants of AES (DPL-EE & DPL-noEE). Incidentally, DPL was also taken as an example where mutual information performs well in §6.3 of the MIA original paper [4].

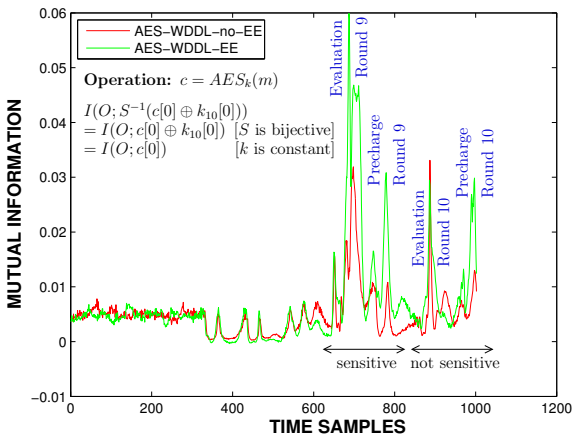

Figure 4: Comparison of Mutual Information leaked from an AES protected using DPL-EE and DPL-noEE.

We correlate the traces, used for DPA, with the output of the first SBOX (one byte) and calculate the mutual information knowing the right key. As shown in figure 4, the mutual information for a DPL-EE is almost twice as that for DPL-noEE. Since everything else is kept unchanged, the reduction in leakage is caused by early evaluation. The remainder leakage could be due to technological bias (routing imbalance of the complementary logic) which is common to the two implementations. Indeed a detailed analysis of the floorplan revealed that the leaking bits are routed using long routing channels. On the other hand, non-leaking bits are connected to adjacent LUTs thus avoiding routing channels. Therefore it can be said that imbalanced routing introduce delays in the circuit exploitable by attackers. Controlling

[1]In this table, X signifies failure of the mounted attack after 40 000 traces.

routing of a design, to our knowledge, is difficult because the routing algorithms are kept secret by the FPGA vendors. Nevertheless, some options to control routing are available. It would be interesting to see what level of balancing could be acheived using the given routing options.

As already discussed, in DPL, every combinational block receives input and stores output by a pair of registers in master-slave mode. If we consider the circuit at various stages, the connection between a master and a slave is pretty much balanced as very few wires are used. Similarly, for the connection between a combinational output and master register input, the routing is balanced. For the connection between slave register and input of the combinational part, the fan out is high. It is at this point (denoted Q_f and Q_f in Fig. 2) that the circuit suffers from high imbalance in the routing of the two circuits. In our analysis, the two circuits are routed almost identically. [1] Assuming equal leakage due to routing from both circuits, countering the early evaluation has alone reduced the leakage by half. Thus early evaluation is a major flaw in dual-rail logic and countering it could considerably reduce the leakage.

In figure 4, we see that the leakage is occurring over a period of 4 clock cycles. We correlate the traces with the value of the secret in penultimate round. In DPL, each round is composed of two clock cycles: precharge and evaluation. For penultimate round and the ultimate round the data stays in the circuit for two cycles per round. Since, the last round does not have MixColumns and the SubBytes is bijective, therefore a byte in round 9 is correlated to byte in round 10 located at the corresponding spot. This means that if we observe correlation with a byte in round 9, we should also observe the correlation in round 10. Hence we see 4 peaks for precharge and evaluation cycles in round 9 and round 10, in figure 4, for each of the DPL-EE and DPL-noEE.

We wish to illustrate that the leakage put forward by the MIM indeed opens the door to successful key recovery attacks, and that the attack is easier if the MIM indicator is high; this means that the MIM value, for any characterization (*e.g.* target bit in a sensitive AES state byte) already indicates the speed of the attack: roughly speaking, the speed of the attack is related to the MIM value. In corollary, the different values of the MIM give a precious feedback information to the designer: the bits that yield the largest MIM should indeed be corrected with priority in the design.

We also aim to test the real vulnerability of the flaws identified in the two designs. To reach this goal, we compare two attacks: DPA and MIA. The DPA is known to be appropriate if the leakage model is correct, whereas the MIA is both matching the MIM analysis (both are based on PDF estimation) and more resilient to leakage model imperfections.

Comparing table 4 with figure 5,6, we see that the bits broken using DPA are the ones leaking in the MIM. Although this observation seems intuitive and justifies the relevance of robustness metrics, such as the MIM, we emphasize that, to our best knowledge, it is the first time a relationship between the amount of leakage and the speed of an attack is put forward experimentally. Roughly, lesser the MIM value of a leaking bit, higher is the number of traces needed for a successful DPA. On the other hand, bits not leaking in MIM are also not broken by DPA. Also in figure 6, bit 2 is leaking much more than the other 7 bits which are leaking almost nothing. A detailed analysis of the post synthesis netlist showed that the synthesizer didn't respected the DPL-noEE encoding mask for this bit during the optimisation process. However the other bits, which we checked were compliant with the DPL-noEE logic. Thus the unexpected leakage observed is due to some unwanted optimization by the tool and imbalanced routing. From a designer point of view, such information is very important as the designer can cover the loop holes once known.

Our conclusion is that side-channel techniques based on information theory (*e.g.* MIM) and on statistics (*e.g.* MIA) can complement well. In our case study, MIM is relevant to identify the most leaky resources, whereas the DPA is the most efficient attack once the largest leakage is identified.

[1] More precisely, the routing obtained automatically for AES DPL-EE has been saved (back-annotated) and reused as a constraint for the routing of AES DPL-noEE. Quartus reports that 99.96% of constraints has been respected.

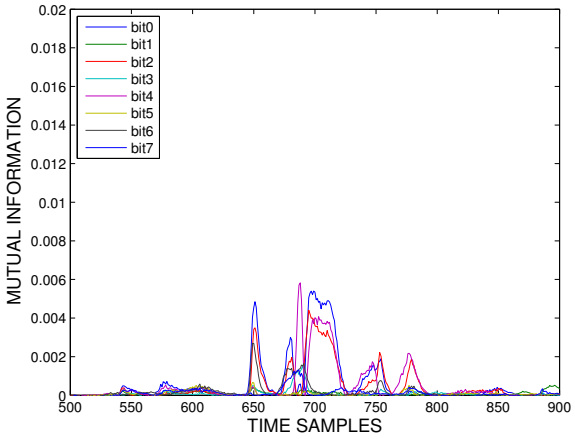

Figure 5: Bitwise leakage of $SBOX0$ in DPL-EE.

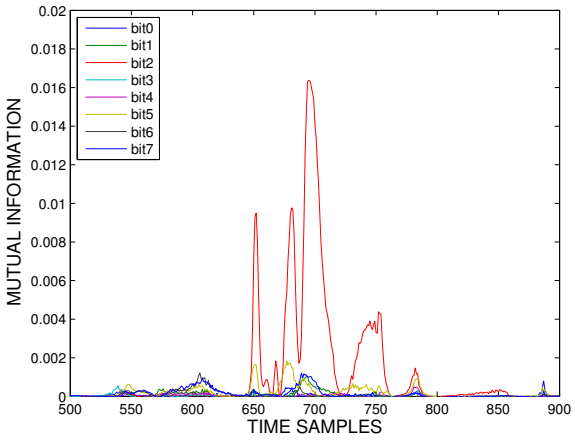

Figure 6: Bitwise leakage of $SBOX0$ in DPL-noEE.

4.3 Protection against faults

In [21], N.Selmane et al. have demonstrated that WDDL is immune against multiple asymmetric faults such as those

caused by setup violations. Basically, the idea is that asymmetric faults are able to turn any VALID token into a NULL value. For instance, the fault can induce a mutation from any VALID to the NULL spacer. The NULL token can propagate until the outputs, being even amplified. However, the NULL wave propagation acts as an eraser, which means that the outputs have eventually lost any information about the faulty values.

In [2] S.Guilley et al. have shown that results obtained in [21] can be extended to all DPLs. Similarly, DPL-noEE will not disclose the faulty result in the presence of a setup violation. Instead, it will propagate the NULL on the fault fanout, even if a VALID value could have been deduced. This is the logic behavior of `'X'` in VHDL. We would like to mention that such construction make the logic resistant to asymmetrical faults. Symmetrical faults, which are very difficult to realize, are still a security threat. Such faults can be resisted using coding techniques. A detailed description of DPL-noEE resistance against faults can be found in [2]

5. CONCLUSION

In this article, we demonstrate the effect of early evaluation which, to our knowledge, was only discussed theoretically before. Using mutual information metric we were able to infer that only removing early evaluation and keeping everything else constant reduces the leakage to half. This could have been more if we had more control over the FPGA synthesis and *P&R*. When using mono-bit DPA on first SBOX, attacks on 3 bits were successful for a design implemented with DPL-EE, as compared to 1 for DPL-noEE. Along with this analysis, we also propose a basic technique to remove early evaluation from DPL. The technique not only removes the vulnerability but also allows designers to use non-positive gates for DPL synthesis, which can enable better optimization. We would like to mention that these are just the basic guidelines for designing a robust countermeasure and not the final product.

We would also like to design a proper countermeasure based on these guidelines and achieve the optimised synthesis, as part of future projects. This will also include more controlled synthesis avoiding such violations. Another task would be to reduce the leakage due to technological bias and imbalanced routing [27, 9, 6] with none or little compromise with the leakage by early evaluation.

Acknowledgments

This work has been supported by the french national research agency (ANR), through the `ANR-07-ARFU-010` grant "SeFPGA" (Secured Embedded FPGAs). We acknowledge interesting discussions and encouragements with Renaud Pacalet from the LabSoC laboratory of TELECOM ParisTech at Sophia-Antipolis. We also like to thank the anonymous reviewers for their valuable comments.

6. REFERENCES

[1] M.-L. Akkar and C. Giraud. An Implementation of DES and AES Secure against Some Attacks. In LNCS, editor, *Proceedings of CHES'01*, volume 2162 of *LNCS*, pages 309–318. Springer, May 2001. Paris, France.

[2] S. Bhasin, J.-L. Danger, F. Flament, T. Graba, S. Guilley, Y. Mathieu, M. Nassar, L. Sauvage, and N. Selmane. Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow. In *ReConFig*, pages 213–218. IEEE Computer Society, December 9–11 2009. Cancún, Quintana Roo, México, DOI: 10.1109/ReConFig.2009.50, `http://hal.archives-ouvertes.fr/hal-00411843/en/`.

[3] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *CRYPTO*, volume 1294 of *LNCS*, pages 513–525. Springer, August 1997. Santa Barbara, California, USA. DOI: 10.1007/BFb0052259.

[4] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *CHES, 10th International Workshop*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA.

[5] J. D. Golic and C. Tymen. Multiplicative Masking and Power Analysis of AES. In *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 198–212. Springer, August 13-15 2002. San Francisco, USA.

[6] 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*, pages 29–35. IEEE Computer Society, 2008. June 9, Anaheim, USA. ISBN = 978-1-4244-2401-6.

[7] 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*, 57(11):1482–1497, nov 2008.

[8] S. Guilley, F. Flament, Y. Mathieu, and R. Pacalet. Security Evaluation of a Balanced Quasi-Delay Insensitive Library. In *DCIS*, Grenoble, France, nov 2008. IEEE. Session 5D – Reliable and Secure Architectures, ISBN: 978-2-84813-124-5.

[9] S. Guilley, P. Hoogvorst, Y. Mathieu, and R. Pacalet. The "Backend Duplication" Method. In *CHES*, volume 3659 of *LNCS*, pages 383–397. Springer, 2005. August 29th – September 1st, Edinburgh, Scotland, UK.

[10] S. Guilley, L. Sauvage, J.-L. Danger, T. Graba, and Y. Mathieu. Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs. In *SSIRI*, pages 16–23, Yokohama, Japan, jul 2008. IEEE Computer Society. DOI: 10.1109/SSIRI.2008.31, `http://hal.archives-ouvertes.fr/hal-00259153/en/`.

[11] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.

[12] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin. Power Attacks on Secure Hardware Based on Early Propagation of Data. In *IOLTS*, pages 131–138. IEEE Computer Society, 2006. Como, Italy.

[13] S. Mangard, N. Pramstaller, and E. Oswald. Successfully Attacking Masked AES Hardware Implementations. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 157–171. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.

[14] A. Moradi, M. Salmasizadeh, and M. T. M. Shalmani. Power analysis attacks on mdpl and drsl implementations. In *ICISC*, volume 4817 of *Lecture Notes in Computer Science*, pages 259–272. Springer,

November 29-30 2007.

[15] E. D. Mulder, B. Gierlichs, B. Preneel, and I. Verbauwhede. Practical DPA Attacks on MDPL. In *First International Workshop on Information Forensics and Security (WIFS)*. IEEE Signal Processing Society, December 6-9 2009. London, United Kingdom. Also `http://eprint.iacr.org/2009/231`.

[16] E. D. Mulder, B. Gierlichs, B. Preneel, and I. Verbauwhede. Practical DPA Attacks on MDPL. Cryptology ePrint Archive, Report 2009/231, 2009. `http://eprint.iacr.org/`.

[17] G. Piret and J.-J. Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In *CHES*, volume 2779 of *LNCS*, pages 77–88. Springer, September 2003. Cologne, Germany.

[18] T. Popp and S. Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 172–186. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.

[19] M. Saeki and D. Suzuki. Security Evaluations of MRSL and DRSL Considering Signal Delays. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(1):176–183, 2008. DOI: 10.1093/ietfec/e91-a.1.176.

[20] L. Sauvage, S. Guilley, J.-L. Danger, Y. Mathieu, and M. Nassar. Successful Attack on an FPGA-based WDDL DES Cryptoprocessor Without Place and Route Constraints. In *DATE*, pages 640–645, Nice, France, apr 2009. IEEE Computer Society.

[21] N. Selmane, S. Bhasin, S. Guilley, T. Graba, and J.-L. Danger. WDDL is Protected Against Setup Time Violation Attacks. In *FDTC*, pages 73–83. IEEE Computer Society, September 6th 2009. In conjunction with CHES'09, Lausanne, Switzerland. DOI: 10.1109/FDTC.2009.40; Online version: `http://hal.archives-ouvertes.fr/hal-00410135/en/`.

[22] F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.

[23] D. Suzuki and M. Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In *CHES*, volume 4249 of *LNCS*, pages 255–269. Springer, 2006. Yokohama, Japan. `http://dx.doi.org/10.1007/11894063_21`.

[24] D. Suzuki and M. Saeki. An Analysis of Leakage Factors for Dual-Rail Pre-Charge Logic Style. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(1):184–192, 2008. DOI: 10.1093/ietfec/e91-a.1.184.

[25] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede. Prototype IC with WDDL and Differential Routing – DPA Resistance Assessment. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 354–365. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.

[26] K. Tiri and I. Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *DATE'04*, pages 246–251. IEEE Computer Society, February 2004. Paris, France. DOI: 10.1109/DATE.2004.1268856.

[27] K. Tiri and I. Verbauwhede. Place and Route for Secure Standard Cell Design. In Kluwer, editor, *Proceedings of WCC / CARDIS*, pages 143–158, Aug 2004. Toulouse, France.

[28] N. Veyrat-Charvillon and F.-X. Standaert. Mutual Information Analysis: How, When and Why? In *CHES*, volume 5747 of *LNCS*, pages 429–443. Springer, September 6-9 2009. Lausanne, Switzerland.