Novel Controller for Dummy Rounds Scheme DPA Countermeasure

Petr Moucha, Stanislav Jeřábek, Martin Novotný
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
{mouchpe1, jerabst1, novotnym}@fit.cvut.cz

Abstract—In our previous work, we developed the Dummy Rounds countermeasure to protect the hardware design against side-channel attacks. The scheme employs hiding in time and hiding in consumption. After several improvements of the datapath, the leakage has been minimized significantly. In this paper, we present the enhancement of the Dummy Rounds controller. This enhancement enables further reduction of the leakage. We tested the method on PRESENT cipher implemented in the Sakura-G board. The design was evaluated using Welch’s t-test.

Index Terms—cryptography, round-based ciphers, SCA countermeasure, dummy rounds, controller

I. INTRODUCTION

Modern cryptographic devices use strong ciphers to achieve the highest level of security. Even if the system uses modern, state-of-the-art cipher, it still may be vulnerable to so-called side-channel attacks (SCA). The side-channel analysis exploits the properties of physical implementation of the cryptographic system, namely its power consumption (e.g., Differential Power Analysis (DPA) [1], [2], or Correlation Power Analysis (CPA) [3]), acoustics [4], electromagnetic emanation [5], and more. The side-channel analysis links these properties to the secret intermediate value, which is (at a certain moment) processed in the cryptographic algorithm.

To make the cryptographic system more resistant to SCA, the designer can use techniques generally called Side-channel countermeasures. A lot of Side-channel countermeasures apply to programmable hardware design.

A. SCA Countermeasures

To protect the device against side-channel analysis, the cryptographer may use countermeasures based on masking [6], [7], hiding [8], shuffling [9], or on the combination of these general principles.

Masking is based on mixing the secret intermediate value with a random value. The device power consumption then corresponds to the random value rather than the correct secret value, effectively protecting the device against the first-order attack. Protection against arbitrary-order attack can be achieved by advanced masking methods, e.g., Threshold implementation [10] or Domain-Oriented Masking [11].

Hiding is used to confuse the attacker by the execution of the critical operation at various times or to hide the crucial operation by power consumption noise of other operations. According to the chosen method, we can distinguish between hiding in time and hiding in power. Dual precharge logic [12] [13] can also be considered as a hiding technique because its goal is to achieve constant switching activity (and so power consumption) of the device.

II. PREVIOUS WORK AND OUR CONTRIBUTION

In [14], Jeřábek et al. proposed a Dummy Rounds Scheme, a novel SCA countermeasure that can be utilized in hardware designs of round-based ciphers, such as Feistel Networks [15], or Substitution-Permutation Networks [16]. Later in [17], Jeřábek and Schmidt analyzed the Dummy Rounds countermeasure, they identified some weaknesses, and they also proposed ideas of possible solutions. However, these proposals were not implemented in the same paper.

The proposals from [17] were implemented in our recent work [18]. In this work, we also proposed and implemented further improvements to minimize the side-channel leakage. Test vector leakage analysis [19] revealed that potential leakage still remains within the first clock cycle of encryption, even if no active rounds are processed during the first clock cycle. This problem was also discussed in [17]. To tackle this problem, we propose a new Dummy Rounds controller in this paper.

The paper is structured as follows: In Section III, we summarize the Dummy Rounds Countermeasure. In Section IV, we discuss the proposed modification of the controller. We have experimentally evaluated the proposed controller with the methodology used in [18]. The results of our analysis are summarized in Section V, and our findings are concluded in Section VI.

III. DUMMY ROUNDS COUNTERMEASURE

The Dummy Rounds scheme enables to execute up to \( M \) rounds in one clock cycle; in both Fig. 1 and Fig. 2 the circuits are implementing \( M = 3 \) rounds. The minimum number of rounds that must be executed in one clock cycle is \( m \). The result of \( \mu_i \) rounds, \( m \leq \mu_i \leq M \), is used as the output of the \( i \)-th clock cycle, while the outputs of remaining rounds are discarded. Fig. 1 depicts the circuit introduced in [14]. It executes between \( m = 1 \) and \( M = 3 \) rounds in each clock cycle.

Both \( m \) and \( M \) are constants given by the design of the datapath and the controller. The controller must ensure that
the cryptographic circuit executes exactly $C$ rounds of given cipher within $N$ clock cycles of encryption, i.e.,

$$\sum_{i} N \mu_i = C.$$ 

This arbitrary execution hides the real computation both in time and in consumption.

Recently, in [18] we introduced several modifications of the Dummy Rounds scheme. The block diagram of the most advanced version, which we call Design D, is presented in Fig. 2.

- The scheme enables executing empty clock cycles (i.e., $m = 0$) to make the execution less predictable.
- The rounds, whose outputs are discarded, are now fetched with random data.
- We also use switching registers for storing the intermediate results. In odd clock cycles, the first register stores the intermediate result, while the second register is fetched with random data; in even clock cycles, the first register is fetched with random data, while the second register stores the intermediate result. Similarly to register precharge [20], this feature hides the Hamming distance of intermediate results; however, it does not double the number of clock cycles.
- The number of clock cycles $N$ is not constant. It can vary among several encryptions, and is limited only by the maximum number of clock cycles $N_{\text{max}}$, $N \leq N_{\text{max}}$.

IV. PROPOSED CONTROLLER MODIFICATION

The test vector leakage assessment using Welch’s t-test [19] showed that the leakage of Design D had been significantly reduced, compared to [14]. The maximum leakage is now at the beginning of encryption, see Fig. 3, reaching a maximum t-value of 14.27. As discussed in [17], this leakage is caused by the presence of the plaintext in the working registers since the beginning of the encryption, even if multiple empty clock cycles ($\mu_i = 0$) are executed after the start. A similar problem happens by the end of the encryption when the working register holds the ciphertext until the last clock cycle, even if the execution is filled with a series of empty clock cycles.

To tackle this problem, we modified the controller of Design D to fill the working registers with random data after the start of each encryption. The registers are loaded with the plaintext, and the key just before these data are needed, i.e., just before the beginning of the first non-empty clock cycle. At the end of the encryption, once the correct ciphertext is computed and processed by the other parts of the design (e.g., the communication protocol part), it is overwritten with random data.

V. ANALYSIS

In this section, we provide information on the measurement, and we discuss the results we obtained.

A. Measurement Setup

We implemented the Design D of PRESENT cipher [21] with both the original and enhanced controller. The design was implemented in the SAKURA-G board [22], which is equipped with Xilinx Spartan 6 FPGA. The design implements $M = 3$ rounds, with at least $m = 0$ rounds to be executed in every clock cycle throughout the course of $N_{\text{max}} = 35$ clock cycles for the entire encryption. The design is clocked at 1.5 MHz.

The test vector leakage assessment using Welch’s t-test [19] was based on 1,000,000 power traces. The traces were collected by PicoScope 6404D oscilloscope [23] at the sampling frequency of 312 MS/s. Hence, every clock cycle is covered by 208 samples. SICAK toolkit [24] controlled the measurement—it communicated with the design, collected the power traces from the oscilloscope, and evaluated them with the Welch’s t-test.

B. Results

Fig. 3 and Fig. 4 are depicting plots of t-values obtained during execution of Design D controlled by the original controller [18] and the enhanced controller (this paper), respectively. The numbered vertical lines are highlighting the rising edges of clock cycles. Absolute maximum t-values are summarized in Table I.

When the Design D (see Fig. 2) is controlled by the enhanced Dummy Rounds controller, the maximum absolute
more competitive to other SCA hardware-level countermeasures. Although the maximum absolute t-value of 8.63 still exceeds the security threshold of 4.5, it is very close to it. It is suitable for lightweight ciphers and very competitive to standalone countermeasures presented in [20].

**ACKNOWLEDGMENT**

This work was partially funded by the CELSA project “DRASTIC: Dynamically Reconfigurable Architectures for Side-channel analysis protection of Cryptographic implementations” (CELSA/17/033) and CTU grant No. SGS20/211/0HK3/3T/18.

**REFERENCES**


**TABLE I**

<table>
<thead>
<tr>
<th>Design</th>
<th>Max. t-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original controller [18]</td>
<td>14.27</td>
</tr>
<tr>
<td>Proposed enhanced controller (this paper)</td>
<td>8.63</td>
</tr>
</tbody>
</table>