A Multiple Fault Injection Methodology based on Cone Partitioning towards RTL Modeling of Laser Attacks

Athanasios Papadimitriou, David Hély,
Vincent Beroulle
LCIS - Grenoble Institute of Technology
Valence, France
firstname.lastname@lcis.grenoble-inp.fr

Paolo Maistri, Régis Leveugle
Université Grenoble Alpes, TIMA Laboratory
CNRS, TIMA Laboratory
F-38031, Grenoble, France
firstname.lastname@imag.fr

***Abstract*— Laser attacks, especially on circuits manufactured with recent deep submicron semiconductor technologies, pose a threat to secure integrated circuits due to the multiplicity of errors induced by a single attack. An efficient way to neutralize such effects is the design of appropriate countermeasures, according to the circuit implementation and characteristics. Therefore tools which allow the early evaluation of security implementations are necessary. Our efforts involve the development of an RTL fault injection approach more representative of laser attacks than random multi-bit fault injections and the utilization and evolution of state of the art emulation techniques to reduce the duration of the fault injection campaigns. This will ultimately lead to the design and validation of new countermeasures against laser attacks, on ASICs implementing cryptographic algorithms.**

***Keywords*— Security; Integrated circuits; Laser attacks modeling; Fault injections**

I. Introduction

Hardware errors pose a threat to cryptographic circuit implementations, as described by [1]. Lasers provide a very effective means to perform fault injection attacks on Integrated Circuits, mainly because of their high precision locality, accurate timing and high occurrence probability [2]. Therefore, proper countermeasures have to be employed to secure cryptographic circuits from such attacks, by not allowing the exposure of critical information to the attacker. These countermeasures, and then the fault injection analysis of the circuit, must take into account the context of an attack. This includes the capabilities of an attacker with a given laser equipment, as well as the added overhead, both temporally and spatially [3]. A laser specific fault injection model at the Register Transfer Level (RTL) level of abstraction can enhance the design flow with the capability to perform early fault injection campaigns, and to avoid costly feedback runs. The design and integration of efficient countermeasures in a cryptographic circuit against an attack highly depends on the methods available for early validation in the design stage. Register Transfer Level analysis tools can provide the means to efficiently expose vulnerabilities of security oriented circuit designs, and at the same time assist the implementation of both defensive and preventive mechanisms.

Since the early days of integrated circuit manufacturing, several fault models have been used to describe faults, originating either from fabrication issues or from upsets caused by the interaction of high energy particles with an integrated circuit [4]. For particle fault analysis, classical approaches include both the utilization of the stuck-at and Single or Multiple Bit-Flipping models. An inherent disadvantage single bit-flip and stuck-at models possess, is that they were defined to describe single faults occurring in older technologies. In these technology nodes, the size of the transistors in combination with high energy particles, as a disturbance source, allowed these fault models to describe accurately these types of interactions.

The previously mentioned models, if defined with the capability of being transient, can be used to model the effects of a laser on an integrated circuit [5]. In the case of faults caused by a laser and especially in the security context, the fault analysis should deal efficiently with the added complexity imposed by the laser characteristics and with the purpose of fault injection, which is the intention to extract hidden information. The complexity rises from the fact that a laser attack, especially in recent manufacturing technologies, provides to the attacker the flexibility of an excellent controllability over location and timing. Even a minimum spot size of the order of 1μm would affect several elements. Single bit flipping in registers does not describe the phenomenon accurately and multiple bit flipping fault models have to be used [6]. Additionally, the use of laser technologies that produce pulses with low jitter and high repetition rate influences the time domain aspects of an attack.

As far as we know, in the literature there exists no comprehensive RTL Laser Fault Model. In multiple different approaches generic fault injection platforms are used, as in [15], [7], with the capability to introduce multiple faults, either by simulation or emulation, but without any correlation with the capabilities of a laser. Fault modeling at RT Level has the benefits of occurring early in the design flow and of accelerating the analysis with respect to gate level models. Besides these advantages, it has the disadvantage that optimizations and technology mapping taking place in later steps of the synthesis flow, as well as placement, are unknown at this level of abstraction. Therefore, the registers and the important nodes of a design, for which we know in advance that they will not be affected by the synthesis flow, play a crucial role in the analysis. In this way, the transient multi bit stuck-at and bit-flip fault models can be used as the basic elements to imple-

978-3-9815370-2-4/DATE14/©2014 EDAA

ment a model of the effects of a laser in both the combinational and sequential parts of a circuit [7]. On the other hand, the complexity of such a fault injection campaign under exhaustive analyses can create an enormous fault space. The fault space derived by such an approach may lead to impractical computational durations, which make the simplification of the models a necessary step, in order for simulation or emulation to be completed within reasonable time. Furthermore, a large percentage of these faults will not correspond to possible fault attacks.

The main goal of this work is to define a methodology for multiple fault injection, which would reduce the fault space of laser fault injection campaigns. This can be accomplished by using the locality characteristic of laser fault, and through a partitioning of the RTL description of the circuit. Thus, our efforts involve the development of an RTL fault injection approach more representative of laser attacks than random multi bits fault injection.

The next sections of this article are organized as follows. In Section II, an RTL cone partitioning methodology is presented in order to assist the implementation of a laser fault model, based on multiple bit flipping of the registers of a design. In Section III, we describe the implemented algorithms by applying them to a simple example, as well as to a sub-circuit of an implementation of the Advanced Encryption Standard. Section IV includes the conclusions of the article and describes future steps of our work in progress.

II. Cone Partitioning for Fault Injection

Faults that occur in a circuit impacted by a localized radiation source (either high energy particles or a laser) can be categorized in two groups. The first group contains faults directly injected into one or more flip flops, when the radiation deposits enough energy to flip their contents; the second group includes faults which occur in the combinational part of the circuit. The latter will either be functionally propagated and finally stored to the registers that are connected to the fanout network of the affected elements, or they will not affect the operation of the circuit. This can be formalized by stating that the targeted combinational element belongs to the input logic cone of the affected flip flop.

This fact can enable us to model both types of faults with multiple bit flipping in the registers of an RTL design, in a deterministic way. Therefore by injecting bit flips in one or more specific flip flops, the evaluation can cover all the faults that affect any combination of cells belonging to the corresponding cones. As an alternative to exhaustive or statistical fault injection, this approach relies heavily on efficiently selecting the flip flops for the fault injection campaign.

Another advantage that our approach offers is the capability to model the spatial characteristics of the attack in relation to the controllability over the location of fault injection. This property of the model will aid to define a measure of how successful an attack can be in terms of the controllability over space. As an example, defocusing the spot to illuminate a large area of the circuit would include in the analysis all the registers of the design in this region. By doing so, the attacker would influence a greater number of registers and RTL elements, but would also loose the capability to inject a specific type of fault.

In this article we present a logic cone partitioning methodology, as a strategy towards the development of an accurate laser fault model, and a time-efficient fault injection platform. Logic cone partitioning at gate level has been used in the past in order to perform automated fault diagnosis and locate the origin of one or multiple faults, given some faulty outputs [12], [13]. In [14] the authors use gate level logic cone partitioning as part of a Failure Mode and Effects Analysis methodology.

We define a logic cone as the set of all the nets, combinational instances, and primary inputs that reside in the transitive fan-in, of the input net of a flip flop [8]. Fig. 1 illustrates that the extraction of the logic cones of a design can enable us to determine the effect of fault injection over a particular section of the RTL design. This holds under the assumption that a functional relation continues to exist between the elements or operators, contained in each cone, even after placement and routing of the design. With this assumption in mind (which will be thoroughly investigated, as described in the last section or this article), we can identify and mark for bit-flipping analysis just the flip-flops that are influenced by an attack. On the contrary, flip-flops that reside outside the affected area do not need to be injected with faults, unless their logic cones contain elements which also exist in the cones which are considered affected by the laser.

In our analysis, we consider cones that are bounded by a starting net (father) and expand backwards, from the outputs towards the inputs, up to either flip flops or primary inputs of the circuit. In Fig. 1 we can see a simplified partitioning of an abstract circuit in logic cones, as well as the flip flops that exist in the outputs of cones A, B, and C. The idea behind our reasoning is that if a fault is injected into one or more elements that belong in Logic cone A, then we can assume that the result of that fault will potentially be stored into either flip flop A or flip flop B, or both, but it will not be stored into flip flop C, as cone C does not intersect with cone A. If the affected elements are located strictly into cone A, and not into the intersection of cones A and B, then the fault may potentially be stored only into flip flop A.

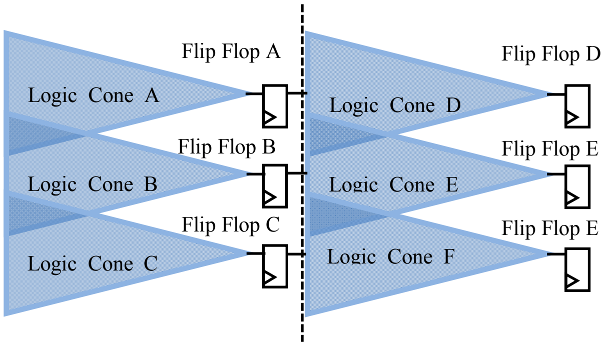

Figure 1. Logic cone partitioning

Assuming only one or more elements in cone A are affected, then the fault will be potentially recorded into flip flop A and/or B but multiple faults affecting simultaneously flip-flops A, B, and C or flip-flops A and C are not relevant since cones A, B and C do not have a common intersection. By combining the locality of a laser attack with the partitioning, we will be able to derive which flip flops will be potentially affected by a given attack. Then, the multiple fault injection can be limited to this set of flip flops that will be smaller with respect to the set of all the flip flops of the circuit involved in a random fault injection approach. After the fault injection, the circuit will be either simulated or emulated in order to monitor the error propagations in the rest of the circuit. In case faults are to be injected in flip flops A,B and C, then this set of flip flops will form the fan-out boundary of the fault injection campaign for this specific attack (as we can see by the dashed line in Fig. 1). Therefore the propagation of such faults to flip flops D, E, and F, through their fan-in cones, will be revealed when simulating or emulating the fault injected circuit. The procedure of flipping the contents of flip flops as well as analyzing the effects of such an event is very efficiently emulated by making use of modern FPGAs and reconfiguration techniques.

III. Partitioning and Intersection Functions

In order to perform the partitioning, a VHDL and Verilog front-end, provided by Verific Design Automation Inc., was used [11]. An abstract VHDL or Verilog design is first analyzed, followed by RTL elaboration. The result of this processing is a netlist database containing basic combinational, sequential, and operator primitives. The C++ API of the front-end is then used to implement the analysis algorithms, and the design is flattened to Verific primitives. Using the clocked signals of the design as an input, all the flip flops are extracted and stored for further processing. In order to extract the logic cones, we implemented a function, which takes as an argument one flip flop of the circuit: its input net is then used as the starting point to perform a depth first search [9] on the directed graph of the netlist, with the aid of the API routines. This function is recursive in nature and it explores each branch, until it encounters a primary input or a flip flop, before backtracking. Visited nodes are memorized in order to traverse each loop of the circuit only once. Explored nets and combinational instances are stored in memory, having as a reference the flip flop of the cone from which the extraction function started. This procedure is then repeated for all the flip flops of the design; at the end, we have at our disposal all the cones, which are equal in number to the flip flops of the design.

After the cone extraction, the sets containing the nets and instances of the cones are processed by means of an intersection function. Each cone is considered as the basic block of the analysis and the output of this function is the intersection of one cone with all the remaining ones. Although so far in our analysis the minimum affected part is one single cone of the design, depending on the spatial characteristics of the attack, we need to consider also the internal structure of each cone as injected with faults, as will be demonstrated in the example that follows.

In order to explain our methodology, first a simple example is given to illustrate the partitioning and optimization of the fault space as well as the ability to keep the localization of the injected fault. Afterwards, an AES sub-circuit is analyzed by the implemented algorithms. Let us consider the example circuit depicted in Fig. 2: by applying the partitioning and intersection algorithms, we obtain the results summarized in Table I. Instance i3 is located in the intersection of cones one and two, as a result of the fact that this instance has a fan-out larger than one. This case shows the advantages and goals of the method, since by performing bit-flipping on flip flops one and two we can simulate or emulate the effect of fault injection on any of the instances which are contained in the first logic cone and thus reduce the complexity of the fault injection campaign. The same reasoning applies also for the remaining cones, two and three. Concerning the fourth one, we notice that it does not intersect with any other; therefore, in order to analyze its behavior, we just need to perform the bit flipping analysis on the flip flop located at its output.

The second circuit chosen to be analyzed by the implemented algorithms is the encryption data path of a parity protected AES implementation [10]. This component is a complex and important part of an AES implementation since it holds the data block of an encryption or decryption operation and performs all the AES transformations, as well as their inverse; on the other hand, the control logic and the key scheduling are not included. The complexity of the data unit and the large number of flip flops it contains, make it suitable to illustrate the partitioning methodology. On the other hand, the results may be biased by the intrinsic parallelism that exists in the AES round computation, where many operations work on very narrow cones specific to each operation and data: key addition, for example, works on single bits; SubBytes operates on all the bytes of the state independently of each other. At first, the 512 flip flops of the design were extracted from the RTL netlist. For each flip flop, the cone was extracted and the intersection function was used to determine the dependency to all the remaining cones.

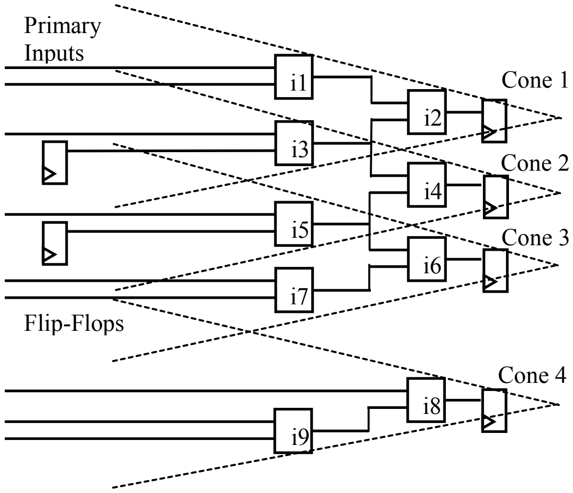

Figure 2. Example of partitioning a simple design

TABLE I. RESULTS OF PARTITIONING AND INTERSECTION ALGORITHMS FOR THE EXAMPLE IN FIG. 2

Logic Cone	Instances	Intersecting Cones
1	i1, i2, i3	2
2	i4, i3, i5	1, 3
3	i6, i5, i7	2
4	i8, i9	-

In Fig. 3, we summarize the results of our analysis. The horizontal axis contains all the flip flops of the design, while the vertical axis contains the percentage of the remaining cones that each cone intersects with. It is noticeable that this percentage is below 12.2% for all the flip flops of the design, while more than half of them are below 5%. These results emphasize that for a moderately complex design such as the considered AES, the cone overlap is not large. Since this overlap will have a large contribution to the determination of the fault multiplicity for a localized fault attack, it forms a measure of the reduction of the fault space in relation to the exhaustive random bit flipping approach. Furthermore it is expected that when only a subset of a cone is affected, the overlap of this portion of the cone with the remaining cones will lead to even smaller multiplicities.

IV. CONCLUSION AND FUTURE WORK

A logic cone partitioning algorithm, based on depth first search on the RTL elaborated graph, has been implemented, tested, and then used to analyze the encryption data path of an AES implementation. An intersection function was used in order to determine which flip flops are influenced in case that one cone is affected by a laser source and impacted by faults. The results show that we have a reduction on the fault space when compared to exhaustive fault injection campaigns, i.e. performing random multi bit flipping of all the flip flops of a given design. This work is a part of the ANR project "LIESSE" where other partners work at lower levels, including the experimental testing and characterization of basic circuit blocks as well as the implementation of TCAD models. Comparison of the results between different levels of abstraction will be used to validate the results obtained with our RTL tools. Next steps of our work will focus on the development of a Laser specific RTL fault model and of an emulation/simulation platform to apply the model to state-of-the-art cryptographic implementations. Afterwards, the implemented platform will be used to strengthen the security of cryptographic circuits by the development of efficient countermeasures.

ACKNOWLEDGMENT

We would like to thank Verific Design Automation Inc. for the offer of their VHDL and Verilog front-end, as these tools gave us the capability to directly and efficiently start the implementation of our methodology.

This work has been supported by the French National Research Agency project "LIESSE" [ANR-2012-INSE-0008-03]. TIMA is Partner of the Labex PERSYVAL Lab (ANR-11-LABX-0025).

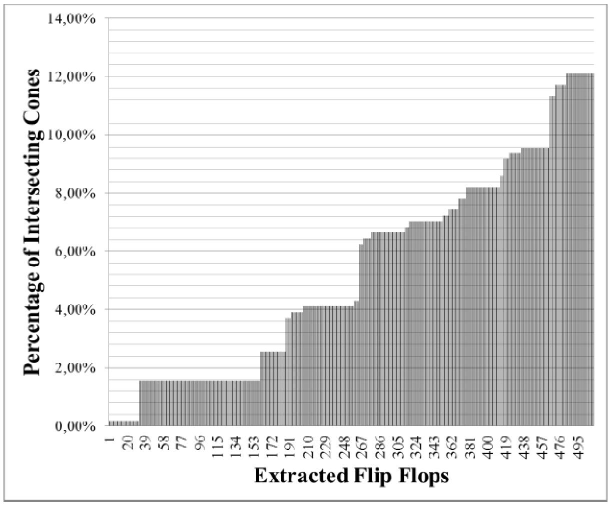

Figure 3. Cone intersection percentages

REFERENCES

[1] D. Boneh, et al, "On the Importance of Eliminating Errors in Cryptographic Computations," Journal of Cryptology, vol. 14, no. 2, pp. 101–119, Jan. 2001.

[2] J. G. van Woudenberg, M. F. Witteman, and F. Menarini, "Practical optical fault injection on secure microcontrollers," in Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on, 2011, pp. 91–99.

[3] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," Proceedings of the IEEE, vol. 94, no. 2, pp. 370–382, 2006.

[4] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," IEEE Transactions on Nuclear Science, vol. 50, no. 3, pp. 583–602, Jun. 2003.

[5] R. Leveugle,"Early Analysis of Fault-based Attack Effects in Secure Circuits," IEEE Transactions on Computers, vol. 56, no. 10, pp. 1431–1434, Oct. 2007.

[6] D. Hély, V. Beroulle, F. Lu, and J. R. O. Garcia, "Towards an unified IP verification and robustness analysis platform," in Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2011 IEEE 14th International Symposium on, 2011, pp. 53–58.

[7] A. Janning, J. Heyszl, F. Stumpf, and G. Sigl, "A Cost-Effective FPGA-based Fault Simulation Environment," Workshop on Fault Diagnosis and Tolerance in Cryptography 2011, pp. 21–31.

[8] Niraj Jha and Sandeep Gupta, "Testing of Digital Systems", Cambridge University Press.

[9] Shimon Even, "Graph Algorithms", 2nd edition

[10] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard," *Computers, IEEE Transactions on*, vol. 52, no. 4, pp. 492–505, 2003.

[11] www.verific.com

[12] S. Venkataraman and S. B. Drummonds, "Poirot: Applications of a logic fault diagnosis tool," *Design & Test of Computers, IEEE*, vol. 18, no. 1, pp. 19–30, 2001.

[13] VLSI Test Principles and Architectures: Design for Testabillity, Morgan Kaufmann - 2006

[14] R. Mariani, G. Boschi, and F. Colucci, "Using an innovative SoC-level FMEA methodology to design in compliance with IEC61508," in *Proceedings of the conference on Design, automation and test in Europe*, 2007, pp. 492–497.

[15] J. Grinschgl, A. Krieg, C. Steger, R. Weiss, H. Bock, and J. Haid, "Automatic saboteur placement for emulation-based multi-bit fault injection," in *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2011 6th International Workshop on*, 2011, pp. 1–8.