Research Article

Exploiting Dual-Output Programmable Blocks to Balance Secure Dual-Rail Logics

Laurent Sauvage, Maxime Nassar, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu

Département COMELEC, Institut TELECOM/TELECOM ParisTech, CNRS LTCI (UMR 5141), 46 Rue Barrault, 75 634 Paris Cedex 13, France

Correspondence should be addressed to Laurent Sauvage, laurent.sauvage@telecom-paristech.fr and Sylvain Guilley, sylvain.guilley@telecom-paristech.fr

Received 2 March 2010; Accepted 5 October 2010

Academic Editor: Lionel Torres

Copyright © 2010 Laurent Sauvage et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

FPGA design of side-channel analysis countermeasures using unmasked dual-rail with precharge logic appears to be a great challenge. Indeed, the robustness of such a solution relies on careful differential placement and routing whereas both FPGA layout and FPGA EDA tools are not developed for such purposes. However, assessing the security level which can be achieved with them is an important issue, as it is directly related to the suitability to use commercial FPGA instead of proprietary custom FPGA for this kind of protection. In this article, we experimentally gave evidence that differential placement and routing of an FPGA implementation can be done with a granularity fine enough to improve the security gain. However, so far, this gain turned out to be lower for FPGAs than for ASICs. The solutions demonstrated in this article exploit the dual-output of modern FPGAs to achieve a better balance of dual-rail interconnections. However, we expect that an in-depth analysis of routing resources power consumption could still help reduce the interconnect differential leakage.

1. Introduction

During the last decade, a considerable number of countermeasures have been proposed to protect cryptographic devices against Side Channel Analysis (SCA). They customarily fall in two categories. The first one, masking [1, 2], is very popular in the smart card community as it can be implemented at the algorithm level. Intermediate data processed by a cryptoprocessor are concealed by a value called mask, randomly chosen, which in turn makes power consumption random. In the second category, hiding, intermediate values remain the same as for unprotected implementations, but power consumption is made as constant as possible. According to the state-of-the-art attacks, masked implementation on a Field Programmable Gates Array (FPGA) could be broken with Higher-Order Differential Power Analysis (HODPA) using 12,000 power consumption traces [3] whereas 1,500,000 measurements are not sufficient to disclose the entire secret key of an Application Specific Integrated Circuit (ASIC) cryptoprocessor protected by Wave Dynamic Differential Logic (WDDL) [4], the most popular hiding countermeasure developed by Tiri and Verbauwhede. Proposals for merging both techniques with a view to compensate for weakness of masking against HODPA and for backend operations hardness of hiding have been reported in [5, 6], but this approach unfortunately remains vulnerable when the masking relies on one single bit of entropy [7].

Unmasked dual-rail with Precharge Logics (DPLs) in general seem thus to be a sound solution. As WDDL is based on a standard cell flow, it is the most suited for FPGA implementation. Guidelines for synthesis can be found in [8, 9]. We notice incidentally that those articles target 4 → 1 LUT-based FPGA technologies and thus do not take full advantage of the advanced features of modern FPGAs, such as ALM (Adaptative Logic Modules) configurable blocks and
dual-output logic blocks. DPL robustness against SCA relies on perfectly matching differential routing, which appears to be incredibly hard to achieve for large FPGA designs because of the following.

(i) The only way to evaluate the imbalance between two differential paths is to compute the difference of path delays. But as delays provided by development tools are maximum values, not typical, unbalance evaluation is coarse.

(ii) Routing resources are limited and have thus to be properly shared between each component of the design. This may affect placement, which could no longer simply consist in placing differential components side by side.

(iii) Commercial off-the-shelf (COTS) FPGA layout has been designed to provide flexibility, not differential capability. As a consequence, the performance of DPL is expected to be less efficient than when embedded in ASIC, for which differential components can be placed as close as possible, decreasing the impact of intradie process variability.

To overcome these problems, Yu and Schaumont suggest the Double WDDL (DWDDL) [10] design strategy: from a first direct WDDL module, a complementary clone with identical routing is obtained by duplication, relocation, and logic modification. This way, leakages due to imbalances of each module are expected to compensate one with each other. Unfortunately, some registers of DWDDL never go to precharge value and introduce leakages in the Hamming Distance (HD) model. McEvoy et al. discovered this flaw and propose Isolated WDDL (IWDDL) [11] to solve it. To decrease the fourfold area increase of DWDDL and IWDDL, Baddam and Zwolinski published methods suitable for both ASIC and FPGA. Path Switching [12] balances true and false paths with long high capacitive lines by random swaps. Attractive from a theoretical standpoint, in practice, Path Switching is realized by active elements, which consume power, and thus might diminish robustness against SCA. Divided Backend Duplication [13] consists in splitting dual networks by replacing inverters by exclusive-or (XOR) gates: in precharge phase, the XOR gates are configured as an identity function (a → 0 ⊕ a = a) thus propagating the precharge wave, whereas in evaluation phase, they become functional (a → 1 ⊕ a = 1). Unfortunately, this logic has never proved to be glitch-free, nor of constant activity (because of unwanted spurious glitches). Some logic-level upgrades of WDDL have also been proposed, amongst which iMDPL [14], DRSL [6], STTL [15, 16], SecLib [17], WDDL w/o early evaluation [18], and BCDL [19]. All these styles can be mapped onto an FPGA fabric and are thus concerned with dual-rail balancing.

In this article, we do not search to add logic to balance dual networks but rather finely direct the place-and-route (PAR) tool to avoid area increase. The research is led with a view to assess the security level which can be achieved using commercial FPGAs and conclude on their suitability for balanced DPL. Impact of constrained placement of SBoxes has already been described in [20]. Here, we go several steps further: routing is also constrained and we study and experimentally evaluate a complete cryptoprocessor. This latter is presented in Section 2, along with a thorough characterization of the PAR at design-level. Then, we explain how to best take advantage of dual-output programmable logic blocks. The experimental results we obtain against SCA are provided in Section 3. Finally, conclusions and perspectives are discussed in Section 4.

2. Fitting WDDL 3DES in Stratix II

2.1. WDDL Roots. In DPL, each bit is represented by a couple of signals, and calculations alternate between precharge phase (PRE) and evaluation phase (EVA). Figure 1 is an example of a WDDL AND gate. In PRE, all of the inputs on the left are forced to “0”, which in turn forces the true output s and the false output s′ to “0”. Then, in EVA, after inputs toggle, only s or s′ will be set to “1”. This way, whatever the temporal transition, EVA to PRE or PRE to EVA, and whatever the processed data, only one of the differential outputs commutes, yielding a constant transition count. This protocol is illustrated in Figure 2. But this does not suffice to ensure a constant power consumption: the capacitive load of the True and False networks should be the same. Special care should thus be taken for back-end operations, which is the main purpose of this article.

To avoid glitches, a security flaw of WDDL as they are data dependant, synthesis should use only positive functions. For example, logical exclusive OR (XOR) operator cannot be directly used and will be replaced by a ⊕ b = a ⊙ b + a ⊙ b. This induces cross connections between True and False paths, which increases PAR difficulty.

2.2. WDDL 3DES Cryptoprocessor. The implementation evaluated in this article conforms to the simple and triple Data Encryption Standard (3DES) [21], with all of the specified modes of operations. Although DES has been replaced by the Advanced Encryption Standard (AES) since year 2001, the American National Institute of Standards and Technology (NIST) considers 3DES to be appropriate through year 2030, and the electronic payments industry still uses it for its compactness when implemented in hardware, a quarter of AESs.
The architecture of Figure 3 corresponds to the part of our WDDL 3DES cryptoprocessor which is in charge of processing the 32-bit block \( R \), as defined in page 10 and depicted by page 9 of [21]. This implementation has a straightforward parallel execution, scheduled at one round per clock cycle. The upper and lower paths correspond to the True and False dual networks, respectively. At the beginning of an encryption, the right half of the value resulting of an initial permutation (IP) of the message is loaded into the \( R \) master register \((R_M)\), while the \( R \) slave register \((R_S)\) still holds the zero precharge value. The latter is applied to the combinatorial logic of the DES datapath, comprised of the following:

(i) the expansion permutation \((E)\) function,

(ii) a bit-by-bit addition modulo 2 between the output of \( E \) and the round key \( K_n \), referred in the following as XOR\(_K\),

(iii) Substitution Boxes \((SBoxes)\) nonlinear functions,

(iv) the permutation \((P)\) function,

(v) a bit-by-bit addition modulo 2 between the output of \( P \) and the left part of the LR register \((L)\), referred in the following as XOR\(_L\),

which in turn outputs zero. Due to the feedback, at the next rising edge of the clock \( R_{\neg M} \) will load zero while \( R_S \) will sample the right half of IP, and the combinatorial logic computes the new intermediate value. This alternates until the end of the encryption.

2.3. Place-and-Route Strategies. Some previous articles [8, 9] already described the mapping of dual-rail logics into FPGA. However, all of them target 4 → 1 look-up-table FPGAs. This choice is interesting from the synthesis point of view, since the mapping is very close to that of an ASIC; as discussed in [4], a simple DPL-compliant design flow is based on the netlist duplication. However, in the FPGA context, keeping two dual instances side-by-side is not trivial; for instance, [22] investigates on the various constraints (for Xilinx & Altera EDA tools) that can be defined at the user-level to force a pairwise placement. A typical mono-output LuT FPGA is the Stratix; Figure 4 illustrates its schematic. Experiments on this target reveal how hard it is to guarantee equal routing for differential signals. For instance, in Figure 5, a complete DES in WDDL is shown. The figure underlines the fact that one net with an extremely high fanout (one bit of the “LR” state addressing the S-Boxes) is doomed to have a different routing, even if the S-Boxes are implemented according to a balanced technique [23]. The idea is that the different location of the “true” and “false” instances will induce a difference in at least one routing path.

Therefore, a native packing of two dual instances into the same reconfigurable resource would be welcome. We choose to assess the robustness of our WDDL 3DES cryptoprocessor when programmed in an Altera Stratix II 90-nm FPGA. Its architecture relies on Adaptive Logic Modules \((ALMs)\), basic building blocks of logic, whose high-level block diagram is shown in Figure 6. On the left, eight inputs drive a combinatorial logic block programmable as either one 6-bit Look-up-Table \((6-LuT)\) or two Adaptive LuTs \((ALuTs)\), both having their own register, named, respectively, reg0 and reg1, on the right.

We have synthesized our implementation using 4-LuT, with the aim to test two PAR strategies. The first one, called “Vertical” strategy, places True and False networks as close as possible by assembling each dual component \((R_M, R_S, XOR_K, \text{etc})\) in the same ALM (see Figure 7(a)). With the second one, the “Horizontal” strategy, routing resources are identical for the True and False networks: \(R_M, R_S\) and XOR\(_K\) are packed into the same ALM, and the dual part is placed in the adjacent ALM (see Figure 7(b)).

The floorplan of the Vertical strategy is shown in the layout of Figure 8. The four ALMs on the top left correspond to four bits of \(R_M\). Two dual wires output from each of them, and they all go to \(R_S\) in the middle, then to XOR\(_K\), on the right, and finally reach the SBox and XOR\(_L\) (both outside of the figure). The wires on the middle top realize the expansion \((E)\) function towards the SBox number \(i + 1\), while those on the middle bottom come from the SBox number \(i - 1\). As explained in Section 2.1, synthesis with positive functions may induce cross connections. Those of XOR\(_K\) are clearly visible on the top right of the figure. This schematic is reproduced for each SBox, thus eight times. Post PAR timing annotations give a first idea of the balance between

\[\begin{align*}
\text{Phase:} & \quad \text{PRE} \quad \text{EVA} \quad \text{PRE} \quad \text{EVA} \quad \text{PRE} \quad \text{EVA} \\
\text{Operation:} & \quad \begin{array}{c}
\text{PRE} \quad \text{EVA} \quad \text{PRE} \quad \text{EVA} \\
0 \quad 0 \quad 1 \quad 1 \\
0 \quad 0 \quad 0 \quad 0 \\
0 \quad 0 \quad 1 \quad 1 \\
1 \quad 1 \quad 0 \quad 0 \\
1 \quad 1 \quad 1 \quad 1 \\
\end{array}
\end{align*}\]

**Figure 2:** Secure return-to-null protocol for a constant activity of the WDDL AND gate depicted in Figure 1.
Figure 4: High-Level block diagram of a LUT, extracted from "Stratix Device Family Data Sheet, volume 1" [24].

Figure 5: Floorplan of a full DES in WDDL (top) and slightly unbalanced fanout of one register of the "LR" state (bottom).

dual wires: for example, the bit of $R_M$ on the top has an imbalance of $289 - 290 = -1$ ps.

Figure 9 illustrates the floorplan for the Horizontal strategy. The upper and lower ALMs contain the true and false networks, respectively. The $reg1$ register from Figure 6 is assigned to the master register while the slave register is implemented in $reg0$ and $XOR_K$ by the combinatorial logic block, on the left. The gain in balance is optimum as timing annotations have exactly the same value.

2.4. Altera Place-and-Route Constraints. Under Altera’s Quartus II design Software, placement is constrained via the LogicLock (LL) regions feature. An LL is a rectangular-shaped region defined by the following:

1. a $state$: locked at an origin location or floating;
2. a $size$: fixed user-defined or automatically determined by Quartus;
3. a $reserved$ property: defining whether entities not assigned to this LL region can use its remaining resources.

These properties can be set in a TCL script, according to the syntax shown in Figure 10.

Constraining the routing follows a different procedure. After fitting, a back-annotation of the routing proposed by
Quartus should be requested. This generates a Routing Constraints File (RCF), whose format is given in Figure 11.

The signal Input1 of this example comes from the output of the LE_BUFFER resource (corresponding to the cluster located at \( X = 1 \), \( Y = 1 \), S(ub-location) = 1, I(ndex) = 0) (device-dependent coordinates, described in the “QSF Assignment Descriptions Document”). It then goes through a vertical wire of length 4 (C4) and the local interconnect of the cluster located just above at \( X = 1 \) and \( Y = 2 \). The final destination is DATAC, an input port on the InputReg1 block.

2.5. Experimental Characterization. Differential traces obtained by DPA can be helpful for a designer to pinpoint countermeasure weaknesses. Indeed, for the right key and monobit analyses, the amplitude of the correlation peak directly gives an experimental evaluation of the power consumption imbalance between true and false networks of a single bit. Many analyses can then be performed. In the following, we present two of them, focusing on the SBox1.

The trace in Figure 11, at the top, corresponds to the power consumption of the FPGA during the beginning of an encryption. At Round 0 evaluation phase, the IP of the message is loaded into LR_S (see Table 1 and Figure 3) after the rising edge of the master clock.
Table 1: Sequence at the beginning of the encryption.

<table>
<thead>
<tr>
<th>DES Round</th>
<th>Initial State</th>
<th>0 (IP)</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDDL phase</td>
<td>Precharge (PRE)</td>
<td>Precharge (PRE)</td>
<td>Evaluation (EVA)</td>
</tr>
<tr>
<td>LR Master</td>
<td>0</td>
<td>IP</td>
<td>0</td>
</tr>
<tr>
<td>LR Slave</td>
<td>0</td>
<td>0</td>
<td>IP</td>
</tr>
</tbody>
</table>

Figure 9: Horizontal strategy.

A few nanoseconds later, the result of the computation is available at the SBoxes output. The differential trace in the middle, obtained by targeting the bit 1 of the SBox1, indeed confirms this. Hence, one correlation peak clearly appears, slightly shifted on the right according to the rising edge of the consumption trace at $t = 418$ ns. Its amplitude, about $-35$ nV, is negative, suggesting that the false network has a power consumption higher than that of the true network.

Finally, on the rising edge Round 1 precharge phase, the value at the output of XOR_K is sampled by LR_M. Once again, a correlation peak emerges in the differential trace in the bottom, generated when targeting bit 16 of LR_M (after the P permutation, the bit 1 of SBox1 becomes bit 16 of LR). This time, the amplitude of the peaks is close to $-60$ nV. As DPA looks for the maximum amplitude, attacks on the LR register are the most powerful in our case.

This phenomenon can either be observed with single rail logic: indeed, registers consume more power than combinatorial logic.

Here, the explanation is that in combinatorial logic, commutation dates are data-dependant. Thus, the power consumption of each encryption vanishes in the differential trace because of a noncoherent averaging due to the intrinsic DPA processing whereas the power consumption of the register is well synchronized.

To gather it, we have to try to use a 4th-order integration [25], unsuccessfully.

To close these analyses, we have reported in Figure 10 the amplitude of the SBox1 output bits correlation peaks when targeting LR. Bit 2 has the highest amplitude, which justifies that it delivers the best performance for the attack (see Table 3). The imbalance is positive for the first and last bits, negative for the others. This experimentally confirms that attacking four bits at once as with unprotected module is less efficient, because bit imbalances counterbalance each other. An improved attack against DPL may be summing up the absolute value of each correlation peak.

2.6. Static Evaluation of the Place-and-Route. In terms of timing differences, the study of the differential PAR quality can be achieved with an analysis of the Standard Delay Format (SDF) file generated by the Quartus II Compiler. Such a file gives interconnection and propagation delays of each instance in the FPGA. Analysis of registers and XOR function timing delays is trivial as less than three couples of dual wires per bit have to be considered. Determining the imbalance in the SBoxes is more difficult; indeed, all delay differences along the datapath have to be summed up.

Mean and standard deviation statistics on timing differences between the true and false network are given in Table 2 in picoseconds. The first column corresponds to the imbalance without specific PAR constraints. It serves as a reference to show the improvement generated by the two PAR strategies. On average, timing imbalances of SBoxes are not much affected by PAR strategies while those of the XOR_K are halved. The Horizontal strategy has a great impact on register imbalance, diminishing them to 3 ps. If the robustness is strongly correlated to the timing differences, this strategy should be the most robust. The next section on experimental results of the robustness evaluation shows that the opposite holds true.

Table 2: Static evaluation of timing imbalance, in ps.

<table>
<thead>
<tr>
<th>PAR Strategy</th>
<th>None</th>
<th>Vertical</th>
<th>Horizontal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Mean</td>
<td>Std Dev</td>
<td>Mean</td>
</tr>
<tr>
<td>LR Master Register</td>
<td>251</td>
<td>322</td>
<td>24</td>
</tr>
<tr>
<td>LR Slave Register</td>
<td>566</td>
<td>327</td>
<td>137</td>
</tr>
<tr>
<td>XOR_K function</td>
<td>501</td>
<td>298</td>
<td>272</td>
</tr>
<tr>
<td>SBox1</td>
<td>202</td>
<td>174</td>
<td>157</td>
</tr>
<tr>
<td>SBox2</td>
<td>169</td>
<td>150</td>
<td>169</td>
</tr>
<tr>
<td>SBox3</td>
<td>165</td>
<td>155</td>
<td>169</td>
</tr>
<tr>
<td>SBox4</td>
<td>146</td>
<td>120</td>
<td>175</td>
</tr>
<tr>
<td>SBox5</td>
<td>131</td>
<td>114</td>
<td>167</td>
</tr>
<tr>
<td>SBox6</td>
<td>149</td>
<td>154</td>
<td>172</td>
</tr>
<tr>
<td>SBox7</td>
<td>170</td>
<td>152</td>
<td>160</td>
</tr>
<tr>
<td>SBox8</td>
<td>156</td>
<td>123</td>
<td>173</td>
</tr>
</tbody>
</table>


3. Security Evaluation

3.1. Background Material. Security evaluation is extremely influenced by various parameters: setups, acquisition conditions, target algorithms, target board, attack models, and so forth. Thus, it is hard to compare SCA countermeasures and attacks of different laboratories on a fair ground. To contribute to the elaboration of common platforms, guaranteeing experiments reproducibility for research institutions, the Tohoku University, and the Japanese Research Center for Information Security (RCIS) has developed in 2007 “Side-channel Attack Standard Evaluation Boards” (SASEBOs) [26]. Dedicated to the evaluation of SCA countermeasures, these electronic boards are distributed free of charge to academic laboratories leading innovative researches in the field of embedded security. Thereby, everyone can reproduce, analyse, and criticize results of their peers.

There are four versions, depending on the target chip. All experiments in this paper have been realized using the SASEBO-B, which incorporates two Altera [24] Stratix II FPGA: one EP2S30F672C5N supposed to embed all control modules, and one EP2S15F484C5N for the cryptographic modules. Having two FPGAs enhances the accuracy of the measurements as it uncouples the power consumption of the cryptographic parts from the power consumption of the others. Another measurement improvement feature is the possibility of using separate power supplies for the core and the input and output pads of the FPGA. Measurements have been done using the original 1 Ω spying shunt resistor in the positive rail of the cryptographic FPGA core power supply. The acquisition board is depicted in Figure 12.

To improve experiment reproducibility by peers, we have used “Eve (Eavesdropper) SoC”, a System on Chip (SoC) providing a flexible way to easily and rapidly design real-life cryptographic applications: when a stand-alone hardware module protected by an SCA countermeasure is fully functional, it can simply be bound to EveSoC as a plug-and-play custom coprocessor, so as to constitute a complete cryptographic device. VHDL code files of EveSoC along with its documentation and miscellaneous tools including scripts for SASEBOs are freely downloadable from its SourceForge development website [27].

Power consumption measurements, referred in the following as traces, have been collected using a 1132A differential probe and a 54855 Infiniium oscilloscope from Agilent Technologies. The final setup has a 6 GHz bandwidth and a 40 GSa/s maximal sample rate.

3.2. Security Gain. As explained in the above section, various parameters affect the security evaluation, and it is always difficult to determine whether the hardness of an attack is caused by a good countermeasure or by a weak adversary. In this context, a framework has been proposed in [28] which suggests to start with an information theoretic analysis to assess the maximal amount of information which could be extracted. This corresponds in practice to the worst case attack, which is difficult to devise in the case of hardware implementations. Then, various distinguishers have to be used to see how a given adversary can take advantage of the
Figure 13: From (a)–(c) single trace and differential traces when targeting the SBox1 and the LR registers.

Figure 14: SASEBO-B configuration for the DPA attacks.
Table 3: Statistics for Bits of R_S Register.

(a) WDDL 3DES Module without PAR Constraints

<table>
<thead>
<tr>
<th>SBox</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit of R</td>
<td>9</td>
<td>17</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>266</td>
<td>917</td>
<td>796</td>
<td>308</td>
</tr>
<tr>
<td>Security Gain</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SBox</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Bit of R</td>
<td>8</td>
<td>14</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>290</td>
<td>686</td>
<td>944</td>
<td>1,014</td>
</tr>
<tr>
<td>Security Gain</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) WDDL 3DES Module with Vertical Strategy

<table>
<thead>
<tr>
<th>SBox</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit of R</td>
<td>9</td>
<td>17</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>38</td>
<td>64</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>Security Gain</td>
<td>50</td>
<td>10</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>SBox</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Bit of R</td>
<td>8</td>
<td>14</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>25</td>
<td>1</td>
<td>29</td>
<td>1</td>
</tr>
<tr>
<td>Security Gain</td>
<td>2</td>
<td>19</td>
<td>13</td>
<td>—</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>2</td>
<td>5</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

(c) WDDL 3DES Module with Horizontal Strategy

<table>
<thead>
<tr>
<th>SBox</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit of R</td>
<td>9</td>
<td>17</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>325</td>
<td>125</td>
<td>125</td>
<td>152</td>
</tr>
<tr>
<td>Security Gain</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>2</td>
<td>1</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>SBox</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Bit of R</td>
<td>8</td>
<td>14</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Timing imbalance, in ps</td>
<td>26</td>
<td>39</td>
<td>204</td>
<td>136</td>
</tr>
<tr>
<td>Security Gain</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Minimal Security Gain</td>
<td>2</td>
<td>8</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

Information leakage. However, with hardware implementation such as the one studied in this article, the construction of templates and the information theoretic analysis is very intensive and may not lead to better results than directly performing an attack. Thus, for a first study, we prefer to limit the security analysis and focus on fewer meaningful attacks.

The strength of these attacks, and thus the robustness against SCA of an implementation which is attacked, was first quantified by evaluating how many Measurements to Disclose (MTD) the secret key are needed. But this number has appeared to be dependent on the values, on the order of the messages, and on the secret key. Therefore, a sound approach is to perform a lot of attacks with different keys, for instance, one hundred, and then to consider the MTD for a given Global Success Rate (GSR), saying 90% or 95%. However, this absolute MTD remains not significant for a firm conclusion. Indeed, what could we think of a countermeasure which needs 1,500,000 traces to be broken, whereas its unprotected version resists analyses up to 1,000,000 measurements? Thus, in the following, we will quantify the security robustness by computing the security gain (SG), that is, the ratio between the MTD of a protected module and the MTD of an unprotected module:

\[
SG = \frac{\text{MTD-GSR90\%protected}}{\text{MTD-GSR90\%unprotected}}
\]

3.3. Experimental Results. With a view to cover a large SCA threat range, we have acquired 6,400,000 traces and performed numerous analyses: Differential Power Analysis (DPA) [29], but also Correlation Power Analysis (CPA) [30], with both Hamming Weight (HW) and Hamming Distance (HD) models, by guessing one to four bits of the main
internal states of the algorithm, that is, the output values of R (register), XOR_K, SBoxes, but not XOR_L as it is tantamount to guessing R. We discard other analyses, such as the template attacks [31], as they have never proved to be practical on parallel implementations of block ciphers. Besides, Mutual Information Analysis (MIA) [32] outperforms CPA [33] only when the side-channel signals are noisy, and leakages multiple, which is not our case. Moreover, we restrict the number of hypotheses to those on the first round key, seen as eight classes of 6-bit each. Hence, an attack on the unprotected module using the HD can only concern R.

Once again, experiments confirm that CPA is more powerful than the DPA in presence of noise (traces have not been averaged) [34]. Best results on unprotected implementations are obtained with the HD model, as it matches the physical phenomenon responsible for power consumption (commutations), and by targeting four bits at the same time, increasing the signal-to-noise (SNR) ratio of the correct peak with respect to the peaks present in incorrect guesses. For WDDL, because of the zero value precharge, best analyses are done by guessing the HW as it equals the number of commutations: HD(0, x) = HW(x). Targeting a single bit is more powerful than four bits: indeed, the leakage in WDDL is caused by the imbalance between the True and the False networks, and this unbalance could be the opposite for targeted bits and therefore counterbalance each other. The same observation has been basically done for quasidelay insensitive asynchronous circuits [35].

We observe that no PAR strategy resists SCA attacks. As the weak element of the implementation is R, we focus in the following on the robustness of each bit of this register. Table 3 summarizes the results for all of the three modules, which are truly comparable as traces have been acquired using the same experimental setup, and the same pseudorandom messages, generated from the same seed. Table 3(a) concerns an unconstrained WDDL cryptoprocessor, serving as reference to estimate the security gain provided by WDDL, and to study the impact of the differential PAR. Tables 3(b) and 3(c) deal, respectively, with the results of Vertical and Horizontal PAR strategies. For Tables 3(a), 3(b), and 3(c), bits coming from the same SBox have been grouped together, and their position in the R register after (P) permutation of 3DES is recalled by the second and sixth lines. For example, bit 1 of SBox 1 becomes bit 9 of R. The third and seventh lines correspond to the timing imbalance in picoseconds. They detail values of Table 2. “Zero” means under the resolution of the timing analysis tool (one picosecond). Fourth and eighth lines provide the SG defined in Section 3.2. The HW model monobit attack has been one hundred times reiterated to get a representative success rate. The SG is then computed with the minimal number of traces needed to recover the full secret key (100% success rate, worst case).

Boldface cells correspond to best cases, that is, smallest value for the timing imbalance and largest value for the SG. The presence of two boldface cells in a same column means that static and experimental evaluations are correlated. An overall look at Table 3 shows that it is never the case. This tends to confirm that evaluating the imbalance with delays is not accurate enough, as they are maximum values, and not typical. One solution to improve analysis may be to rather consider the type of the line, short or long, and to take into account the fanout.

Best cases show that SG is increased by at most a factor 22 when using WDDL without specific efforts of PAR, 364 when using the Horizontal PAR strategy. The Vertical PAR strategy seems to be the more robust, as bit 31 and bit 18, marked with black shaded cells, do not disclose the secret key using 6,400,000 messages. Unfortunately, a reliable security assessment considers the worst case. With 3DES, SG of one SBox equals the minimal SG amongst the SG of its four output bits. Thus, a more accurate conclusion is that the WDDL implementation without specific efforts of PAR increases the robustness a little; the Horizontal PAR strategy multiplies it by 18 to break the SBox 7, but by 6.5 on average; the SG of the Vertical PAR strategy equals 11 for the SBox 7, 5 on average. Referring to Table 1 of [4], MTDs of the unprotected and WDDL-protected modules of Kris Tiri equal 320 and 21,185, respectively, discarding the 5 key bytes not disclosed. The SG is thus close to 66, overriding by one order of magnitude that of our FPGA implementation. However, some bits of the Vertical Strategy present a high robustness: two of them do not result in a disclosure, and two others have an SG of 214 and 352. These results are very promising: we think that we could in the future increase the SG of all of the bits to this level. A reliable approach to reach this goal seems to be in-depth analyses of the link between the routing resources used by a differential path and the SG of its corresponding bit.

4. Conclusion and Perspectives

We have presented in this article a WDDL 3DES implementation with a fully-fledged placement and routing reaching a timing balance lower than 290 ps according to static evaluation. The unconstrained DPL design is shown to be definitely more secure than the reference design. In addition, the PAR strategies we introduced increase further the secure level on top of the unconstrained design. However, despite these encouraging results, the implementations are still attackable with SCA, albeit with more measurements, thus making the attack more difficult.

A detailed analysis has shown that weaknesses originate from register imbalance, although, up to now, efforts were devoted to the design of SBoxes. Security evaluation cannot rely on experiments on a single cryptographic primitive and should concern a complete and real-life application to avoid side effects, by means of solutions such as EveSoc and SASEBO.

Efficient differential place-and-route on FPGA appears to be a great challenge. Perspectives for future works are to validate the influence of technological characteristic variability over the same chip by renewing experiments at different locations, but on another FPGA families as well. We plan also to test other FPGA design solutions, such as Xilinx’s ISE, which may propose more accurate timing analysis tools.
References


Submit your manuscripts at http://www.hindawi.com