On the Development of a New Countermeasure Based on a Laser Attack RTL Fault Model

Charalampos Ananiadis, Athanasios Papadimitriou, David Hély, Vincent Beroulle
Univ. Grenoble Alpes, LCIS
F-26000, Valence, France
firstname.lastname@lcis.grenoble-inp.fr

Paolo Maistri, Régis Leveugle
CNRS, TIMA, F-38000 Grenoble, France
Univ. Grenoble Alpes, TIMA, F-38000 Grenoble, France
firstname.lastname@imag.fr

***Abstract*— Secure integrated circuits that implement cryptographic algorithms (e.g., AES) require protection against laser attacks. The goal of such attacks is to inject errors during the computation and then use these errors to retrieve the secret key. Laser attacks can produce single or multiple-bit errors, but have a local and usually transient impact in the circuit. In order to detect such attacks, countermeasures must take into account the circuit implementation. This paper proposes a countermeasure implemented at the Register Transfer Level (RTL) according to a previously proposed laser attack RTL fault model. Efficiency of the implemented countermeasure is evaluated on a case study in terms of area overhead, error detection rates at RTL and fault detection capabilities with respect to layout information.**

Keywords*— *Security; Integrated circuits; Laser attacks; Countermeasure; Fault injections; AES

I. INTRODUCTION

The need for security and protection of private and sensitive data in modern secure hardware devices such as smartcards, set-top boxes or mobile devices is unquestionable. Thus, in order to achieve both high levels of security and performance, cryptographic services are implemented on dedicated hardware. However, these secure circuits are the targets of attacks by hackers that aim at gaining access and exploit the secret key and protected information of the cryptographic circuit [1]. Hardware fault injection is a powerful method to attack crypto devices: in particular, laser attacks provide effective means to produce exploitable faults on the functionality of secure integrated circuits. This effectiveness is due to the precise spatial focus (with the ability to affect small parts of the circuit), the accurate synchronization, and the high occurrence probability [2]. In order to avoid exploitation of laser attacks, the detection of such attacks is essential. The faster the detection, the less exploitable information will end up to the attacker. Hence, appropriate countermeasures should be deployed, ensuring high levels of detectability against laser attacks on hardware.

Countermeasures can be developed and described at different levels of abstraction. We consider countermeasures developed at the RTL abstraction level. At this design stage, lacking of a laser fault model will inevitably lead to countermeasure reevaluation and design re-spins. Such design re-spins are feedback loops from a later to an earlier stage and can cause increase of the design costs. This paper proposes an implementation of a countermeasure applied at RTL, which appears early in the design flow, and therefore the needed feedback loops for evaluation are kept within the RTL design stage.

Countermeasures are inevitably increasing the cost mainly because of the added hardware. Protecting block ciphers is often based on redundancy, which can be classified into: hardware redundancy, temporal redundancy, and information redundancy [3]. In our work, we propose a hybrid approach that is based on both hardware and information redundancy.

This paper focuses on the implementation of a countermeasure which is based upon an existing laser attack RTL fault model. Indeed, as proposed in [4], a laser attack RTL fault model permits modeling the faults originating from local laser attacks, either in combinational or sequential instances. This fault model allows the designers to perform early evaluations of their implementations with faults more representative of laser attacks. The fault model predicts the effects of a laser attack by extracting the flip flops (FFs) of the design which can be the concurrent recipients of faults.

The advantage of our countermeasure is that it can be applied to different secure circuits and provides a configurable and adaptable solution. The goal of this paper is to present a countermeasure that can provide high levels of detectability against laser attacks by achieving 100% detection rate according to a proposed laser attack RTL fault model. Furthermore, the goal is to provide a generic, effective and simple countermeasure at the RTL level that uses a hybrid approach of redundancy. Finally, we show that the countermeasure overhead in terms of area presents an acceptable cost.

The paper is organized as follows. Section II summarizes existing countermeasures. Section III presents two fault models this work bases its analysis on. Section IV is dedicated to the description of the implementation of our countermeasure. Section V describes a case study for an AES cryptographic circuit. Section VI comprehensively presents the results about area overhead, fault injection error rates and detection capabilities, with respect to the layout, achieved by the protected AES circuit.

II. COUNTERMEASURES AGAINST FAULT ATTACKS

Mostly, countermeasures that can be developed at RTL are redundancy based countermeasures. Such countermeasures can be categorized in: hardware redundancy, temporal redundancy and information redundancy. In the literature there exist countermeasures that are implemented at RTL for the

protection against fault attacks but not specialized for laser fault attacks.

The hardware redundancy countermeasures are mainly implemented by replicating functional blocks of the circuit [5]. The results these separate blocks are producing are compared and if a difference is found that means that an error occurred in at least one of them. The drawback of this method is the large area overhead impact due to the replication.

The temporal redundancy countermeasures actually use the same hardware repetitively in order to compute the same process at least twice [6]. They can detect errors with minimum area penalty but this method entails 100% time overhead, and thus a big reduction in performance.

The information redundancy countermeasures are based on error detecting codes. An error detecting code is made of check bits generated from the information under protection, propagated through computation, and validated at the end. One suitable solution, special for ciphers such as AES, is parity [7]. Although the parity code is simple and cheap, the main disadvantage is that it cannot detect any even number of injected faults. Apart from parity, other solutions have been proposed that are more effective but entail bigger area overhead [8] [9]. According to [10], nonlinear robust codes provide theoretical guaranteed detection probability for each error. In [11] the authors provide results, which show that these (theoretical) detection probabilities depend also on structural properties of the protected circuits and therefore are not always valid. Especially in [11], the authors have used gate hardening to strengthen nonlinear robust codes against laser fault attacks.

III. Fault Models

Many fault injection models have been developed in order to predict errors generated by actual fault attacks in a circuit. To evaluate a circuit under errors caused by a laser attack, two fault models are taken into account and are presented next.

A. Laser Attack RTL Fault Model

In [4] a laser attack RTL fault model is presented that models the effects of laser attacks at RTL. A cone partitioning of the design's RTL elaborated netlist is proposed for the development of a fault model more representative of laser attacks than already existing fault models. The cone fault model reduces significantly the fault space if compared with the random fault model [12], and provides a more accurate modeling due to taking into account functional dependencies between logic cones. More specifically, a logic cone consists of the nets, combinational instances, and primary inputs, that belong to the input logic that drives the sequential elements such as FFs. Logic cones are bounded by a starting net (father) and expand backwards, from the outputs towards the inputs, up to either FFs or primary inputs of the circuit.

After logic cone extraction on the elaborated netlist of the RTL description, cones are processed by means of an intersection function and cones that share common elements, either combinational instances or primary inputs, end up in the same set. Figure 2 shows an example where logic cones 3 and 4 are intersecting because of sharing instance i9, while cones 1, 2 and 3 are non-intersecting. Each set is composed of the FFs having their logic cones intersecting. Thus, the cone analysis provides sets of FFs that can be concurrent targets of a laser attack.

Depending on the size of the design, the content of a set may start from one FF to few hundreds of FFs. Thus, in order to perform injection campaigns, a huge number of all possible combinations of FFs from sets need to be checked. Although this fault model reduces the fault space, sampling of these sets is still necessary. Fault samples are obtained from sampling the FF sets, with respect to multiplicity of samples and acceptable margin of error as described in [13].

B. Layout Fault Model

The layout fault model predicts the effects of laser attacks in a circuit using the layout information [14]. Depending on the laser spot size, for each fault attack, a set of cells of the circuit are simultaneously affected. The first step of this analysis includes a spot partitioning of the final layout of the circuit in order to correlate each spot with the corresponding affected instances and finally the affected FFs. After the place-and-route step, the required information about the layout of the design has been determined. Depending on the size of the spot examined each time, the design is partitioned: each spot represents a possible laser attack on the design. For every spot, the corresponding cells, included in the spot region, are obtained. At the end, the affected instances per spot are verified and the spot itself is characterized as detected or undetected.

IV. A Countermeasure Against Laser Fault Attacks

The implementation architecture of our countermeasure is based on a hybrid approach mixing spatial hardware redundancy and information redundancy. More specifically, the information which has to be protected is separated in groups and one parity bit is calculated for each group. To maximize the effectiveness of such a protection countermeasure, at most one fault must be concurrently injected in each group. In the same sense a concurrent error detection scheme presented in [15] for the detection of single stuck-at faults, but not specialized for laser attacks, proposes parity groups that will contain only one fault. In our work in order to achieve at most one fault on each parity group we use the results of the analysis of the design according to the types of concurrent faults predicted by the laser attack RTL fault model.

Figure 1 shows the principle of group parity generation with prediction [16] as a concurrent error detection scheme on which our countermeasure is based. A simple parity code is used as information redundancy, and the duplication of functional blocks for the prediction of parity is used as hardware redundancy. This countermeasure aims at protecting both the combinational and sequential instances of the design, which can be achieved due to the protection capabilities of group parity generation with prediction. The proposed countermeasure is based on three procedural steps, first step is the calculation of parity from the bits of the original design, second step is the prediction of calculated parity from the

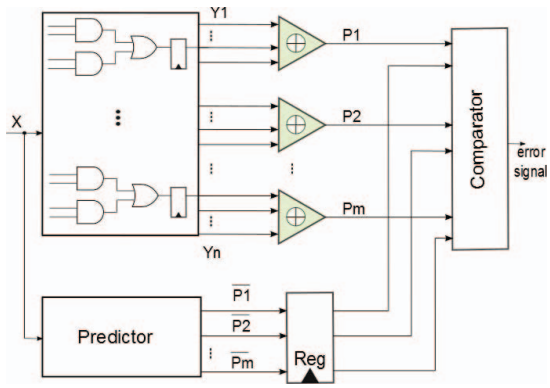

Figure 1. General countermeasure architecture: (1) Pi parity computed from the FFs of the original design (2) $\overline{P_i}$ parity predicted from inputs x (3) comparison of each parity Pi and $\overline{P_i}$

duplicated components, and the final step is the comparison of these parities. A mismatch in calculated and predicted parity means that an error occurred in either the combinational or sequential instances. The comparator is indicating if an error occurred by outputting an error signal.

The proposed countermeasure is based on parity code. Parity bits are calculated from fixed predetermined vectors of FF bits Yi. Every vector of FF bits forms a parity group. Depending on the size of the vector, the protection that parity provides against laser attacks is characterized either robust or less robust. For example, one parity bit calculated for an 8-bit vector is less capable to detect a laser attack than one parity bit for a 4-bit vector as it will be explained later.

The functional blocks that we want to protect are duplicated. These duplicated blocks constitute the predictor in our countermeasure (Fig. 1). During synthesis, the predictor and the original components are optimized by the EDA tools as much as possible so as to output only the parity bits. The gain in the overheads presented later is mainly due to this strategy. Secondly, in order to further reduce the area overhead imposed by duplication, a technique as in [17] is used. This technique proposes the reduction of duplicated registers. In the duplicated components the n-bit registers of the original design that are used to save information are replaced with 1-bit registers to save the predicted parities.

Every calculated and predicted parity bit ends up in the comparator. The comparator consists of checkers that implement the comparison between the parity bits. For each parity pair, predicted and calculated, there exists a checker that outputs an error bit signal which indicates if an error occurred or not in this specific combination. Consequently, the final sequence of error bits forms the output error signal vector.

The size and number of parity groups is fundamental for the protection of circuit against laser attacks and is dependent on three factors. These three factors answer to three specific questions. The first question is how are the bits selected to be combined together in parity groups efficiently? The second question is how many bits are going to be in each group? The third question is which bits are possible to be included in the same parity group? These three factors are respectively answered by the following:

1) laser attack RTL fault model
2) robustness vs area tradeoff
3) architectural characteristics of the circuit in conjunction with the laser attack RTL fault model

The first answer is based on the laser attack RTL fault model analysis. An RTL tool that implements the laser attack RTL fault model provides the needed information about the possible independent combinations of bits in parity groups. The laser attack RTL fault model described in the previous section identifies the structural dependencies among all the logic cones. Independent cones are those that do not share any common elements; on the other hand, dependent cones are those that share at least one common element or primary input. The possible candidates for the same parity group are these independent cones. The utilized laser attack RTL fault model dictates that any laser attack will be confined inside a single logic cone: then, if each parity group of our countermeasure contains the outputs of independent logic cones, this will lead to a theoretical detection rate of 100%. As can be seen in Fig. 2, cones 1, 2 and 3 are independent and thus the output bits of these cones can be combined in a parity group. On the other hand, cones 3 and 4 are dependent and thus they cannot exist in the same parity group. The logic behind this reasoning is that cones that are independent in RTL design are more likely those that will be further away from each other after place-and-route compared to dependent cones. Thus, in case of a laser attack it is expected that it will be more unlikely to attack simultaneously independent cones than dependent ones, and a single laser shot will not succeed in injecting more than one fault into each group. In the end, the actual possible candidates for grouping are the output bits of FFs whose input logic belongs to independent cones. Although logic cones in RTL may differ from cones after synthesis, the method has already been validated in [14].

The second factor that influences the implementation of the countermeasure is the effectiveness of the protection scheme, which depends on the size of the parity group. The bigger the group is, the more bits are included and concurrently protected in the same parity group. The smaller the group is, the fewer bits are included in the same parity group. Moreover, we know that the parity code cannot detect an even number of faults, occurring in the same protected vector. Hence, the bigger the

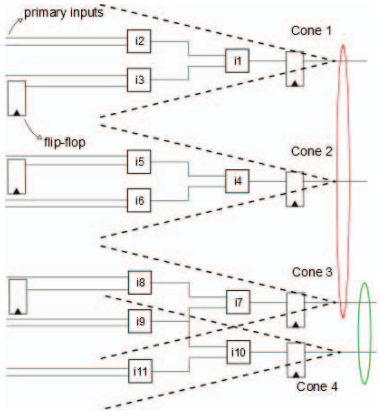

Figure 2. Cone partitioning and parity grouping example

parity group is, the bigger the possibility of an even number of faults to be injected in the same parity group. This analysis, finally, ends up as a tradeoff between protection and area overhead. A representative example is going to be presented in the next section.

The third factor that affects the implementation of our countermeasure is the architectural characteristics of the design. The combination of bits in each parity group is determined by the structure of the circuit. The goal is to include all the possible candidate bits in parity groups that will minimize the design time.

The proposed countermeasure is based on a simple and well known architecture, and it can protect both the combinational and sequential logic of the design. Finally, different versions of the countermeasure may be designed, depending on the tradeoffs between protection and area overhead. These aspects are going to be further analyzed in the following sections by applying the methodology on a practical AES description.

V. Case Study

In this section, we describe the procedure to implement our countermeasure on a secure circuit. The basic AES implementation, which is described in [18], is used without its countermeasures. In particular, the design has the advantage of a symmetric implementation by dividing the default 128-bit block in 16 blocks of 8-bits. The AES design consists of three basic components, which are: Data Unit (DU) (the encryption data path), Key Unit (KU) (round key generation) and Control Unit (CU). In our case study, we applied the proposed countermeasure to the DU and the KU components. The DU component consists of 16 blocks and each block operates on an 8-bit vector. The KU component is responsible for the secret key and works on a 128-bit vector which is the size of the key (also keys of 192-bits and 256-bits are allowed).

A cone analysis with the RTL tool that implements the laser attack RTL fault model indicates the possible combination of bits to include in parity groups. As can be seen in Fig. 3, the DU component consists of four rows and each row represents a 32-bit block. On each row there exist four identical 8-bit blocks that implement the main operations of AES. Each ellipsis in the diagonals is made of two cells and indicates different parity groups at the bit level, resulting from the RTL tool analysis. As an example: a parity group consists of the first bit of the first cell of the first row (block A0) that is combined with the first bit of the second cell of the second row (block B1) and so on, until the eighth bit which is the last parity group of this ellipsis. Thus, each ellipsis also shows the combination of bits that form the parity groups.

The main countermeasure developed includes the full protection of DU and KU. However, in the literature there are no countermeasures specially developed for the protection against laser attacks. So in order to evaluate the efficiency of our countermeasure and to emphasize the tradeoff opportunities a second type of countermeasure has been developed. This second type of countermeasure is considered as a classical approach and aims at protecting groups of 8-bits from the same register without taking into account

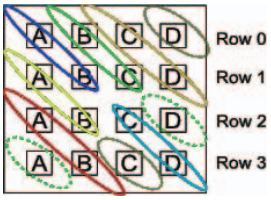

Figure 3. Example of parity groups on AES circuit

the dependencies of the logic cones. The implemented countermeasures hence are:

- Protection of DU and KU based on the laser attack RTL fault model with parity groups made from independent cones (FM)
- Protection of DU and KU, with parity groups made from dependent cones not using the laser attack RTL fault model (NFM)

VI. Analysis of Countermeasures

Firstly, we evaluate the countermeasures in terms of area overhead, with respect to the original AES design. The countermeasures have been synthesized in NANGATES 45nm open technology using Synopsys Design Compiler. The first synthesis strategy uses the simple compilation options (SC), while the second one uses the ultra compilation options (UC). Synthesis results can be seen in Table I. This table shows that with the simple compilation option the FM countermeasure has approximately the same area overhead as the NFM countermeasure. On the other hand, with the ultra compilation option, a difference in the area overhead is observed. In fact, the NFM countermeasure is better optimized than the FM.

Secondly, we analyze the countermeasures with respect to two evaluation criteria. The first is based on RTL injection campaigns using the laser attack RTL fault model. These campaigns validate the expected theoretical 100% detection rate to be achieved by the proposed FM countermeasure. Also, it shows that the two countermeasures achieve different error rates. The second evaluation criterion aims at practically evaluating the efficiency of the two countermeasures using layout information. This more representative detection capability analysis, based on the layout fault model, shows the efficacy of the laser attack RTL fault model to predict layout information which leads to a more efficient countermeasure.
Fault injection campaigns at RTL aim at simulating the effects of laser shots on the design according to the laser attack RTL fault model described in [4]. The fault samples used for these campaigns are randomly generated with respect to a given

Table I. Synthesis results

Design	Synthesis	
	Area (μm^2)	Overhead (%)
FM (SC)	28133	79.9
NFM (SC)	27870	78.3
FM (UC)	21191	63
NFM (UC)	19635	51

Table II. Fault injection error rates for laser attack RTL fault model. M2-M10 denotes fault multiplicity.

Fault injection error rates		M2	M3	M4	M5	M6	M7	M8	M9	M10
FM	Detected	18.4	24	28.3	31.6	34.1	36.7	38.2	39.9	40.6
	Undetected	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	Silent	1.5	0.5	0.3	0.4	0.3	0.3	0.1	0	0
	False positive	79.9	75.4	71.3	67.9	65.5	62.9	61.6	60	59.3
	Final detection rate	**98.3**	**99.4**	**99.6**	**99.5**	**99.6**	**99.6**	**99.8**	**99.9**	**99.9**
NFM	Detected	19.9	28.2	30	35.1	35.2	39.1	38.3	41.8	40.8
	Undetected	2.9	0.2	2	0.1	1.7	0.1	1.3	0.1	1.4
	Silent	9.2	1.1	4.9	0.4	4.4	0.3	4.5	0.8	4.2
	False positive	68	70.5	63.1	64.4	58.7	60.5	55.9	57.3	53.7
	Final detection rate	**87.9**	**98.7**	**93.1**	**99.5**	**93.9**	**99.6**	**94.2**	**99.1**	**94.5**

multiplicity and margin of error, within each extracted set of FFs [13]. The final error rates are derived by comparing the outputs of the design with the golden values which are the fault free results. Table II includes the error rates achieved by the countermeasures for laser attack RTL fault model. For the injection campaigns a fault multiplicity of 2 up to 10 has been chosen, per fault sample (M2-M10). For each multiplicity, the number of samples has been chosen, so as to achieve a worst case margin of error of 10%. In our case, the margin of error depends on the number of samples generated per set of FFs [13]. The more samples generated per set of FFs the smaller the margin of error is and vice versa. The categories of faults are the following:

- Detected: the faulty result is identified
- Undetected: the result is faulty, but no error detected
- Silent: the result is unchanged, and no error is shown
- False-positive: detection occurs, but no error in the result

Table II shows the final detection rates achieved and represents the detection capability of each countermeasure at RTL. The final detection rate is calculated by adding the detected and the false positive rate. The results show that the detection capability of the proposed countermeasure FM is very close to the theoretical detection rate of 100%. However, there are undetected faults in FM due to the faults injected in the Control Unit, which is unprotected. Also, we observe that the NFM countermeasure has low detection capability for the even multiplicities. In particular, the small differences in the final detection rates between the countermeasures highlight the benefits of FM over NFM. We observe that the FM countermeasure has better detection rates in the even multiplicities, while in the odd multiplicities the NFM countermeasure presents also high detection rates. So a more representative analysis, taking into account the circuit layout, will help us evaluate deeply the countermeasures, and thus the physical locality of laser attacks.

The evaluation of the countermeasures in the layout, in terms of fault detection percentages, is based on the layout fault model. The analysis of the countermeasure's robustness with the layout fault model requires first to build a model of each countermeasure. Each countermeasure model consists of the names of FFs that form the parity groups and the FF that saves the predicted parity. Every parity group is paired with the corresponding FF that saves the predicted parity. After that, the layout fault model, obtained by partitioning the design in spots, provides two cases of affected instances: first case is when the FFs are directly affected by a laser attack, and the second case is when both combinational instances and FFs are affected by a laser attack. The second case is further analyzed in order to find potential propagated faults in the FFs that are driven from the combinational instances affected. These two cases are:

1. Direct FF attacks (Multiple Event Upset, MEU)
2. Direct FF & Combinational attacks (Multiple Event Upset & Multiple Event Transient, MEU & MET)

According to the layout validation approach of [14], the layout of the design is partitioned in spots. Then for each spot the FFs which may capture faults of either combinational or sequential origin are extracted. These FFs are all potential candidates for multiple fault injections. At first we check if the FFs of a spot are all located in different parity groups. In this case any combination of faults originating from this spot will be detected since we will have at most one fault in each corresponding parity group. The spots which do not fulfill these criteria are further examined per multiplicity. Since all possible combinations for these cases generate a huge fault list, we apply a statistical approach [13]. For each spot we fix the maximum multiplicity of concurrent errors to ten. For every spot and multiplicity from 2 up to 10 (since we care for multiple faults, we ignore multiplicities of 1) we choose a number of samples which will lead to a margin of error of 5%.

Here we have to distinguish when a sample generated from a spot is considered as undetected and when as detected. The first undetected case is if an even number of FFs obtained from the examined sample is found in the same parity group. This means there is an even number of faults in the same parity group, which is undetectable with the parity code. The second undetected case is if an odd number of FFs is found in the same parity group but at the same time the corresponding FF that saves the predicted parity is affected. Any other case is considered as detected, and these are: if there are only an odd number of FFs affected in the same parity group or if only the predictor's FF is affected. We must stress here that for this

Table III. Fault detection percentages for layout fault model.

Design	Spot 1µm		Spot 5µm	
	MEU	MEU & MET	MEU	MEU & MET
FM (SC)	99.6	99.1	98.8	98
NFM (SC)	86.6	66.6	68.6	72.2
FM (UC)	98.3	88	94.7	93.4
NFM (UC)	84.4	57.3	68.8	73

analysis if at least one parity group leads to detection of a fault, then this attack sample is considered as detected.

Table III presents the detection rates achieved for each countermeasure, i.e., the percentage of detected spots on each design, for the layout fault model and for two different spot sizes (size of the spot is the diameter of the laser beam). The table shows the detection capability of the countermeasures after layout. Firstly, the theoretical expectation of 100% detection rate for the FM countermeasure is closely confirmed. In fact, the proposed countermeasure is achieving very high detection rates, close to 100%, for the simple compilation option, and high detection rates for the ultra compilation option. However, while the ultra compilation option reduces significantly the area overhead, it does not reduce the detection rates proportionally. Although the countermeasures have approximately the same area overhead, the FM countermeasure achieves higher detection rates. In particular, the FM countermeasure preserves the high detection rates for both the affected cases (MEU, MEU & MET) while the NFM countermeasure has a big reduction in the detection rate between the two affected cases. This is mainly because the parity grouping strategy of our countermeasure takes into consideration the combinational logic of the circuit with the assistance of the laser attack RTL fault model.

VII. Conclusion

This paper presents a simple and adaptable countermeasure for the protection of secure circuits, especially developed to protect both the combinational and sequential instances of the circuit. A previously proposed laser attack RTL fault model has assisted in the implementation process and early evaluation of the countermeasure. The novelty our countermeasure is introducing, consists in providing an approach on how parity groups should be constructed in order to increase the detectability rates against laser attacks considering a dedicated high level laser attack RTL fault model. The analysis results show that the proposed countermeasure can achieve higher fault detection rates than a similar countermeasure with approximately the same area overhead.

Acknowledgment

We would like to thank Verific Design Automation Inc. for providing the SystemVerilog and VHDL front-end used for the implementation of our RTL methodology.

This work has been supported by the French National Research Agency project “LIESSE” [ANR-2012-INSE-0008]. TIMA is Partner of the Labex PERSYVAL Lab (ANR-11-LABX-0025).

References

[1] Boneh, D., DeMillo, R. A., & Lipton, R. J. (2001). On the importance of eliminating errors in cryptographic computations. *Journal of cryptology*, *14*(2), 101-119.

[2] Van Woudenberg, J. G., Witteman, M. F., & Menarini, F. (2011, September). Practical optical fault injection on secure microcontrollers. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on* (pp. 91-99). IEEE.

[3] Barenghi, A., Breveglieri, L., Koren, I., & Naccache, D. (2012). Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proceedings of the IEEE*, *100*(11), 3056-3076.

[4] Papadimitriou, A., Hély, D., Beroulle, V., Maistri, P., & Leveugle, R. (2014, March). A multiple fault injection methodology based on cone partitioning towards RTL modeling of laser attacks. In *Proceedings of the conference on Design, Automation & Test in Europe* (p. 206). European Design and Automation Association.

[5] Di Natale, G., Doulcier, M., Flottes, M. L., & Rouzeyre, B. (2009). A reliable architecture for parallel implementations of the advanced encryption standard. *Journal of Electronic Testing*, *25*(4-5), 269-278.

[6] Rajendran, J., Borad, H., Mantravadi, S., & Karri, R. (2010, June). SLICED: Slide-based concurrent error detection technique for symmetric block ciphers. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on* (pp. 70-75). IEEE.

[7] Bertoni, G., Breveglieri, L., Koren, I., Maistri, P., & Piuri, V. (2003). Error analysis and detection procedures for a hardware implementation of the advanced encryption standard. *Computers, IEEE Transactions on*, *52*(4), 492-505.

[8] Yen, C. H., & Wu, B. F. (2006). Simple error detection methods for hardware implementation of advanced encryption standard. *Computers, IEEE Transactions on*, *55*(6), 720-731.

[9] Karpovsky, M., Kulikowski, K. J., & Taubin, A. (2004, June). Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard. In *Dependable Systems and Networks, 2004 International Conference on* (pp. 93-101). IEEE.

[10] Karpovsky, M. G., Kulikowski, K. J., & Wang, Z. (2007). Robust error detection in communication and computational channels. In *Spectral Methods and Multirate Signal Processing. SMMSP'2007. 2007 International Workshop on.*

[11] Tomashevich, V., Srinivasan, S., Foerg, F., & Polian, I. (2012, June). Cross-level protection of circuits against faults and malicious attacks. In *On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International* (pp. 150-155). IEEE.

[12] Vanhauwaert, P., Maistri, P., Leveugle, R., Papadimitriou, A., Hely, D., & Beroulle, V. (2014, May). On error models for RTL security evaluations. In *Design & Technology of Integrated Systems In Nanoscale Era (DTIS), 2014 9th IEEE International Conference On* (pp. 1-6). IEEE.

[13] Leveugle, R., Calvez, A., Maistri, P., & Vanhauwaert, P. (2009, April). Statistical fault injection: quantified error and confidence. In *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.* (pp. 502-506). IEEE.

[14] Papadimitriou, A., Tampas, M., Hely, D., Beroulle, V., Maistri, P., & Leveugle, R. (2015, May). Validation of RTL laser fault injection model with respect to layout information. In *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on* (pp. 78-81).

[15] Touba, N., & McCluskey, E. J. (1997). Logic synthesis of multilevel circuits with concurrent error detection. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, *16*(7), 783-789.

[16] Sogomonyan, E. S., & Goessel, M. (1993). Design of self-testing and on-line fault detection combinational circuits with weakly independent outputs. *Journal of Electronic Testing*, *4*(3), 267-281.

[17] Pistoulet, P. (2008). *U.S. Patent No. 7,428,694*. Washington, DC: U.S. Patent and Trademark Office.

[18] Bertoni, G., Breveglieri, L., Koren, I., & Maistri, P. (2004, October). An efficient hardware-based fault diagnosis scheme for AES: performances and cost. In*Defect and Fault Tolerance in VLSI Systems, 2004. DFT 2004. Proceedings. 19th IEEE International Symposium on* (pp. 130-138). IEEE.