Side-Channel Leakage through Static Power
Should We Care about in Practice?

Amir Moradi

Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
amir.moradi@rub.de

Abstract. By shrinking the technology *static* power consumption of CMOS circuits is becoming a major concern. In this paper, we present the first practical results of exploiting *static* power consumption of FPGA-based cryptographic devices in order to mount a key-recovery side-channel attack. The experiments represented here are based on three Xilinx FPGAs built on 65 nm, 45 nm, and 28 nm process technologies. By means of a sophisticated measurement setup and methodology we demonstrate an exploitable information leakage through *static* power of the underlying FPGAs. The current work highlights the feasibility of side-channel analysis attacks by *static* power that have been known for years but have not been performed and investigated in practice yet. This is a starting point for further research investigations, and may have a significant impact on the efficiency of DPA countermeasures in the near future.

1 Introduction

After the introduction of execution time [15] in scientific literature as the first practical side channel to recover the secret key of implementations of cryptographic algorithms, other side-channel analysis approaches have been introduced one after each other. For example, power consumption [16], electromagnetic emanation [2,10,21], acoustic [11], optical emission [9], and temperature [14] are amongst those which have been brought to the attention of scientific communities. However, due to their efficiency, low-cost, and simplicity power consumption and electromagnetic emanation side channels have been widely investigated and applied in academia as well as in industry more then the others.

During the golden years of side-channel analysis when academia showed interest in the field, researchers have put much effort in exploring and analyzing the theoretical and practical aspects of side-channel analysis. Not all, but most of the activities in this area have been done based on the principles of CMOS circuits, i.e., focusing on the main power consumption factor of the circuits, namely *dynamic* power consumption. Therefore, the attacks and analysis schemes as well as countermeasure techniques introduced to the community are mainly based on the dynamic power consumption of the underlying circuit. However, during the last years by shrinking the technology the VLSI community reported the dependency of *static* power consumption of a CMOS circuit to its internals (see [13,17]). Moreover, interesting results are shown in [3] and [4], where an

L. Batina and M. Robshaw (Eds.): CHES 2014, LNCS 8731, pp. 562–579, 2014.

attack using static power is called *Leakage Power Analysis* (LPA). There even exist a few works proposing related countermeasures (see [5,27]). This issue, which has been denoted mainly based on the simulation results, was not taken as a serious threat by the side-channel community.

The main reason behind disregarding this information leakage source is due to the very small scale of the signal amplitude (of static power consumption) which cannot be easily measured in practice by means of the currently available facilities and equipments. Indeed, the belief of the community – which is not much far away from reality – is that the information available through the dynamic power consumption channel is much more and much easier to detect compared to that of the static power.

This article demonstrates the first practical results of a side-channel analysis using information leakage through static power consumption. All the experiments shown here are based on Xilinx FPGAs. In order to make the analyses more comprehensive three FPGA families (Virtex-5, Spartan-6, Kintex-7) with three different process technologies, namely 65 nm, 45 nm, and 28 nm, are considered in the experiments.

We first illustrate the measurement setup and the methodology used to exploit the static power of the considered platforms. This includes a couple of engineering adjustments and tricks which make the desired measurement possible. Our experiments start with investigation of dependency of static power to the content of basic elements of FPGA internals, e.g., registers, LUTs, and connections (i.e., routings done by the switch boxes). By means of these experiments we elaborate on a clear dependency between the static power and each of the aforementioned resources for all the targeted platforms. We extend our experiments toward a crypto device by evaluating the static power of an exemplary circuit containing an 8-bit key addition followed by an AES S-box. We demonstrate how to make use of its static power to recover the 8-bit secret key. One step further, we examine a masked AES S-box, and show how to apply a second-order attack through static power consumption. As the final step a complete implementation of an AES encryption engine equipped with both masking and shuffling is considered. We demonstrate in which circumstances an attack using static power can overcome the protection provided by the aforementioned countermeasures.

2 Methodology

Three Side-channel Attack Standard Evaluation Boards (SASEBO) [1]

- SASEBO-GII, with *Target* FPGA as a Virtex-5 (65 nm),
- SAKURA-G, with *Target* FPGA as a Spartan-6 (45 nm),
- SAKURA-X, with *Target* FPGA as a Kintex-7 (28 nm)

are the platforms considered in our experiments. On each board there exists another FPGA (so-called *Control*) responsible to communicate with *Target* as well as with the PC via UART. We have developed a dedicated framework (designs for both *Control* and *Target*) for each of the platforms to fulfill the requirements

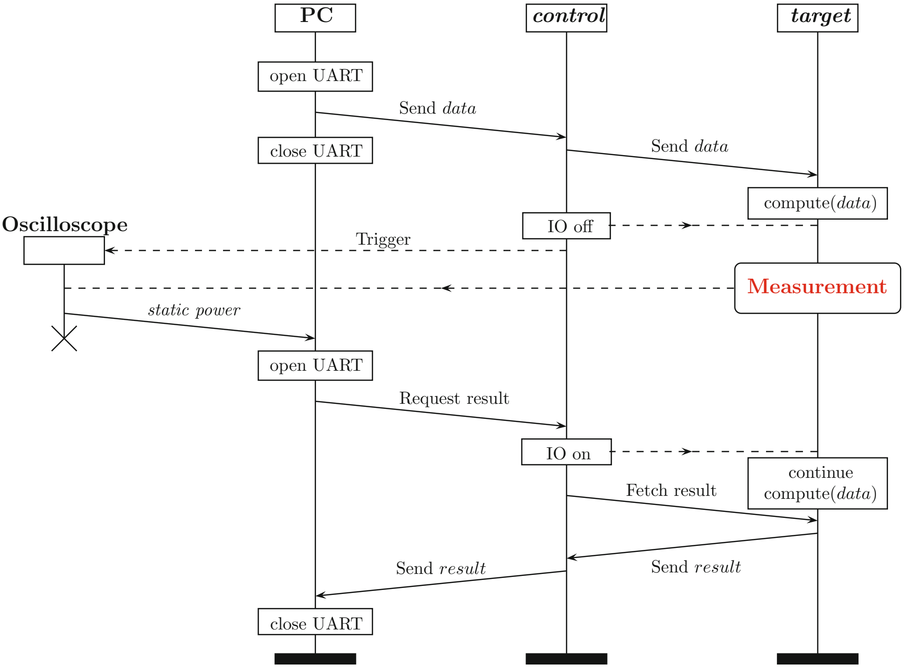


Fig. 1. Procedure of a single measurement of static power consumption

we explain in details below. *Target* is in contact only with *Control*, and all its input signals including the clock signal are provided by *Control*.

2.1 Communication

The procedure which is followed to measure static power consumption of a design embedded on *Target* is depicted by graphics in Fig. 1.

1. The PC opens the UART communication channel and sends *data* to *Control*. Right after that, the PC closes the UART channel.
2. *Control* communicates with *Target* and sends the corresponding *data*. Right after finishing the desired operations on *Target*, *Control* switches off all the IO pins including the clock of *Target*.
3. *Control* issues a trigger signal to the oscilloscope. After that, the static power consumption of *Target* can be measured (explained later).
4. The PC opens the UART channel and requests *Control* to send the result back.
5. *Control* switches on the IO signals, drives the *Target* clock, and fetches the result of the desired computation done by *Target*.
6. *Control* sends the fetched result to the PC via UART, and the PC closes the UART channel right after the reception.

Our experiments show that the IO signal values have a significant effect on the amount of static power consumption. Therefore, as stated in the above procedure, the output signals of *Target* as well as of *Control* which drive the inputs of *Target* must be at a constant state (e.g., all at LO) during the measurement. Further, noise of the UART channel, which is realized by a USB module (FTDI chip[1]) on SASEBO platforms, also hugely affects the static power. So, keeping the UART channel closed during the measurement is inevitable.

2.2 Measurement

The measurement point provided by the SASEBO boards is the heads of a resistor placed in the Vdd path of *Target* internal core. According to the SASEBO quick start guides [1], a usual way to measure the voltage drop over this shunt resistor is to monitor the voltage of the *Target* Vdd pins. It should be noted that setting the coupling of the corresponding oscilloscope channel to AC, which might be beneficial to reduce the measurement noise when measuring dynamic power, cannot be used in our case. It is because the AC coupling is a kind of a high-pass filter which stops the DC part of the signal. However, we are interested to measure the DC shift of the power consumption signal to be able to monitor the static power. Therefore, keeping the DC coupling of the corresponding oscilloscope channel is a must.

Another issue regarding the measurement is due to the small-scale shunt resistor. The resistor originally embedded on the SASEBO boards is 1 Ω for SASEBO-GII (Virtex-5) and SAKURA-G (Spartan-6) and 10 mΩ for SAKURA-X (Kintex-7). Since we are planing to measure the current passing through this resistor by monitoring its voltage, the magnitude and type of this resistor significantly affect our measurement accuracy. Therefore, we replaced the shunt resistor of all the platforms by a certain type 1.0 Ω resistor with low temperature coefficient. So, we use the same shunt resistor in all our experiments. Further, due to the voltage drop by the shunt resistor we modified the boards[2] to supply a certain voltage thereby driving exactly 1.0 V at *Target* internal core Vdd pins. This way, *Target* of all our platforms are supplied by the same voltage magnitude.

We should mention that amplifiers like `ZFL-1000LN+` from Mini-Circuits[3] or `PA303` from Langer EMV-Technik[4], which are common components used for enhancing measurement of small-scale dynamic power signals, cannot be equipped in our setup. That is because these amplifiers have a high-pass filter at their input removing the DC shift of the incoming signal. The same holds for the amplifier originally embedded on SAKURA-G (Spartan-6). Instead, we have used a LeCroy AP 033 differential probe which includes a ×10 internal amplifier and does not cause the aforementioned problem. By means of the differential probe and a LeCroy HRO66Zi WaveRunner 12-bit oscilloscope we monitored

[1] Future Technology Devices International Ltd. `http://www.ftdichip.com/`

[2] By adjusting the potentiometer of the corresponding voltage regulator.

[3] `http://www.minicircuits.com/`

[4] `http://www.langer-emv.de/`

the voltage drop by the shunt resistor. Since the differential probes consist in active components, they usually introduce higher noise to the resulting signal compared to common coaxial-cable passive probes.

Each measurement is performed by sampling the amplified signal (output of the differential probe) with the highest vertical accuracy (200 μV/div in our setup), at a sampling rate of 1 GS/s and bandwidth limit of 20 MHz. A long trace with a length of 10 ms containing 10 M sample points is measured, and its average is computed by the oscilloscope. In contrast to a dynamic power measurement, where a trace over time is collected, a singular value (the aforementioned averaged value) is the result of a static power measurement. This procedure (see Fig. 1) can be repeated to collect the magnitude of static power consumption for different *data* values.

3 Preliminary Studies

According to the VLSI theory and the simulation results [13,17] *leakage current* (directly proportional to static power) of a CMOS gate depends on the content of its output as well as its inputs. In the following – by means of a couple of case studies – we try to investigate the effect of the FPGA internals on the amount of the chip's leakage current.

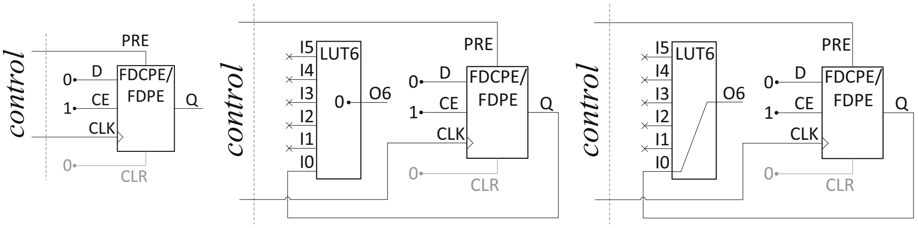
3.1 Registers

As the simplest case study we consider the registers as of fundamental elements available in any FPGA. We first considered *Target* of SASEBO-GII (Virtex-5) and made a design consisting of several registers. All registers are configured as FDCPE, i.e., "D Flip-Flop with Clock Enable and Asynchronous Preset and Clear" (see [25]). As shown by Fig. 2(a), CE (clock enable) is always '1', and D (register input) is connected to '0'. So, by a positive edge at CLK (clock) the register stores '0'. Further, since the register is configured as "with Asynchronous Preset and Clear", the register stores '1' by seeing HI level ('1') at the PRE (preset) signal. CLK and PRE of all registers are connected together and are handled by *Control*. Therefore, *Control* can change the content of the registers by handling these two signals. More precisely, when signals (CLK, PRE) change from (0,0) to (1,0) or to (0,1) the registers save '0' or '1' respectively. Also changing the signals back to (0,0) does not alter the registers content. This indeed helps us to switch off the IO signals without affecting the internals when measuring leakage current as explained in Section 2.

We have implemented 14 400 instances[5] of the above explained register, and controlled the placement process to place them in desired locations[6]. An important issue is regarding the Q(out) signal of the registers. These signals are not

[5] Half of the available registers in Virtex-5 LX50.

[6] The placement of the registers does not affect the result of this experiment, but the manual placement is done to keep its consistency with the next experiments as explained later.



(a) Register (b) Register+Connection (c) Register+Connection+LUT

Fig. 2. Design of basic elements of the preliminary case studies

connected to anywhere. This gives us the chance to examine only the effect of the register contents on static power. In order to avoid optimization and trimming the unconnected resources by the synthesizer tools, we explicitly forced the tools to keep these signals[7] thereby preventing the registers to be trimmed.

In the measurement phase, the leakage current of two cases is to be measured: *i*) when all the registers contain '0' and *ii*) when all the registers contain '1'. As stated before, during both measurements the environmental situation like the IO signals – of both *Target* and *Control* – must be the same, and the difference between these two cases must only be the content of the registers. We followed the procedure explained in Section 2 for each case separately to obtain two singular values as amounts of corresponding leakage current. Repeating this process 1000 times (done in 17 minutes) led to two curves shown by Fig. 3(a).

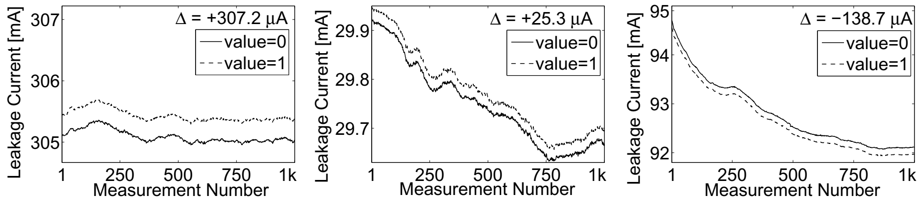
As shown by the graphics, the dependency of the leakage current to the registers' content is clear. Although the leakage currents greatly vary over time, their difference (of two cases) is relatively constant. We realized that the reason behind this remarkable variation is temperature inconstancy. The chip temperature as well as room temperature significantly affects the leakage current measurements[8]. Since the temperature of the equipped differential probe steadily increases after power up, it also has a huge impact on the measured leakage current. In order to diminish these issues we employed a thermobox to isolate the platform and the differential probe from environmental temperature variations. This makes the situation better, but does not completely solve the problem.

By repeating the same experiment with the same number of registers on two other platforms, SAKURA-G (Spartan-6) and SAKURA-X (Kintex-7)[9], we obtained the leakage current curves shown by Fig. 3. Dependency between static power and the registers' content is obviously shown, but comparing these three results brings some interesting conclusions:

[7] By KEEP and SAVE NET FLAG constraints (see [23]).

[8] As an interesting experience, approaching human body ($\sim 37°$C) to the FPGA chip causes the leakage current to rapidly change.

[9] FDCPE instances are replaced by FDPE "D Flip-Flop with Clock Enable and Asynchronous Preset" as FDCPE does not exist in Spartan-6 and Kintex-7 libraries (see [24] and [26]).



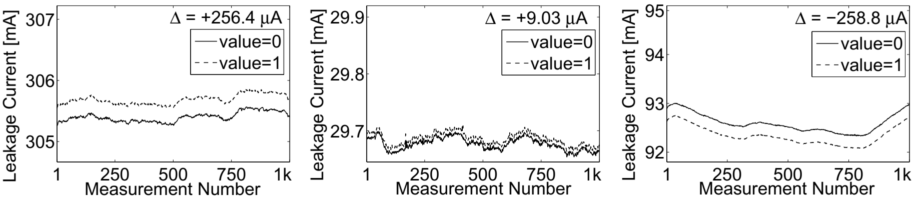
(a) SASEBO-GII (Virtex-5) (b) SAKURA-G (Spartan-6) (c) SAKURA-X (Kintex-7)

Fig. 3. Measured leakage current of 14 400 registers on all three platforms

- In case of SASEBO-GII (Virtex-5) and SAKURA-G (Spartan-6), when the content of the registers is ‘1’, the leakage current is higher compared to when the registers stored ‘0’. This polarity is reversed in case of SAKURA-X (Kintex-7). Since the underlying FPGAs are from different families with different technologies, and the details of each process technology are not publicly available, we cannot comment on this behavior.
- Leakage current of SASEBO-GII (Virtex-5 65 nm) ∼ 300 mA is much higher than that of other platforms with lower process technology. Also it does not decrease by shrinking the technology as ∼ 30 mA for SAKURA-G (Spartan-6 45 nm) and ∼ 90 mA for SAKURA-X (Kintex-7 28 nm). Note that we supplied all three FPGAs with the same internal core voltage (1.0 V).
- Moreover, the part of the leakage current related to the registers’ content is not higher for smaller process technologies. 307 μA, 25 μA, and 138 μA respectively for the 65 nm, 45 nm, and 28 nm chips. It means that side-channel vulnerability of these circuits through static power does not necessarily increase by shrinking the technology. In our experiments, the difference between leakage current of two cases (registers = ‘1’ or ‘0’) of SASEBO-GII (Virtex-5 65 nm) is the highest compared to that of the others.

3.2 Connections

The FPGA internal connections are realized by programmable switch boxes which play an important role regarding the amount of (dynamic) power consumed by a design. The number of switch boxes, which exist in the routing of a signal, significantly affects its delay as well as the energy consumed when it toggles. The more switch boxes a signal passes, the higher is its toggles’ power consumption. Accordingly, the amount of leakage current of a switch box, which is made by CMOS circuits, should be affected by the value of the signal. In order to examine this issue we have developed the next experiment. As shown in Fig. 2(b), in the same way as in the last experiment a register is employed. The output of the register is given to a look-up table (LUT6) whose



(a) SASEBO-GII (Virtex-5) (b) SAKURA-G (Spartan-6) (c) SAKURA-X (Kintex-7)

Fig. 4. Measured leakage current of 14 400 registers + connections on all three platforms

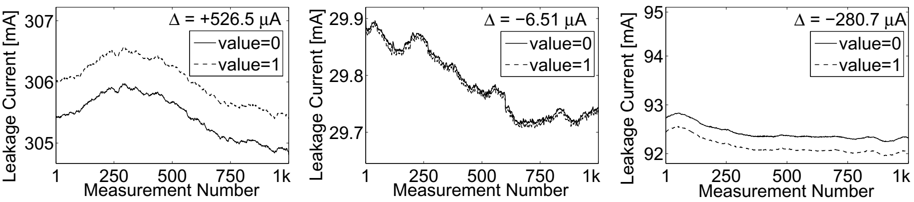
output – regardless of its inputs – is always ‘0’. This gives us the opportunity to exclude the effect of the LUT6 output toggles in our investigations. However, having the LUT6 in the design is mandatory; otherwise the signal routing (switch box connection) will not be realized.

In order to limit the number of switch boxes involved in each signal routing, the register and the connected LUT6 instance are forced to be placed at the same slice by manual placement. In order to make the connection the register output must be routed to the CLB[10]-dedicated switch box and come back to the same slice to be connected to the LUT6 input (see Fig. 12). It indeed makes a loop going out of and coming back to the slice[11]. It also guarantees that only one switch box is involved in each signal routing. Similar to the last experiment, we developed a design as *Target* with 14 400 instances of these elements. The same placement as that of the last experiment is done here to keep the consistency of the two experiments. Further, we provided appropriate constraints to avoid trimming the registers and the LUT6 instances since the LUT6 output is connected to nowhere.

After developing this design on all our platforms we measured the corresponding leakage current 1000 times for each of the cases of the registers' content. The results of the measurements are shown by Fig. 4. By comparing the results of the last and the current experiments on SASEBO-GII (Virtex-5) (Fig. 3(a) and Fig. 4(a)), it becomes clear that the difference between leakage current of two cases of the registers' content is smaller when the connections are added to the design. It can be concluded that the polarity of the dependency of leakage current to the value of the connected signals is the inverse of that to the registers' content. The same behavior is seen for the second platform SAKURA-G (Spartan-6); compare Fig. 3(b) and Fig. 4(b). However, the last platform SAKURA-X (Kintex-7) behaves differently (see Fig. 3(c) and Fig. 4(c)). Introducing the connections to the design increases the difference between the measured leakage currents. It means that the effect of the value of the connected signals on leakage current has the same polarity as that of the register's content.

[10] Configurable Logic Block: containing two slices in Virtex-5, Spartan-6, and Kintex-7 families. Each slice consists of four LUT6 and at least four registers.

[11] Note that an opposite connection which connects a LUT6 output to a register input at the same slice does not necessarily leave the slice and pass a switch box.



(a) SASEBO-GII (Virtex-5) (b) SAKURA-G (Spartan-6) (c) SAKURA-X (Kintex-7)

Fig. 5. Measured leakage current of 14 400 registers + connections + LUTs on all three platforms

As stated before, since these FPGAs are developed under different technologies, these observed dissimilar behaviors cannot be easily justified. It is worth to mention that these results stand for a couple of connections made around a slice then repeated several times for other slices. Based on these results we cannot conclude about the effect of every connection made by switch boxes in an FPGA.

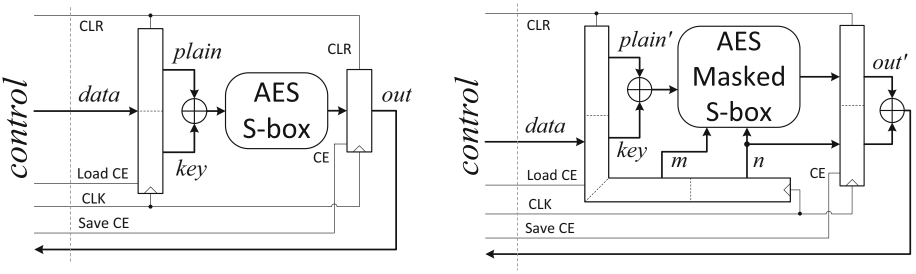
3.3 Look-Up Tables

As the last experiment with FPGA fundamental elements, we examine the effect of the look-up table's (LUT) output value on leakage current. As shown by Fig. 2(c), compared to the last experiment we only changed the configuration of the LUT6 to make its output always the same as its first input, i.e., the register output. In this design when the register output toggles, the value of the routed signal (connection) as well as the value of the LUT6 output changes. This way, sum of the effect of all these three elements on leakage current is observed in this experiment. Repeating the last experiment with the slightly modified designs (only changing the LUT6 configurations) led to the results shown in Fig. 5. It should be noted that everything including the number of elements (14 400), placement, and routing are the same as that of the experiment expressed in Section 3.2.

Comparing the results of this experiment with that of two previous ones it can be concluded that the LUT6 output value has a considerable impact on leakage current in case of SASEBO-GII (Virtex-5). The same influence with a smaller factor can be seen on the other platforms. The polarity of this dependency – similar to the previous experiments – is different from one platform to another.

Table 1. Dependency of leakage current to basic FPGA internal elements

Platform	FPGA	Technology	Register			Connection			LUT		
			μA	%	↕	μA	%	↕	μA	%	↕
SASEBO-GII	Virtex-5	65 nm	307.20	49	↑	50.80	8	↓	270.10	43	↑
SAKURA-G	Spartan-6	45 nm	25.30	44	↑	9.03	29	↓	6.51	27	↓
SAKURA-X	Kintex-7	28 nm	138.70	49	↓	120.10	43	↓	21.90	8	↓



(a) S-box

(b) Masked S-box

Fig. 6. Design of exemplary circuits

To sum up the result of the experiments expressed so far Table 1 presents the amount and polarity of dependency of leakage current to the targeted fundamental elements on all our three platforms.

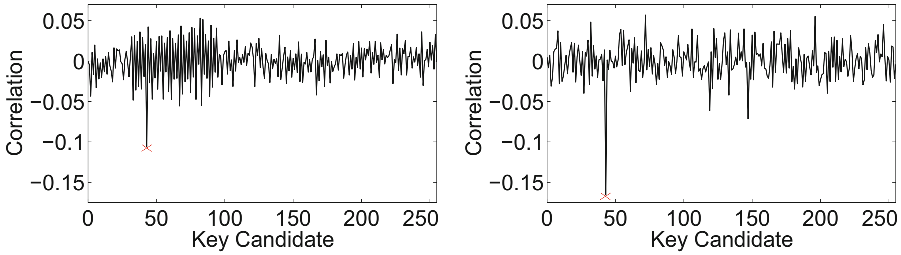
3.4 AES S-box

Up to now all the presented case studies were based on many registers, connections, and LUTs having the same value. In order to move toward a crypto device, and examine whether a side-channel attack is possible we developed the fourth case study as an 8-bit key XOR followed by an AES S-box. For the S-box circuit we took the very small design of [7]. A diagram of the circuit is shown by Fig. 6(a). Two 8-bit registers supply the inputs of the key addition and the S-box, and one register is responsible to save the S-box output. All the registers are handled by *Control*.

As shown in previous experiments, the dominant term affecting leakage current is temperature variations. Therefore, to exploit the amount of leakage current relevant to the processed data we should continuously measure the leakage current of a deterministic state of the underlying device, e.g., RESET after power up. In case of our exemplary AES S-box design, forcing the device to RESET state is done by handling the `CLR` signal which causes all three registers to clear their content. Therefore, to diminish the effect of temperature we followed the below procedure:

1. *Control* forces *Target* to RESET state by setting `CLR` signal.
2. The procedure of Fig. 1 is followed to measure leakage current as l_{RESET}.
3. $data = (plain, key)$ as input is provided by *Control* for *Target*.
4. Again based on the procedure of Fig. 1 leakage current as l_{data} is measured when the S-box output is ready to be saved in the register.
5. The amount of leakage current related to *data* is reported as $(l_{data} - l_{\mathrm{RESET}})$.

Therefore, for each given *data* the measurement process should be performed twice to obtain a singular value as relevant leakage current. Apparently the delay between these two measurements should be kept as small as possible.



(a) Normal Placement

(b) Wide Placement

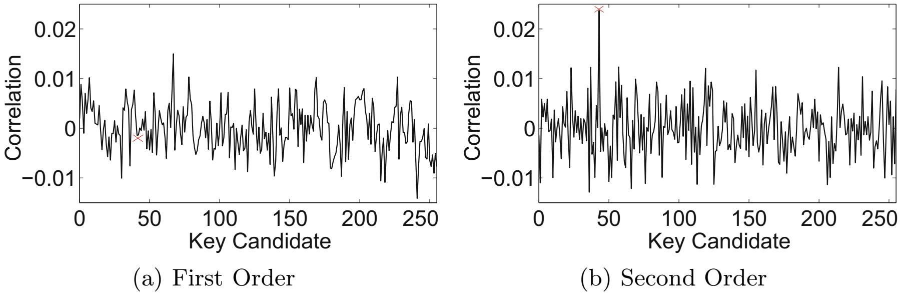
Fig. 7. CPA attack using HW of the S-box output on leakage current of the design of Fig. 6(a) implemented on SAKURA-X (Kintex-7), 10 000 measurements

For the rest of the experiments we focus on the third platform SAKURA-X (Kintex-7). After implementing the aforementioned design on *Target*, we kept the 8-bit *key* constant and performed the procedure explained above 10 000 times with random 8-bit *plain* values. This way we obtained 10 000 measurements of leakage current related to the known *plain* values. In sum, whole of the measurement process took around 2.5 hours. Similar to when applied on dynamic power traces, a power analysis attack can now be mounted using the collected measurements of leakage current. Several techniques like DPA [16], CPA [6], and MIA [12] can be used to examine whether the selected *key* value can be extracted. An obvious difference to when they are applied on dynamic power traces is the absence of time domain since each measured static power (leakage current) is a singular value.

We have tried the aforementioned power analysis techniques with different hypothetical models. The result of a CPA attack with Hamming weight (HW) model (S-box output) is shown in Fig. 7(a). The efficiency of the attack is obvious, but it is strongly affected by the placement and routing strategy of *Target*. A different placement and routing causes a different number and types of connections to be used to realize the design. As shown by the presented experiments, this directly affects the amount of leakage current related to the value of the connected signals. For example, forcing the S-box output register to be placed far away from the S-box combinatorial circuit causes the corresponding connections to be very long passing many switch boxes. This has a huge impact on the attack (CPA-HW) efficiency as shown in Fig. 7(b). As a short notice, since the internal connections (signal routings) are amongst the dominant factors affecting leakage current of an FPGA, in contrast to what is reported in [17] for a simulated ASIC, HW model might be not necessarily a suitable model in case of FPGAs.

3.5 Masked AES S-box

Now an interesting question is whether a higher-order attack is possible through static leakage when the implementation is equipped with a masking



(a) First Order

(b) Second Order

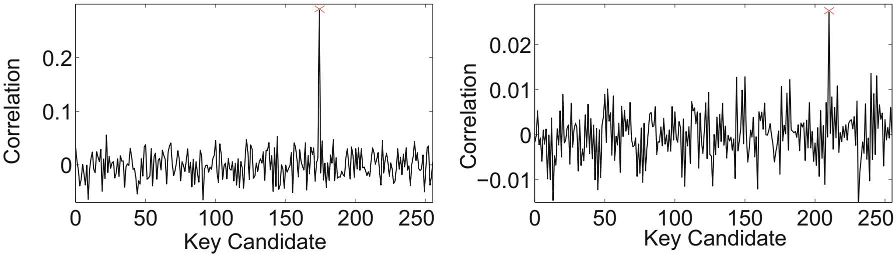
Fig. 8. CPA attack using HW of the S-box output on leakage current of the design of Fig. 6(b) implemented on SAKURA-X (Kintex-7), 50 000 measurements

countermeasure. So, another exemplary design as shown in Fig. 6(b) is taken into account. The masked AES S-box is taken from the very compact design of [8] which realizes first-order Boolean masking. For each given *plain* value, two 8-bit values m and n as input and output masks are randomly selected. *data* as $(plain', key, m, n)$ is composed and sent to *Target* by *Control*, where $plain'$ denotes $plain \oplus m$ (masked input). When the input registers have stored *data*, the masked S-box output as $S(plain \oplus key) \oplus n$ is ready to be saved in the output register. Similar to the last experiment, we took SAKURA-X (Kintex-7) as the platform and performed the leakage current measurements according to the procedure explained in Section 3.4. During all 50 000 measurements (taken in 12 hours) *plain* as well as m and n were selected randomly while the *key* value was kept constant.

The first-order leakage of the underlying masked S-box design through dynamic power is known (see [18]). We have also tried to mount a *correlation collision attack* [18] to examine its first-order leakage through leakage current (see Appendix). Nevertheless, as shown by Fig. 8(a) CPA attacks using common models (S-box output HW) are ineffective to recover the secret. However, a second-order attack is expected to be efficient. So, the collected leakage current values are made mean free and then squared. Afterwards, the same CPA attack, indeed a zero-offset second-order attack [22], is performed whose result is depicted by Fig. 8(b). It clearly shows that the same principles of higher-order attacks are valid in case of leakage current. The main difference is due to having only univariate measurements in this case.

4 Realistic Scenario

After performing quite exhaustive preliminary experiments, it is now time to examine under which conditions a crypto device can be attacked through its static power. We have developed a full AES-128 encryption engine with a 32-bit width datapath, where at each clock cycle a column is processed as four S-boxes or one MixCoulmns. Figure 13 shows an overview of the design, as can be seen both masking and shuffling are employed. The masking scheme is

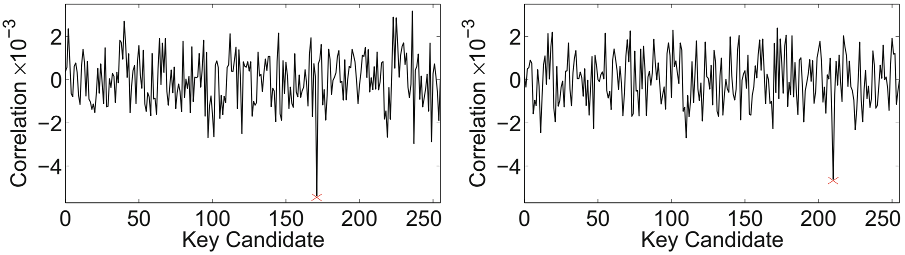


(a) Key byte 5 (b) Key byte 6

Fig. 9. AES-128 encryption engine, PRNG off, CPA attack results with S-box output HW model, 100 000 measurements

first-order Boolean, and the underlying masked S-box is the same as the one used in Section 3.5. The shuffling is realized by randomly selecting the order of processing the columns (Sel_Col signal). Moreover, during the computation of SubBytes the four instances of the S-box circuit are randomly assigned to the given column (Instance Shuffling signal). Within loading the plaintext, key, and masks the initial masking as well as AddRoundKey are performed. Then, the SubBytes operation is performed in 4 clock cycles. Afterwards, it takes 4 clock cycles to finish MixColumns and AddRoundKey at the same time. During this period the S-box instances are used by KeySchedule. The "Mask Correction" unit also changes the masks after each MixColumns and prepares the round output to be again masked by input mask m. Clearly at the "Final Round" MixColumns is not operated, and the mask of the S-box output is removed after the last AddRoundKey and before saving them back to the state register.

PRNG Off. The design is implemented on SAKURA-X (Kintex-7), and for the first try the PRNG which generates random values for input and output masks m and n as well as for shuffling (Sel_Col and Instance Shuffling) is switched off. As stated before, for leakage current measurements we require a deterministic state, e.g., RESET, to continuously measure its relevant leakage current. Since in the underlying FPGAs the content of the registers after power up is deterministic (specified as '0' or '1' by the bitstream), we continuously power down and up the *Target* FPGA in order to obtain l_{RESET} for each *data*-dependent leakage current measurement. After supplying a new *data* by *Control*, *Target* is kept running till end of the SubBytes operation of the first cipher round, i.e., 4 clock cycles after starting the encryption. As stated before, at this time instance all IO signals provided by *Control* including `CLK` go LO; then the leakage current of *Target* is measured. It should be noted that for each relative leakage current measurement *Target* is powered down and up again to obtain a new l_{RESET}. In this setting we repeated the leakage current measurements 100 000 times when supplying the inputs by random 128-bit plaintexts and a constant 128-bit key.



(a) Key byte 5

(b) Key byte 6

Fig. 10. AES-128 encryption engine, PRNG on, second-order CPA attack results with S-box output HW model, 1 000 000 measurements

A similar attack as before, CPA by HW of the S-box output, might be able to recover the secrets. During the attack we noticed that four key bytes can be detected much easier than the others. When the leakage current is being measured, one column of the SubBytes result stored in the state registers is available at the MixColumns circuit's input. The key bytes related to this column are discovered easier compared to that of other columns which are only available at the column-selecting multiplexer. Figure 9 shows the result of the attack targeting two different key bytes. Indeed, this result shows the same concept as attack using dynamic power. The leakage solely related to the registers' content is not easily detectable, but when they drive a considerably large combinatorial circuit, e.g., an AES S-box, the data-dependent leakage is much more exploitable. This is in fact the reason behind the high efficiency of Hamming distance (HD) model when attacking a hardware design through its dynamic power.

PRNG On. As the last experiment we repeated the previous procedure while the PRNG is switched on and provides uniformly distributed values. The PRNG is embedded on *Control*, seeded by the PC after each power up, and all required random values are sent to *Target* before starting the encryption. Therefore, reseting *Target* to obtain l_{RESET} does not affect the distribution of the random numbers provided by *Control*. This time we performed 1 000 000 leakage current measurements which took 10.7 days. As shown in Section 3.5, we can mount a second-order attack. In this case when the leakage current is measured, a randomly selected column (by Sel_Col) appears at the MixColumns input. Therefore, all key bytes can be recovered relatively with the same effort. The results of the second-order CPA with HW model on two key bytes are shown in Fig. 10. In short, the attack succeeds even when both masking and shuffling are applied.

5 Conclusions

In this work we have presented the first practical results of using static power to mount a successful side-channel attack. All the results illustrated are based on three FPGAs and a couple of exemplary circuits. Note that it cannot be concluded that any implementation on any FPGA can be broken by means of its static power. The results we observed and the conclusions we gave may not hold for another FPGA family or for an ASIC platform. In addition, there are a couple of important facts which should be noted:

- The main power-consuming components in FPGAs are connections (signal routings). This is not true for ASIC platforms, and wires should not significantly affect the chip leakage current. In this case the registers' content and gates' output should be the main leakage sources.
- Although we have used a specific measurement setup, a dedicated setup to amplify the DC signals as well as to reduce the noise by low-pass filters should be developed for further analyses.
- By means of e.g., a climate chamber a constant temperature should be maintained during the static power measurements.
- The measurements of static power are more time consuming compared to that of dynamic power.
- Due to the very small amplitude of the signal as well as high noise, Signal to Noise Ratio (SNR) in this case is much smaller than that of dynamic power. Therefore, many measurements are required to mount a successful attack.
- Similar to the case of using dynamic power, knowing the design architecture of the device under attack in some cases is essential for a successful key-recover attack. It is more critical to know at which time instance (which clock cycle) the IO should be off to measure the static power.

The current study shows that the attacks using static power are practical, but – using the current facilities and known measurement setup – they are still less efficient than the attacks using dynamic power. Moreover, in case of static power attacks the adversary model is quite strong as he/she ideally needs to control the clock signal. So, many other attacks, e.g., fault injection attacks, are potentially possible.

We should highlight that the results demonstrated here are preliminary, and there are many more issues to be discovered in practice. If it is confirmed by practice for ASIC platforms or micro-controllers (of course by a sophisticated measurement setup) the masking schemes might be in danger. The leakage is always univariate in case of static power, and the leakage of different shares of a shared secret are always added and can be seen through the device static power. Therefore, the designs like [20] as a univariate-resistance approach will be vulnerable through static power (e.g., using higher-order moments) similar to only-first-order-resistant approaches like [19].

References

1. Side-channel Attack Standard Evaluation Board (SASEBO). Further information are available via, `http://www.morita-tech.co.jp/SAKURA/en/index.html`
2. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side-channel(s). In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Heidelberg (2003)

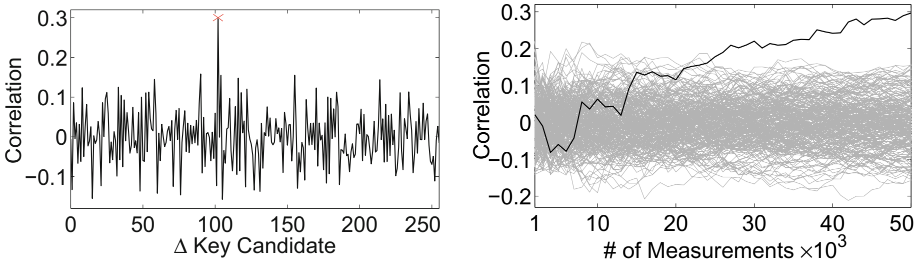
3. Alioto, M., Giancane, L., Scotti, G., Trifiletti, A.: Leakage Power Analysis attacks: Well-defined procedure and first experimental results. In: Microelectronics 2009, pp. 46–49. IEEE (2009)
4. Alioto, M., Giancane, L., Scotti, G., Trifiletti, A.: Leakage Power Analysis Attacks: A Novel Class of Attacks to Nanometer Cryptographic Circuits. IEEE Trans. on Circuits and Systems 57-I(2), 355–367 (2010)
5. Basel Halak, A.Y., Murphy, J.: Power Balanced Circuits for Leakage-Power-Attacks Resilient Design. Cryptology ePrint Archive, Report 2013/048 (2013), `http://eprint.iacr.org/`
6. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
7. Canright, D.: A Very Compact S-Box for AES. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 441–455. Springer, Heidelberg (2005), `http://faculty.nps.edu/drcanrig/pub/index.html`
8. Canright, D., Batina, L.: A Very Compact "Perfectly Masked" S-Box for AES. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 446–459. Springer, Heidelberg (2008), the corrected version at Cryptology ePrint Archive, Report 2009/011, `http://eprint.iacr.org/`
9. Ferrigno, J., Hlavác, M.: When AES blinks: introducing optical side channel. IET Information Security 2(3), 94–98 (2008)
10. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic Analysis: Concrete Results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001)
11. Genkin, D., Shamir, A., Tromer, E.: RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. Cryptology ePrint Archive, Report 2013/857 (2013), `http://eprint.iacr.org/`
12. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008)
13. Giorgetti, J., Scotti, G., Simonetti, A., Trifiletti, A.: Analysis of data dependence of leakage current in CMOS cryptographic hardware. In: ACM Great Lakes Symposium on VLSI, pp. 78–83. ACM (2007)
14. Hutter, M., Schmidt, J.-M.: The Temperature Side Channel and Heating Fault Attacks. In: Francillon, A., Rohatgi, P. (eds.) CARDIS 2013. LNCS, vol. 8419, Springer, Heidelberg (2014)
15. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
16. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
17. Lin, L., Burleson, W.: Leakage-based differential power analysis (LDPA) on sub-90nm CMOS cryptosystems. In: ISCAS 2008, pp. 252–255. IEEE (2008)
18. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-Enhanced Power Analysis Collision Attack. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 125–139. Springer, Heidelberg (2010)
19. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. J. Cryptology 24(2), 292–321 (2011)
20. Prouff, E., Roche, T.: Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 63–78. Springer, Heidelberg (2011)

21. Quisquater, J.-J., Samyde, D.: ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards. In: Attali, S., Jensen, T. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001)
22. Waddle, J., Wagner, D.: Towards Efficient Second-Order Power Analysis. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 1–15. Springer, Heidelberg (2004)
23. Xilinx. Constraints Guide (2008), `http://www.xilinx.com/itp/xilinx10/books/docs/cgd/cgd.pdf`
24. Xilinx. Spartan-6 Libraries Guide for HDL Designs (April 2012), `http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/spartan6_hdl.pdf`
25. Xilinx. Virtex-5 Libraries Guide for HDL Designs (April 2012), `http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/virtex5_hdl.pdf`
26. Xilinx. Xilinx 7 Series FPGA Libraries Guide for HDL Designs (April 2012), `http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/7series_hdl.pdf`
27. Zhu, N., Zhou, Y., Liu, H.: Counteracting leakage power analysis attack using random ring oscillators. In: Sensor Network Security Technology and Privacy Communication System 2013, pp. 74–77. IEEE (2013)

Appendix

First-Order Leakage of the Masked S-box

In order to examine the first-order leakage of the masked AES S-box of Fig. 6(b), we collected two sets of 50 000 leakage current measurements with two different 8-bit *key* values. Similar to that of [18] we estimated the mean of these two sets based on the value of *plain* and obtained two 256-element mean vectors. Permuting one of the mean vectors based on the guessed Δkey and correlating with another mean vector led to the result shown by Fig. 11. Indeed it confirms that the same concept as first-order leakage of the employed masked S-box is valid in case of leakage current as the attack can recover the linear difference between two selected *key* values.



(a) (b)

Fig. 11. Correlation Collision attack on leakage current of the masked AES S-box design of Fig. 6(b) implemented on SAKURA-X (Kintex-7), (a) using 50 000 measurements, (b) over number of measurements

Supporting Figures

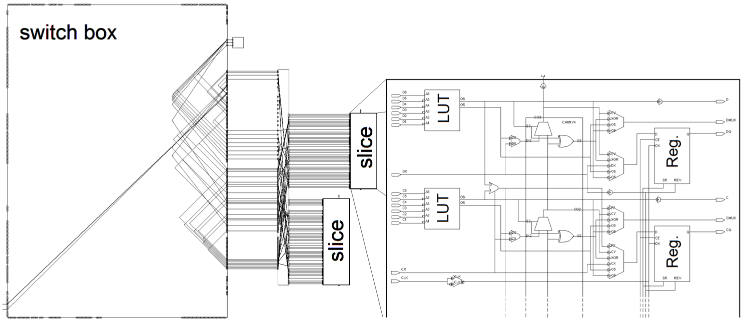


Fig. 12. Virtex-5 internal architecture, CLB (two slices) and its dedicated switch box

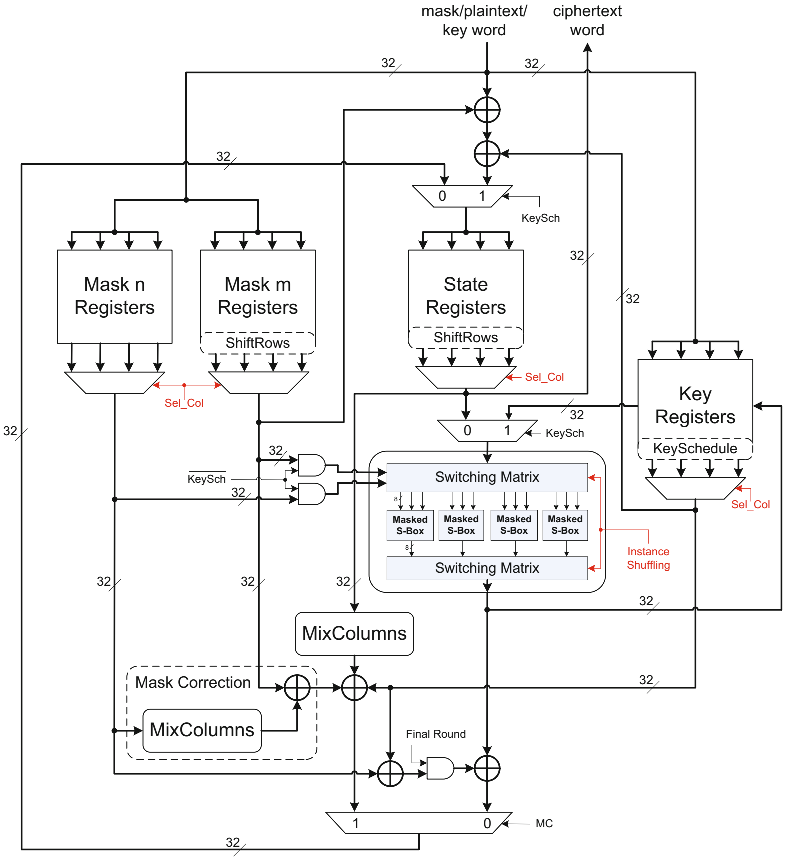


Fig. 13. Architecture of the masked shuffled AES encryption engine