

# IEEE Copyright Notice

Copyright © 2021 IEEE

Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted.

Accepted to be published in: 2021 30th IEEE Asian Test Symposium (ATS'21).

# Side-Channel Attacks on Triple Modular Redundancy Schemes

Felipe Almeida, Levent Aksoy, Jaan Raik, and Samuel Pagliarini  
Department of Computer Systems

Tallinn University of Technology, Tallinn, Estonia

Email: {felipe.almeida, levent.aksoy, jaan.raik, samuel.pagliarini}@taltech.ee

**Abstract**—Triple Modular Redundancy (TMR) is a well-known fault tolerance technique for avoiding errors in the Integrated Circuits (ICs) and it has been used in a wide range of applications. The TMR technique employs three instances of circuits realizing concurrently the same functionality whose outputs are compared through a majority voter. On the other hand, Side-Channel Attacks (SCAs) are powerful techniques to extract secret information from ICs based on the data collected from security critical operations. Over the years, the interplay between security and reliability is poorly studied. In this paper, we explore the performance of SCAs on the well-known Advanced Encryption Standard (AES) and its different realizations using the TMR technique. In this work, three implementations of the AES design under the TMR scheme are used and an SCA, which can collect power dissipation data from the physical netlist through simulations, is developed. The experimental results show that the TMR technique can increase the computation time of SCAs and more importantly, the use of functionally equivalent, but physically and structurally different instances in the TMR scheme can make it impossible for SCAs to discover the secret key.

**Index Terms**—triple modular redundancy, side-channel attacks, advanced encryption standard.

## I. INTRODUCTION

As the semiconductor industry pushes the limits of transistor technology in a never ending pursuit of miniaturization, radiation effects have become a serious concern not only for aerospace and military applications, but also for terrestrial applications. Among many radiation effects an Integrated Circuit (IC) may suffer from, Single-Event Transients (SETs) and Single-Event Upsets (SEUs) [1] are widely studied. The underlying principle is that a charged particle, upon striking the IC, may cause shifts in voltage levels at combinational or sequential elements, creating SETs or SEUs, respectively.

Over the years, many efficient techniques have been used to mitigate radiation effects [2], often making the use of some notion of spatial or temporal redundancy [3]–[7]. Triple Modular Redundancy (TMR), one of the most commonly utilized solutions, is a technique that employs three instances of a module and adds a majority voter at their outputs. The scheme, therefore, protects against any single fault in any of the modules. The TMR technique can be deployed with different levels of granularity [6], [7], with diversification [8], and also with approximation [9]. It also presents partial protection against multiple faults caused by single-event-induced charge sharing [6].

However, when a fault tolerant circuit is implemented using the TMR or a similar technique, its resiliency against security vulnerabilities tends to be overlooked. Recently, the field of Hardware Security has received a lot of attention and defense techniques against various adversaries have been implemented for a range of circuits. Yet, the interplay between security techniques and fault tolerance methods is still poorly understood.

In this paper, our aim is to highlight this interaction by taking an Advanced Encryption Standard (AES) crypto core as a case study. The reliability technique we are concerned with is TMR in its many forms. In this work, we realize three possible AES designs under the TMR scheme. While the first one has the identical AES instances, the second one includes the same AES instances optimized by the synthesis tool, and the third one has functionally equivalent, but physically and structurally different AES instances obtained by the clock gating [10] and retiming [11] design techniques. The security attack we are concerned with is the power analysis based side-channel attack (SCA). We develop our SCA which extracts the simulated power dissipation data from a physical implementation of the AES design and guesses the secret key using a statistical procedure. To the best of our knowledge, for the first time, we perform SCAs on an AES design implemented under the TMR scheme. We show that the discovery of the secret key in the design under a TMR scheme needs a large simulation data, increasing the computation time of the attack when compared to the single AES design. We also point out that the use of functionally equivalent, but physically and structurally different instances in a design implemented using a TMR technique increases the resiliency to SCAs due to different power traces in each AES instance, making it impossible to discover the secret key while the other designs under the TMR scheme are vulnerable to SCAs.

The rest of this paper is organized as follows: In Section II, we present the background concepts related to SCAs on crypto cores. The implementation of an AES crypto core and its different realizations using the TMR technique are described in Section III. We introduce our SCA based on power analysis in Section IV. Experimental results are given in Section V and finally, the paper is concluded in Section VI.

## II. BACKGROUND

In an SCA, an adversary collects, in a non-invasive way, leakage data that can be used to discover private information

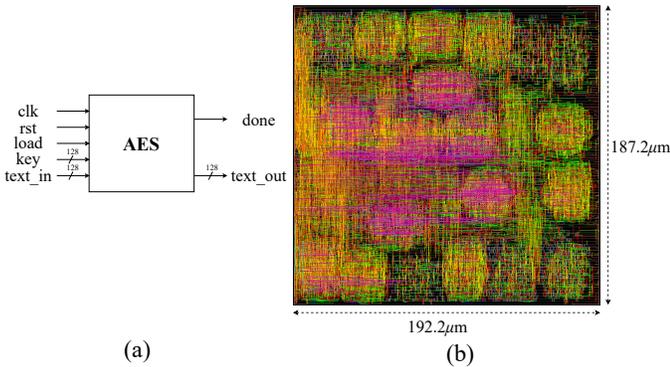


Fig. 1: (a) Block diagram of the AES circuit; (b) its layout.

and/or to gain privileged access to a circuit [12]. Power consumption, timing, electromagnetic emanations, and even sound are examples of side-channels that can and have been exploited. Based on the analysis of this residual information, it is possible to perform an attack that breaks security assumptions. In this paper, our focus is on SCAs that exploit power traces as a form of leakage. The power analysis based SCAs can be categorized in three groups: i) Simple Power Analysis (SPA); ii) Differential Power Analysis (DPA); iii) Correlation Power Analysis (CPA). SPA is a simple graph analysis of the power trace consumption over time. DPA uses statistical analyses at different times to correlate power consumption measurements with functionality. CPA uses a Hamming weight power model method [13] for a more powerful attack.

Crypto cores have been the typical targets of SCAs. In principle, the mathematics behind the crypto function is sound and cannot be broken by formal crypto analysis. However, the physical realization of the crypto function gives adversaries powerful information.

In [14], an evaluation of the sensitivity to DPA of several protected versions of an AES circuit is discussed. In [15], a power analysis attack on an AES hardware implementation is presented and an SCA is mounted on a physical device with the aid of a simple setup (scope and probes). The attack utilizes the power consumption during the first two clock cycles of the AES computation to discover the secret key. The reason for which the attack works is that in the considered AES implementation, an XOR operation between the plaintext and the secret key is executed in the first clock cycle. The result of this operation is saved in an *intermediate register* in the second clock cycle. The adversary can devise a **hypothetical power model** to account for changes in the value of the intermediate register, i.e., the adversary can use bit changes in this register as a proxy for the behavior of the power consumption of the entire AES circuit. Even further, by simulation means, the adversary can analyse all possible changes the register might have, e.g., toggle count, in a cycle-accurate manner. This type of modeling is widely utilized in SCAs to discover the secret key in a device that implements AES.

Previous interactions between reliability and security can be found in mitigating hardware Trojans using the TMR technique. In [16], an optimized graph partitioning of the TMR

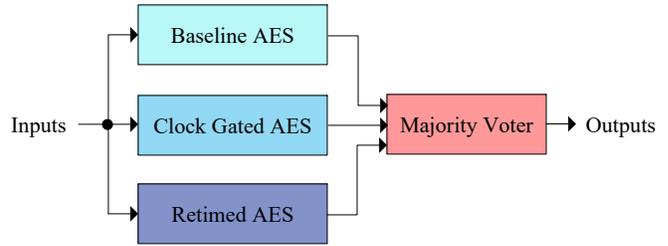


Fig. 2: Structure of the AES\_TMR\_DIF design.

technique is used against hardware Trojan in a reconfigurable hardware and a fine-grain TMR architecture is presented to mitigate multiple faults and hardware Trojan insertion in [17].

This paper explores the performance of an SCA on the AES design under a TMR scheme. The proposed attack focuses on power consumption information leakage to discover the secret key in an AES crypto core. We assume that the AES core is meant for a high-dependability application and therefore, TMR has been applied to it. We also assume that the adversary has access to power traces of the circuit under attack. Furthermore, our approach emulates a physical attack by obtaining detailed power traces from physical synthesis. In practice, a real attack is more complicated because the environment, board, and package become sources of noise that have to be accounted for. We direct the readers to [15] for more details on attack feasibility.

### III. AES CRYPTO CORE IMPLEMENTATION AND ITS REALIZATIONS USING THE TMR TECHNIQUE

As a case study for an SCA on a design under a TMR scheme, the AES crypto core is considered. Fig. 1(a) shows its block diagram. The AES circuit takes a 128-bit secret key (*key*) and a plaintext (*text\_in*) as inputs and produces a ciphertext as an output (*text\_out*).

A 128-bit AES crypto core is obtained from [18] and implemented in a standard design flow. Initially, the logic synthesis of Verilog Hardware Description Language (HDL) codes of the AES circuit into a gate-level netlist is realized using the Cadence Genus tool with a commercial 65 nm standard cell library when the target frequency is 500 MHz. Then, physical synthesis, including floorplanning, placement, clock tree, and routing, is performed by the Cadence Innovus tool. Fig. 1(b) presents the AES crypto core layout. This is our baseline implementation and is referred to as the *single AES* in the rest of the paper.

The same AES crypto core was designed under a coarse-grain TMR architecture. Three different physical designs, called AES\_TMR\_IDE, AES\_TMR\_OPT, and AES\_TMR\_DIF, were considered. In the AES\_TMR\_IDE design, each instance in the TMR architecture is intentionally made identical: all cells and all metal routing lines are the same for all three instances. In the AES\_TMR\_OPT design, the physical synthesis tool is allowed to perform independent optimizations in these three instances if applicable. Finally, in the AES\_TMR\_DIF design, each instance is determined to be physically and structurally

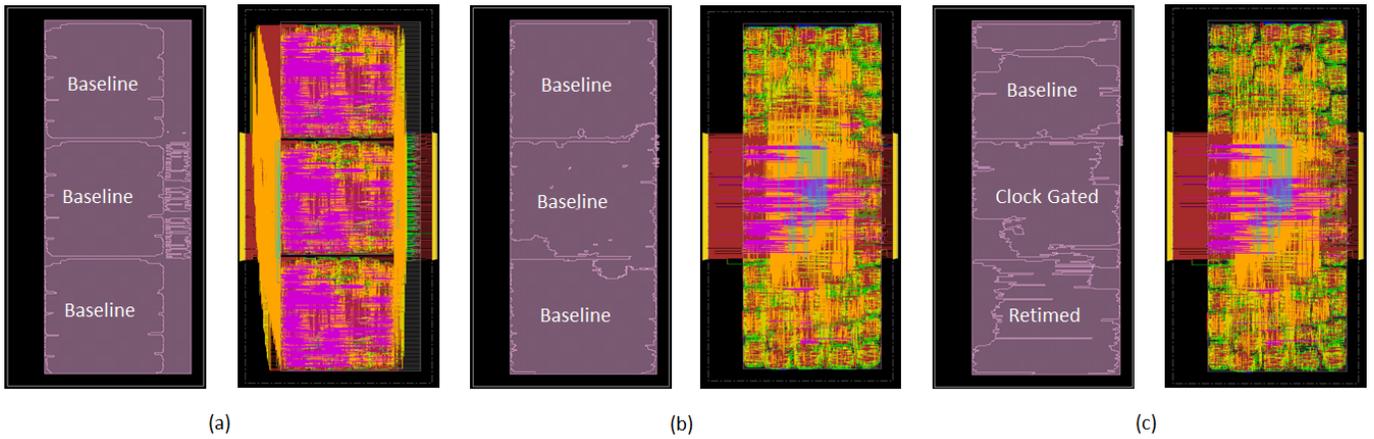


Fig. 3: Amoeba views and layouts of TMR architectures: (a) AES\_TMR\_IDE; (b) AES\_TMR\_OPT; (c) AES\_TMR\_DIF.

different, but functionally equivalent. To do so, we generated three AES crypto cores with different gate-level netlists. The first one is our baseline AES, the second one is obtained after applying the clock gating technique which is used to reduce power dissipation in parts of the circuit that are not being switched (and therefore, has an impact on SCA resiliency), and the third one is obtained after performing the retiming technique which moves the relative location of latches and registers, primarily to improve performance. In the AES\_TMR\_DIF design, the synthesis tool is also allowed to perform logic optimizations. The structure of the AES\_TMR\_DIF design is illustrated in Fig. 2.

Figure 3 presents the amoeba and physical layout views of the AES designs under the TMR scheme. Observe from Figure 3 that the AES\_TMR\_IDE design includes three identical instances of the AES design, the AES\_TMR\_OPT design has three instances of the AES design structurally very close to each other, but with different number of cells and routes, and the AES\_TMR\_DIF design includes three physically and structurally different instances of the AES design. Note that all these TMR designs have the same timing constraints, core area, and pinouts for the sake of a fair comparison.

#### IV. PROPOSED SIDE-CHANNEL POWER ANALYSIS ATTACK

The flow of our side-channel power analysis attack is illustrated in Fig. 4. Compared to the traditional IC design flow, extra steps were included to enable our attack. To cope with the exponential size of all possible keys i.e.,  $2^{128}$ , the simulation data is obtained for  $L$ -bits of the 128-bit secret key, where  $L$  is set to 8 in our experiments. In the text and results that follow, without loss of generality, we perform attacks on 8 bits of the secret key at a time. The same attack can be repeated 16 times to uncover the entire 128-bit secret key.

In our attack, initially, logic synthesis is performed on the design using the timing constraints and design library by the Cadence Genus tool and the gate-level netlist is obtained. Then, this gate-level netlist is simulated using the Cadence Xrun tool under the given test-bench. Because  $L$  is 8, this netlist is instantiated 256 times in the test-bench, i.e., one in-

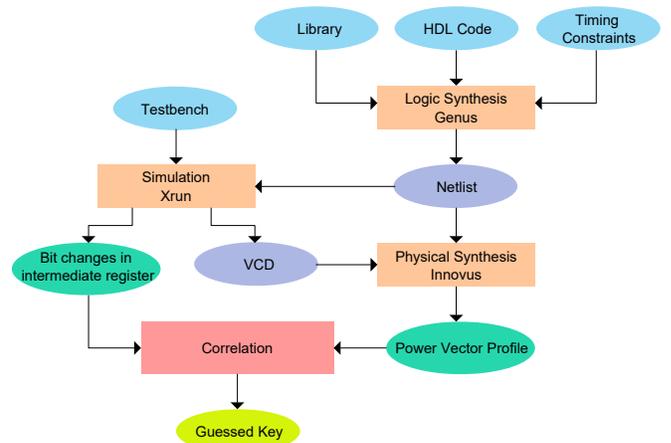


Fig. 4: Flow of the side-channel power analysis attack.

stance for each possible 8-bit key. One output of the simulation is the number of bit-changes in the intermediate register of the AES design, which stores the secret key in the first and second clock cycles as described in [15], under all possible values of the  $L$ -bit key. Note that the number of bit-changes is a high-level representation of power dissipation. Another output of the simulation is the Value Change Dump (VCD) file which annotates any changes in any signals of the design along with the time of change.

Then, the gate-level netlist is passed through the physical synthesis performed by the Cadence Innovus tool. This tool reads the VCD file and generates a vector-based dynamic power report for any time window of interest under all possible values of the  $L$ -bit key. This power estimation is a good representation of the power dissipation of the fabricated chip because it takes into account parasitic information from extraction and representative input patterns from simulation<sup>1</sup>. We obtain the *power data set* which is computed as the

<sup>1</sup>For readers with IC design background, we clarify that we utilize the Voltus power analysis engine of Innovus with VCD and Standard Delay Format (SDF) files. We ask the tool to generate a power estimation at every 1ns to oversample the 500 MHz frequency of operation of the circuit. This matches the capability of an adversary equipped with a typical oscilloscope.

TABLE I: Physical synthesis results of the single AES and its realizations using the TMR technique.

| Design      | gate  | FF   | area   | power |
|-------------|-------|------|--------|-------|
| Single AES  | 11782 | 530  | 33.63  | 9.44  |
| AES_TMR_IDE | 35919 | 1590 | 103.62 | 42.60 |
| AES_TMR_OPT | 35020 | 1590 | 103.92 | 29.09 |
| AES_TMR_DIF | 30124 | 1584 | 79.08  | 50.51 |

TABLE II: Physical synthesis results of each instance of the AES\_TMR\_DIF design.

| Instance        | gate  | FF  | area  | power |
|-----------------|-------|-----|-------|-------|
| Baseline AES    | 9826  | 530 | 25.89 | 15.84 |
| Clock Gated AES | 9853  | 530 | 25.91 | 15.12 |
| Retimed AES     | 10187 | 524 | 25.88 | 16.73 |

difference of the power dissipation values of the AES crypto core in the first and second clock cycles as described in [15]. To obtain these simulation and power data sets, 1000 randomly generated plaintexts were used.

Finally, for each possible key, the Pearson Correlation Coefficient (PCC) is computed between the simulation and power data sets, and the one that leads to the maximum PCC value is determined to be the guessed key.

In order to make the Cadence tools work in harmony, the flow illustrated in Fig. 4 is automated using Python scripting. Note that the runtime to discover the 8 bits of the secret key in the single AES design is approximately 2 hours for 1000 plaintext inputs. The majority of the runtime is spent during the generation of the power vector profile by the Cadence Innovus tool and the correlation calculation is much simpler in comparison.

## V. EXPERIMENTAL RESULTS

In this section, we first present the synthesis results of AES designs described in Section III and then, show the results of our attack introduced in Section IV on these designs.

Table I presents the physical synthesis results of the single AES design and its realizations using the TMR technique. In this table, *gate* and *FF* denote the number of gates and flip-flops, respectively and *area* and *power* stand for the total area in  $\mu m^2$  and power dissipation in mW, respectively. Note that the *area* includes all the cells and routes in the design and the clock frequency for all these designs is 500 MHz.

Observe from Table I that the realizations of the AES design using the TMR technique have around  $3\times$  larger hardware complexity than the single AES design as expected. The AES\_TMR\_IDE and AES\_TMR\_OPT designs have hardware complexity very close to each other. On the other hand, the use of clock gating and retiming techniques in the AES\_TMR\_DIF design and logic optimizations allowed in the synthesis tool lead to around 23% reduction in area with respect to other TMR realizations. However, power dissipation is increased  $1.2\times$  and  $1.7\times$  in this design with respect to the AES\_TMR\_IDE and AES\_TMR\_OPT designs, respectively.

Table II shows the physical synthesis results of each AES instance of the AES\_TMR\_DIF design. Observe from Tables I and II that all the AES instances have less complexity than the

single AES design due to the logic optimizations performed by the synthesis tool. Observe from Table II that although these instances are physically and structurally different from each other, they have similar hardware complexity in terms of area.

Our attack is run on the single AES and its realizations under the TMR scheme when 10 randomly generated 8 Most Significant Bits (MSBs) of the secret key are used. We note that in each experiment with a different secret key, our attack guessed the correct key in the single AES design and the AES\_TMR\_IDE and AES\_TMR\_OPT designs, but guessed the wrong key in the AES\_TMR\_DIF design. As an example, Fig. 5 presents the PCC value for each possible key for the 8 MSBs of the secret key which was set to 222 under all AES designs. In these figures, the red dot denotes the key guessed by the attack which has the maximum correlation value. Observe from Fig. 5 that the proposed SCA can discover the secret key in the single AES design and the AES\_TMR\_IDE and AES\_TMR\_OPT designs. The correlation value of the correct key in these designs are significantly larger than those of the wrong keys. However, our attack guesses a wrong key in the AES\_TMR\_DIF design, i.e., 10, and the correlation value of the guessed key is very close to those of other keys including the correct key. This experiment clearly indicates that the use of physically and structurally different AES designs under a TMR scheme increases the resiliency to SCAs significantly, making the attack to guess a wrong key. This is simply because of different power traces in each AES instance under the AES\_TMR\_DIF design.

Fig. 6 presents the minimum number of plaintexts required to discover the 8 MSBs of the secret key. Observe from Fig. 6 that the number of plaintexts required in the AES\_TMR\_IDE and AES\_TMR\_OPT designs is larger than the one required in the single AES design. This experiment clearly shows that the use of a TMR technique can increase the computational effort in SCAs. We note that our attack could not guess the value of 10 randomly generated secret keys of the AES\_TMR\_DIF design correctly even 2000 plaintexts were used. In this case, the run-time of our attack was almost doubled.

Finally, Fig. 7 presents the normal distribution on the minimum number of plaintexts required to discover the correct value of the 8 MSBs of the secret key obtained for successful attacks under all AES designs, except the AES\_TMR\_DIF design. Note that the dashed line points the average value of the number of plaintexts under the related AES design. Observe from Fig. 7 that while the AES\_TMR\_IDE and AES\_TMR\_OPT designs have a distribution very close to each other, their average values are larger than that of the single AES design. This experiment indicates that the use of a TMR technique can increase the number of plaintexts required to discover the secret key, increasing the computational effort in SCAs.

## VI. CONCLUSION

This paper demonstrated how a fault tolerance technique interferes with security, more precisely with the SCA resiliency, and showed how a TMR scheme with diversity can be leveraged to improve the resiliency of the design to SCAs.

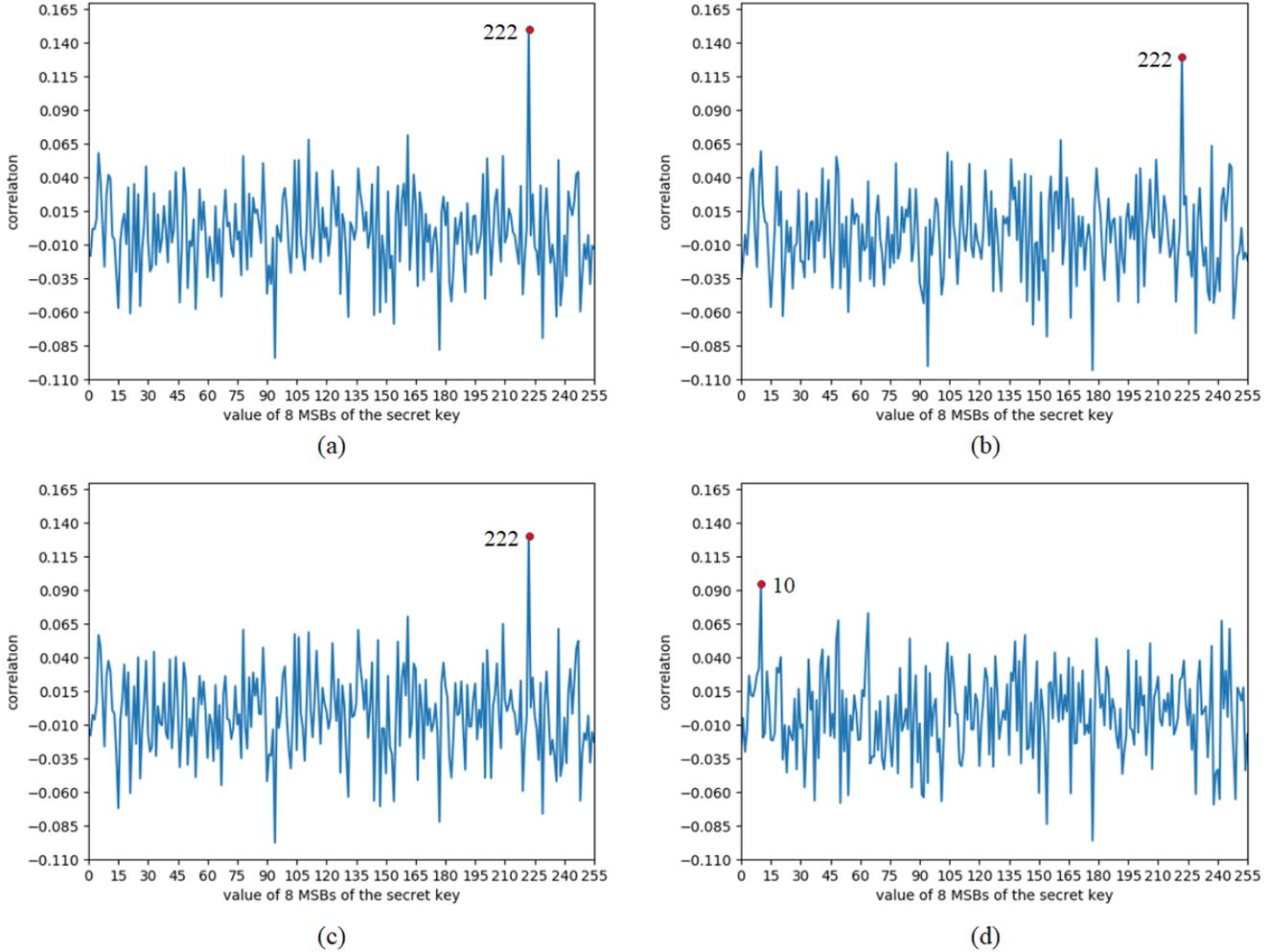


Fig. 5: Correlation between the simulation and power data sets when the 8 MSBs of the secret key was set to 222: (a) single AES, (b) AES\_TMR\_IDE, (c) AES\_TMR\_OPT, and (d) AES\_TMR\_DIF.

The experimental results pointed out that the use of a TMR technique can increase the number of plaintexts required to discover the secret key, increasing the computational effort in SCAs and the use of functionally equivalent, but physically and structurally different instances can make SCAs to guess a wrong key, increasing the resiliency of the design. As it stands, the use of reliability techniques to increase the security of a circuit is a largely unexplored territory. The possibilities for future avenues of research are plenty, including the study of redundancy schemes other than TMR and other crypto cores vulnerable to SCAs.

#### ACKNOWLEDGMENT

This work has been partially conducted in the project “ICT programme” which was supported by the European Union through the ESF. It was also partially supported by the Estonian Research Council grant MOBERC35.

#### REFERENCES

- [1] R. C. Baumann, “Radiation-Induced Soft Errors in Advanced Semiconductor Technologies,” *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 305–316, 2005.
- [2] S. Kasap, E. Weber Wächter, X. Zhai, S. Ehsan, and K. McDonald-Maier, “Survey of Soft Error Mitigation Techniques Applied to LEON3 Soft Processors on SRAM-Based FPGAs,” *IEEE Access*, vol. 8, pp. 28 646–28 658, 2020.
- [3] M. Nicolaidis, “Time Redundancy based Soft-Error Tolerance to Rescue Nanometer Technologies,” in *IEEE VLSI Test Symposium*, 1999, pp. 86–94.
- [4] S. Nascimento Pagliarini, L. Alves De Barros Naviner, and J.-F. Naviner, “Selective Hardening Methodology Concerning Multiple Faults,” in *IEEE Nuclear and Space Radiation Effects Conference*, 2012.
- [5] H. Jeon and M. Annavaram, “Warped-DMR: Light-Weight Error Detection for GPGPU,” in *IEEE/ACM International Symposium on Microarchitecture*, 2012, pp. 37–47.
- [6] F. Almeida *et al.*, “Single-Event-Induced Charge Sharing Effects in TMR with Different Levels of Granularity,” in *Radiation and Its Effects on Components and Systems (RADECS)*, 2012, pp. 1–7.
- [7] S. Pagliarini *et al.*, “Evaluating Architectural, Redundancy, and Implementation Strategies for Radiation Hardening of FinFET Integrated Circuits,” *IEEE Transactions on Nuclear Science*, pp. 1–1, 2021.

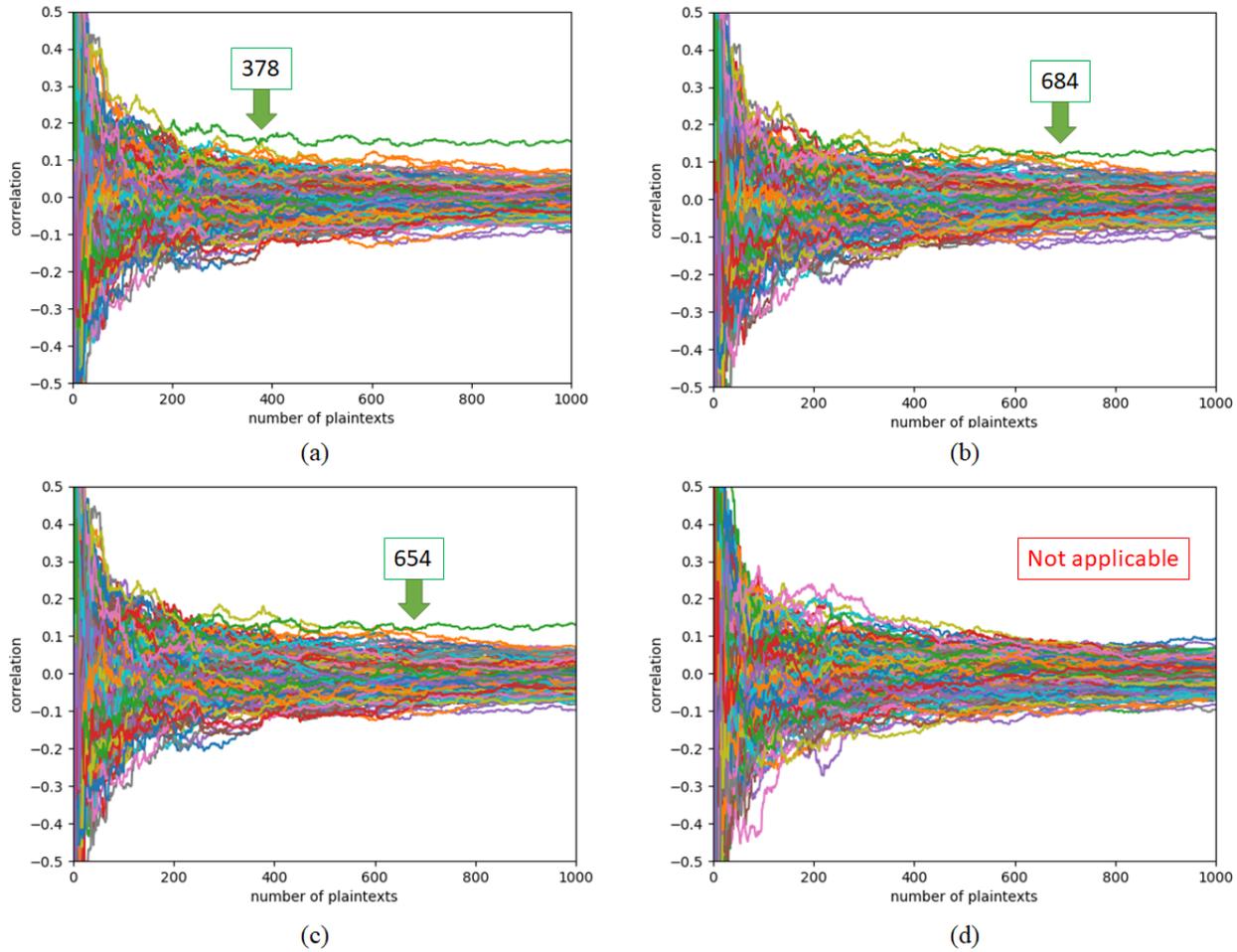


Fig. 6: Number of plaintexts necessary to discover the 8 MSBs of the secret key when they were set to 222: (a) single AES, (b) AES\_TMR\_IDE, (c) AES\_TMR\_OPT, and (d) AES\_TMR\_DIF.

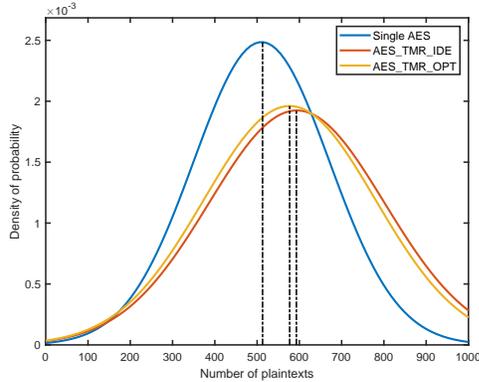


Fig. 7: Normal distribution on the number of plaintexts required to discover the secret key.

[8] S. Mitra, N. R. Saxena, and E. J. McCluskey, "A Design Diversity Metric and Reliability Analysis for Redundant Systems," in *International Test Conference*, 1999, pp. 662–671.

[9] I. A. C. Gomes *et al.*, "Methodology for Achieving Best Trade-off of Area and Fault Masking Coverage in ATMR," in *Latin American Test Workshop*, 2014, pp. 1–6.

[10] H. Li, S. Bhunia, Y. Chen, T. Vijaykumar, and K. Roy, "Deterministic

Clock Gating for Microprocessor Power Reduction," in *International Symposium on High-Performance Computer Architecture*, 2003, pp. 113–122.

[11] C. E. Leiserson and J. B. Rose, Flavio M. and Saxe, "Optimizing Synchronous Circuitry by Retiming (Preliminary Version)," in *Caltech Conference on Very Large Scale Integration*. Springer Berlin Heidelberg, 1983, pp. 87–116.

[12] F.-X. Standaert, "Introduction to Side-Channel Attacks," in *Secure integrated circuits and systems*. Springer, 2010, pp. 27–42.

[13] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer-Verlag, 2007.

[14] V. Maingot and R. Leveugle, "Influence of Error Detecting or Correcting Codes on the Sensitivity to DPA of an AES S-box," in *International Conference on Signals, Circuits and Systems*, 2009, pp. 1–5.

[15] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-Analysis Attack on an ASIC AES Implementation," in *International Conference on Information Technology: Coding and Computing*, 2004, pp. 546–552.

[16] S. Mao and L. Liu, "OPTMR: Optimal Data Flow Graph Partitioning for Triple Modular Redundancy Against Hardware Trojan in Reconfigurable Hardware," in *International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2016, pp. 68–71.

[17] N. B. Gunti and K. Lingasubramanian, "Fault Sensitive Neutralization of Hardware Trojans Using Multi-level Triple Modular Redundancy Scheme," in *International Symposium on Nanoelectronic and Information Systems (iNIS)*, 2017, pp. 105–110.

[18] "AES (Rijndael) IP Core," [https://opencores.org/projects/aes\\_core](https://opencores.org/projects/aes_core), accessed: 2020-12-15.