Securing Cryptographic Circuits by Exploiting Implementation Diversity and Partial Reconfiguration on FPGAs

Benjamin Hettwer[†*], Johannes Petersen[‡], Stefan Gehrer[†], Heike Neumann[‡], Tim Güneysu[*]
[*]Horst Görtz Institute for IT-Security, Ruhr University Bochum, Germany
[†]Robert Bosch GmbH, Corporate Research, Renningen, Germany
[‡]Hamburg University of Applied Sciences, Germany
Email: benjamin.hettwer@de.bosch.com, johannes.petersen@haw-hamburg.de, stefan.gehrer@bosch.com, heike.neumann@haw-hamburg.de, tim.gueneysu@rub.de

***Abstract*—Adaptive and reconfigurable systems such as Field Programmable Gate Arrays (FPGAs) play an integral part of many complex embedded platforms. This implies the capability to perform runtime changes to hardware circuits on demand. In this work, we make use of this feature to propose a novel countermeasure against physical attacks of cryptographic implementations. In particular, we leverage exploration of the implementation space on FPGAs to create various circuits with different hardware layouts from a single design of the Advanced Encryption Standard (AES), that are dynamically exchanged during device operation. We provide evidence from practical experiments based on a modern Xilinx ZYNQ UltraScale+ FPGA that our approach increases the resistance against physical attacks by at least factor two. Furthermore, the genericness of our approach allows an easy adaption to other algorithms and combination with other countermeasures.**

***Index Terms*—Physical attack, side-channel attacks, fault attacks, partial reconfiguration, FPGAs**

I. Introduction

Physical attacks such as Side-Channel Attacks (SCAs) and Fault Attacks (FAs) have been well-known threats for cryptographic devices for more than 20 years. SCAs are mainly passive, and the adversary takes advantage of the correlation between the processed data and physical observations which can be made during normal operation of the device. Notable examples from literature are Correlation Power Analysis (CPA) [1] and Electromagnetic (EM) analysis [2]. The principle of FAs is to actively induce errors while executing the algorithm in order to provoke an abnormal behavior. This can be achieved, e.g., through supply voltage/clock glitching, laser, or EM injection [3].

Along with attacks, countermeasures have been presented as well which are usually dedicated either for SCAs or FAs exclusively. A combination of several approaches is therefore often employed in practice. However, this is not only cost intensive, but can also lead to unintended side-effects which may render one or more countermeasures inoperative [4]. In [5] an approach was presented which takes both of the aforementioned attack vectors into account by changing parts of a cryptographic circuit dynamically using Partial Reconfiguration (PR). This feature of modern Field Programmable Gate Arrays (FPGAs) enables the intrinsic modification of certain blocks of logic during runtime, while the remaining logic continues to operate without interruption.

In this work, we propose an alternative way to protect cryptographic circuits using PR. In particular, we use a single synthesized netlist to generate different physical configurations of the same Register Transfer Level (RTL) description of a cryptographic algorithm that are dynamically exchanged via PR. As a case study we apply our technique to a serialized implementation of the Advanced Encryption Standard (AES). Effectiveness of the proposed countermeasure against power and EM-based SCAs is evaluated with real measurements acquired from a ZYNQ UltraScale+ evaluation board. Additionally, we qualitatively demonstrate the increased robustness against a broad range of FAs. Our countermeasure is especially interesting in the context of self-adaptive, respectively self-healing hardware concepts [6]. These are usually based on PR and thus, the proposed scheme creates only little overhead since the employed reconfiguration infrastructure can be reused.

This work is supported in parts by the German Federal Ministry of Education and Research (BMBF) under grant agreement number 16KIS0606K (Security by Reconfiguration - SecRec)

II. Preliminaries

A. Partial Reconfiguration

PR means that parts of the configuration can be exchanged during runtime (without shutdown), whereas the remainder of the logic stays active. This feature is useful for systems at which certain subfunctions are only needed for shorter time periods (e.g. different signal processing filters) in order to save area. When used in an application, the designer first defines the static and dynamic parts of the design and allocates the corresponding regions on the chip. Then, exactly one bitstream is created for the static part, and separate partial bitstreams for the parts that are dynamically switched during runtime. The size of a partial bitstream is proportional to the size of the reconfigurable area [7] and they can be stored

either on external (e.g. Flash) or internal memory (e.g. Block RAM). For the actual reconfiguration, additional hardware support in form of a Partial Reconfiguration Controller (PRC) is needed. The PRC is a dedicated IP core which fetches the partial bitstreams from memory and transfers them to the Internal Configuration Access Port (ICAP) [8], which in turn configures the Programmable Logic (PL).

B. Constraining Placement and Routing on FPGAs

FPGA implementation refers to the step of placing and routing a synthesized design (i.e. netlist) onto the resources of the target platform. Considering current design tools, e.g., Xilinx Vivado, this process can be controlled by various options including different implementation strategies, physical and timing constraints, and Relatively Placed Macros (RPMs) [9]. Implementation strategies are pre-defined sets of options, each having a dedicated purpose on the implementation result (e.g. to reduce area). They allow to implicitly adjust placing and routing on a very abstract level. Timing constraints also implicitly affect the physical layout of the design but allow fine-grained access to individual signals, e.g., by defining minimum and maximum path delays. In contrast, physical constraints enable to explicitly map logical elements to cell locations and locking down the routing. As an alternative, certain fabric elements may be blocked for placing and routing using physical constraints. RPMs are defined on RTL to group a set of logical elements in order to place them in close proximity of each other on the chip.

III. Countermeasure Conception

The main idea of our countermeasure is to generate a number of functionally invariant but physically different configurations of a cryptographic algorithm, and constantly exchange them during operation using PR. Physical diversity between the configurations is achieved by different placement and routing on the fabric, whereas the netlist is kept constant. The rationale behind the approach is that shuffling of the configurations with varying dynamic power consumptions will increase the number of traces needed for mounting a successful CPA, since averaging the power traces using different input data will tamper the correlation between the real and the estimated power consumption [10]. Furthermore, as the location of sensitive operations as well as the wiring is not static, local EM analysis and FAs will be significantly hampered. Our method is not only a combined countermeasure against several types of physical attacks, but can also be applied to various kinds of cryptographic implementations due to its generic nature.

In the upcoming subsections, we discuss trade-offs that need to be considered when using the approach for a specific design.

A. What to reconfigure?

A straightforward choice when deciding which parts of the cryptographic implementation should be dynamic would be to reconfigure only the sensitive parts, such as S-Boxes in symmetric primitives or the modular multiplication in asymmetric ciphers. This would have the advantage of small partial bitstreams that could be stored in internal memory. Also the time overhead for reconfiguration would be less significant. However, most of the logic would still be static and thus could be attacked as in an unprotected setting. In order to gain a higher degree of diversity, the complete cipher operation could be switched dynamically for the cost of larger bitstreams.

B. When to reconfigure?

There are different options regarding the time of reconfiguration: After a number of round operations, after a number of cipher invocations, or a combination of both. Reconfiguration of the design within an encryption/decryption operation decreases the level of SCA leakage more efficiently due to the higher amount of shuffling induced. However, it comes with the drawback of requiring additional registers for holding the cipher state (context storage). As these context registers are naturally an interesting target for the adversary (e.g. when performing an EM attack), they have to be secured by an additional countermeasure. Reconfiguration after a number of encryption runs has the advantage that no adjustments of the RTL design are needed, and the performance decrease would be less significant. However, in such a setting, the adversary may be able to acquire several traces having the same configuration which makes the analysis easier compared to reconfiguration after some rounds.

C. How to generate circuit variants?

The primary goal is to obtain variants in form of partial bitstreams that exhibit major differences regarding their physical layout. One way to achieve this, considering the possibilities given in Section II-B, is to randomly prohibit a number of slices or Configurable Logic Blocks (CLBs) of the fabric for each variant that is generated. However, there might be implementation specific aspects that should be kept in mind as well. For example, it is favorable to place dedicated parts of the cryptographic design (e.g. S-Boxes and corresponding registers) densely in order to make the routing as short as possible, since a higher fanout induces a greater SCA leakage. Another important aspect that influences the degree of diversity of the variants is the size of the reconfigurable area. When defining a larger area, more freedom regarding placement and routing is available which increases the amount of physically distinct implementations.

IV. Case Study

A. Architecture

For the case study, we used an 8-bit serialized AES-128 encryption core which requires 274 Look-up Tables (LUTs) and 244 registers. We implemented our prototype architecture on a Xilinx ZYNQ UltraScale+ System-on-Chip (SoC) evaluation board. To enable dynamic reconfiguration, we equipped the hardware design with a PRC and an ICAP interface. The PRC is clocked with the maximum frequency of 200 MHz to keep the time needed for changing the partial bitstreams low.

B. Partial Bitstream Generation

We decided to switch the complete AES implementation using PR, while only keeping the control logic static. By doing so, the position and wiring of all important logic elements is changed on each reconfiguration, and we do not have to consider a specific adversary that attacks a dedicated position or part of the algorithm (e.g. the S-box).

In order to guarantee a high level of diversity between the configurations, approximately 25 % of the fabric has been defined as reconfigurable area. The size of this area can be varied depending on the security requirements. For generating the variants, we implemented a Tool Command Language (Tcl) script that automates the process using Xilinx Vivado. The Tcl script takes as input a synthesized netlist of the reconfigurable logic and outputs a partial bitstream. On every run, 80 % of the slices of the reconfigurable are randomly prohibited for placement. Once the placement for the aforementioned parts of the netlist is finished, the remainder of the design is placed, routed, and a partial bitstream is created. 128 different configurations have been generated using the described method. One partial bitstream has a size of 6.16 MB, thus the complete memory overhead for storing all AES variants is 128×6.16 MB ≈ 790 MB. As the internal memory of the FPGA fabric only has a capacity of 32.1 Mb [11], all configuration data is stored on external flash.

V. Evaluation

A. Resistance against Power-based SCAs

We collected power traces by measuring the current through a decoupling capacitor of the power supply of the FPGA as proposed in [12]. Although the physical principal behind measurements on the decoupling capacitors is actually more related to EM attacks, the information contained in the signal corresponds to the power consumption of the complete logic that is loaded on the PL. Our setup is therefore comparable to or even better than standard power-based attacks where the voltage drop over a shunt resistor is measured, as only higher-frequency components, which are of interest for SCAs, are flowing through the decoupling capacitor.

As evaluation metric, we used a CPA with Hamming Distance (HD) power model targeting two consecutive S-box outputs during the first round. The result of such an attack on the unprotected AES version is shown in Fig. 1a. Analogously, we performed the same attack on the protected implementation where another configuration is loaded by the PRC after each encryption. Results for that attack are shown in Fig. 1b. In both cases, we clocked the AES core with 30 MHz. From Fig. 1 one can observe that the reference implementation is broken the first time after around 8250 traces. The protected version with enabled reconfiguration shows the highest correlation with the correct key hypothesis after 24 500 traces. Due to the serialized nature of the used AES implementation we expect similar results when attacking the remaining key bytes, as all of them processed one after the other by the same hardware structures.

In order to explore the influence of the number of configurations on the result, we conducted several measurements with a varying number of partial bitstreams (1, 16, 32, 64, 96, 128) and calculated the correlation values for the correct key hypothesis using 25 000 traces each time. We observed no linear decrease of the correlation. However when comparing the setup with only a single configuration and the setup with 128 configurations, the maximum correlation is decreased by a factor of 2.2. Even though not complete in line with the rule of thumb given in [13] (i.e. halving the correlation coefficient means that four times more traces are needed), it corresponds approximately to the results depicted in Fig. 1.

B. Resistance against EM Attacks

For demonstrating the efficiency of our proposed countermeasure against localized EM attacks, we created correlation heat maps for two exemplary configurations which are shown in Fig. 2. A Langer ICS 105 4-axis system has been used for the positioning of the EM probe over a grid of $15 \times 9 = 135$ positions. On each location, 5000 encryption operations with an averaging factor of 100 were acquired and subsequently correlated with the power model for the correct key hypothesis. From Fig. 2 we can draw several conclusions. First, the results for the two heat maps are completely different, meaning the locations that exhibit the highest leakage are far away from each other. The second important observation is that the maximum correlation for the configuration that was created according to Section IV-B is substantially lower (0.06 compared to 0.1). We can therefore argue that an adversary would need on average n times more measurements to extract the key, if PR is enabled with n different configurations.

C. Resistance against FAs

We revise the adversarial model described in [14] for fault analysis evaluation of our approach. In that setup, the fault induction process is a probabilistic event which can be characterized by the success rate $p = p_A \cdot p_D \cdot p_T$, where p_A is defined as the probability to hit the correct target area, p_D as the probability to hit the correct depth, and p_T as the probability to affect the correct time interval in order to successfully provoke a fault.

Without loss of generality, we set $p_D = 1$ as hitting the suitable target depth is highly dependent on the used fault injection technique. The countermeasure presented above performs the same operations at the same clock cycles. As we neither make assumptions on the timing resolution of the fault setup nor on the cipher operation that is targeted, we can assume p_T also to be 1. With enabled PR, configurations are exchanged in a random order. Considering this and the point that the configurations exhibit major differences regarding their physical layout, p_A can be generally derived to $\frac{1}{n}$. In our specific implementation, the overall success rate can be calculated to $p = 1/128 \approx 0.008$, which means that the probability of injecting a fault is lowered to under 1%.

VI. Conclusion

In this work we have presented a generic countermeasure against different types of physical attacks based on dynamic

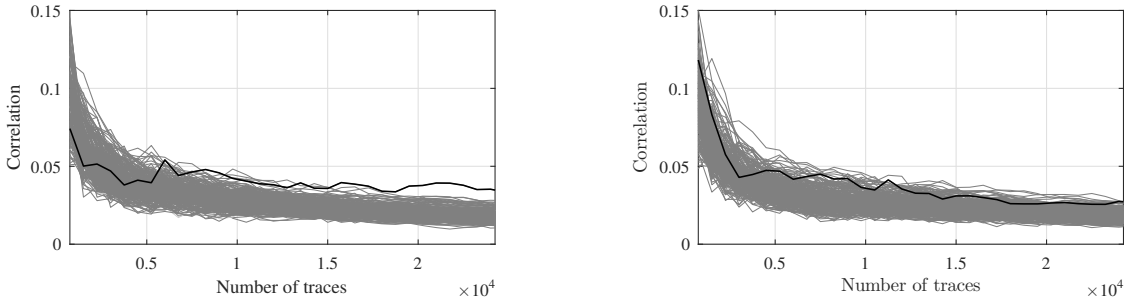

Fig. 1: Maximum absolute correlation over number of traces (a) reference implementation, (b) implementation with enabled reconfiguration. Correct key hypotheses are shown in black.

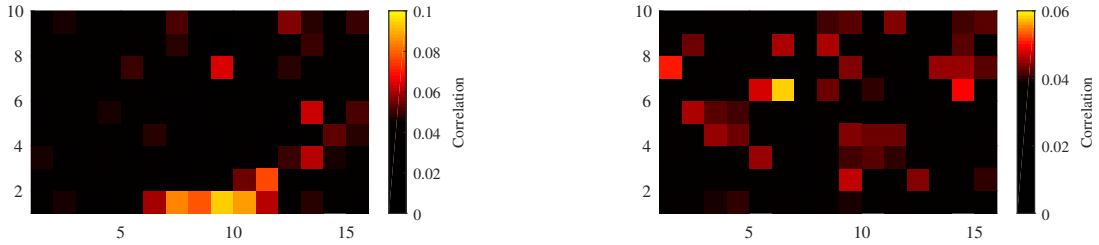

Fig. 2: Correlation heat maps for (a) the reference implementation and, (b) the first configuration of the countermeasure.

partial reconfiguration of FPGAs. We could practically demonstrate that our proposed approach is very effective especially with regard to localized EM analysis and FAs, as the position of sensitive logics on the chip is not static anymore. Drawback of the method is the high resource utilization, which is roughly doubled in terms of logic and internal memory. Also the resistance against power-based SCAs is only increased by a factor of 2-3. However, we stress that our approach can be complemented by additional SCA countermeasures that operate on an algorithmic level (e.g. masking) in a straightforward manner to achieve a higher level of protection.

REFERENCES

[1] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Springer Berlin Heidelberg, 2004, pp. 16–29.

[2] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Springer Berlin Heidelberg, 2001.

[3] M. Joye and M. Tunstall, *Fault Analysis in Cryptography*. Springer Publishing Company, Incorporated, 2012.

[4] F. Regazzoni, L. Breveglieri, P. Ienne, and I. Koren, *Interaction Between Fault Attack Countermeasures and the Resistance Against Power Analysis Attacks*. Springer Berlin Heidelberg, 2012, pp. 257–272.

[5] N. Mentens, B. Gierlichs, and I. Verbauwhede, "Power and fault analysis resistance in hardware through dynamic reconfiguration," in *Cryptographic Hardware and Embedded Systems – CHES 2008*, E. Oswald and P. Rohatgi, Eds. Springer Berlin Heidelberg, 2008, pp. 346–362.

[6] D. C. Keezer and J. Yang, "Biologically inspired hierarchical structure for self-repairing fpgas," in *2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Dec 2017, pp. 1–8.

[7] "AR# 63419: Vivado Partial Reconfiguration - What types of bitstreams are used in Partial Reconfiguration (PR) solutions?" [Online]. Available: https://www.xilinx.com/support/answers/63419.html

[8] "Partial Reconfiguration Controller." [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/prc/v1_3/pg193-partial-reconfiguration-controller.pdf

[9] "Vivado Design Suite User Guide-Implementation." [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug904-vivado-implementation.pdf

[10] M. Stöttinger, S. Malipatlolla, and Q. Tian, "Survey of methods to improve side-channel resistance on partial reconfigurable platforms," in *Design Methodologies for Secure Embedded Systems*, A. Biedermann and H. G. Molter, Eds. Springer Berlin Heidelberg, 2011, pp. 63–84.

[11] "Xilinx Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit." [Online]. Available: https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html#overview

[12] C. O'Flynn and Z. Chen, "A case study of side-channel analysis using decoupling capacitor power measurement with the openadc," in *Foundations and Practice of Security*, J. Garcia-Alfaro, F. Cuppens, N. Cuppens-Boulahia, A. Miri, and N. Tawbi, Eds. Springer Berlin Heidelberg, 2013, pp. 341–356.

[13] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Berlin, Heidelberg: Springer-Verlag, 2007.

[14] K. Lemke-Rust and C. Paar, "An adversarial model for fault analysis against low-cost cryptographic devices," in *Fault Diagnosis and Tolerance in Cryptography*, L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, Eds. Springer Berlin Heidelberg, 2006, pp. 131–143.