Experimental cartography generation methodology for Electromagnetic Fault Injection Attacks

J. C. Rincón-Beneyto[1,3], A. Casado-Galán*[1], F. E. Potestad-Ordóñez[1,3], A. J. Acosta[1,2], E.Tena-Sánchez[1,3]
[1]*Instituto de Microelectrónica de Sevilla (CSIC / University of Seville)*
[2]*Department of Electronics and Electromagnetism, Physics Faculty, University of Seville*
[3]*Department of Electronics Technology, Escuela Politécnica Superior, University of Seville*
Seville, Spain
*casado@imse-cnm.csic.es

***Abstract*—The Electromagnetic Fault Injection (EMFI) is one of the methods to inject faults in the circuits with different purposes, from the security analysis point of view to the study of resilience against environmental conditions of the circuits. Focusing on secure cryptographic applications, in order to study the vulnerability of a circuit and perform successful attacks, it is necessary to induce the electromagnetic (EM) field in a very specific point in the surface of the circuit. This aspect, together with extra inconveniences as for example metal shields of the last metal layers of chips, results in a very poor efficiency in the fault injection through this technique, hindering the possibilities to perform the attacks. This paper presents a experimental cartography generation methodology to carry out automatic EMFI attacks. The presented methodology allows to improve the efficiency in fault injection attacks, showing the areas where the circuit presents greater vulnerabilities against the EM disturbances, allowing to focus the attacks on those points. As demonstrator vehicle, a SRAM is used. Results show that following the steps of the proposed methodology, it is able to detect the point with the maximum fault injection efficiency, along with a great precision of the fault injection, reaching up to been able to inject one bit single fault in the SRAM.**

***Index Terms*—Electromagnetic Fault Injection, cartography, fault injection, hardware security, SRAM.**

I. Introduction

Nowadays, through dedicated devices or Internet of Things (IoT) nodes, the amount of information exchanged daily by users is mostly composed of sensitive data which is susceptible to be intercepted by third parties for malicious purposes. In order to protect this exchange of information, it is necessary to use cryptographic systems that allow the information to be encrypted. These cryptographic systems guarantee the security, integrity and non-repudiation of the information exchanged. While these systems have proven to be mathematically secure due to the length of the keys used to encrypt the information, making brute force attacks useless, attacks on the physical devices that implement these algorithms have proven to be a real problem that must be addressed by designers.

Within this field, recent research work highlights the importance of IoT security analysis [1]. In [1], the authors discuss the risks that exist if attacks on embedded applications used in IoT are not taken into account. Due to this, it is necessary to take on the role of the attacker in order to find vulnerabilities in cryptographic circuits and to minimise the information leakage that may result from them.

Vulnerabilities of implemented cryptographic circuits, come from, in one hand, the passive attacks like Side Channel Analysis (SCAs), those that do not manipulate the circuit and where the properties of the circuit itself are analysed, such as power consumption or electromagnetic emanations [2]. On the other hand, Active Faults Analysis Attacks (FA) are those that manipulate the circuits, introducing certain faults that allow revealing information about the confidential content of the circuit [3].

Considering active attacks, there are many types of techniques, such as manipulation of the clock signals of the circuits, supply voltage variation, temperature variation, pulsed laser or electromagnetic pulse injection [3]. Focusing on electromagnetic fault injections (EMFI), the induction of an electromagnetic field (EM) can cause the malfunction of an encapsulated chip or change the contents of the memory. By inducing Foucault currents on the surface of the chip, it can cause a fault of up to a single bit [4]. In addition to the security aspect, there is also the characterisation and study of the resilience of integrated circuits to fault insertions produced by the environment itself, such as electromagnetic interference or cosmic rays when the circuits are used in space applications [5]. On the other hand, the use of passive protections like passive shield performed by metal layers that cover the sensitive chip parts, makes EM fault injection more difficult due to the fact that a very high level of precision is required [3].

Developing tools and methodologies that allow the study of circuit vulnerabilities with the aim of improving security and to characterise them against fault injection, is a primary objective. Thus, this paper presents an experimental cartography generation methodology to carry out EM fault injection attacks. The obtained cartography of the fault occurrence within the circuit, allows to improve the efficiency in the faults attacks and strength the design of countermeasures. As demonstration vehicle, a static random access memory (SRAM) has been used as Device under Test (DuT). The purpose of selecting a SRAM is twofold. Firstly, the SRAM is a device where the efficiency of the proposed methodology can be easily observed, as the induced failures are highly observable. On the other hand, the procedures and results can be exported to other areas outside cryptography, where fault

insertion is also very important, such as the design of fault-tolerant memories, for instance.

The rest of the paper is organized as follows. Section II introduces the state of the art of the electromagnetic fault injections. Section III describes the proposed methodology. Section IV presents the case of study and results obtained from this work. Finally, in Section V the conclusions of this work are depicted.

II. Electromagnetic fault injection attacks

EMFI is a technique of inserting transient or permanent faults into electronic devices without direct physical contact. Since the work presented by J. J. Quisquater and D. Samyde [4], this technique has been extensively studied and developed to test secure circuits with different purposes. A fault injection method based on a setup using a high voltage pulse generator and a ferrite coil was presented in [7]. This system was developed obtaining an injection system whose resolution, repeatability and control was higher than those presented so far. After this work, similar systems were used to evaluate the effectiveness of EMFIs in attacking different integrated circuits and to analyse the impact of EMFIs [8].

Basically, the EMFI generates a rapid change in a magnetic field directed to the DuT. Changing the magnetic field causes induced currents in the device, which result in a change in the voltage level of the internal signals. To be more precise, it can be said that disturbances mainly alter the behaviour of power and ground networks, in other words, electromagnetic faults occur due to the relationship between the coil of the setup and the Vdd network of the DuT and the other between the coil and the DuT Gnd network [9]. These changes can cause incorrect operations in circuits, affect registers, corrupt memories, among other consequences.

Considering that, this type of technique is very useful when developing cryptanalysis and vulnerability analysis. Depending on the type of cryptanalysis, the use of single-bit or multi-bit fault injections is necessary, as they allow the development of one type of analysis or another. This is why it is necessary to develop systems capable of accurately inject faults such as EMFI. Controlling the number and position of transient faults is of high importance as it allows to break cipher systems efficiently, reducing the attack time and the possibility of causing permanent damage to the systems.

III. Attack methodology

One of the first considerations is the type of injection tip to be used. The size of the injection tip must be carefully selected, regarding the attacked device area and the required fault injection area to cause a exploitable fault. The injection tips used have a ferrite core, through which the necessary voltage discharge is injected to perform the magnetic field that allows to inject the faults. The selected voltage level for the EM pulse is the other critical point of the setup. Note that a low voltage of the EM pulse can not affect the normal operation of the device, thus not causing a faulty keystream. On the other hand, if the EM pulse exceeds the limits of the attacked device, it can be permanently damaged. To this end is important to define the maximum voltage pulse that the DuT can tolerate, and to fix this value in a range that produce the exploitable faults needed for the attacks.

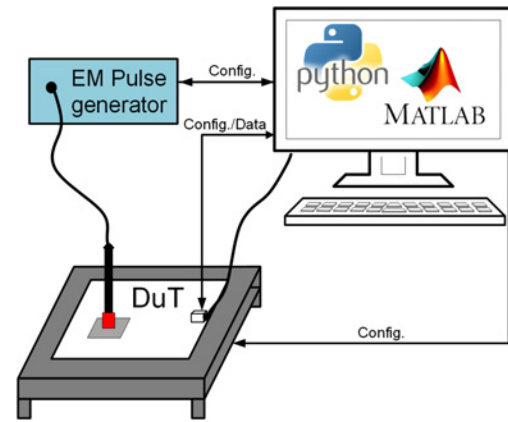

Fig. 1. Generic experimental setup for EMFI cartography generation.

In terms of the different types of faults that can be inserted, four are highlighted:

- Transient faults: modifies the circuit inside it, but only temporarily, after that, the circuit continues working properly.
- Permanent faults: the circuit is permanently modified.
- Deprogramming: caused in certain areas of the DuT which causes the DuT to be deprogrammed.
- Permanent breakage: permanently damage some area.

In order to perform an EMFI, a tool capable of generating a magnetic field in the form of pulses that can cause the DuT to fail must be available. An example of this type of tool is the ChipSHOUTER [6], which has been used in this work. The pulse is generated through an injection tip, which must be connected to the tool that generates the pulse. Then, the injection tip has to be placed on the DuT and once this is done, generate the pulse that causes the faults in the integrated circuit.

Along with the EMP device, an XY-table has been used. The XY-table is a useful equipment for automating the evaluation of vulnerable points in a fault-injection attack. This table allows the attacker to precisely position the injection tip on the DuT and to collect a large number of faulty traces in an efficient way. By automating this process, the risk of human error is reduced and it is ensured that all potential fault locations are thoroughly tested. In the presented methodology the XY-table is used to place the DuT and the probe tip of the EMP device is placed just above it (ideally, the closer the better). In Fig 1, a generic experimental setup for EMFI is shown, where it can be observed the positioning of the tip above the DuT.

In order to extract the information shown in the maps presented in the rest of the paper, the procedure carried out is as follows. First, select the area of the DuT that is desired for evaluation, parametrized by 2 sides of a rectangle. Then divide

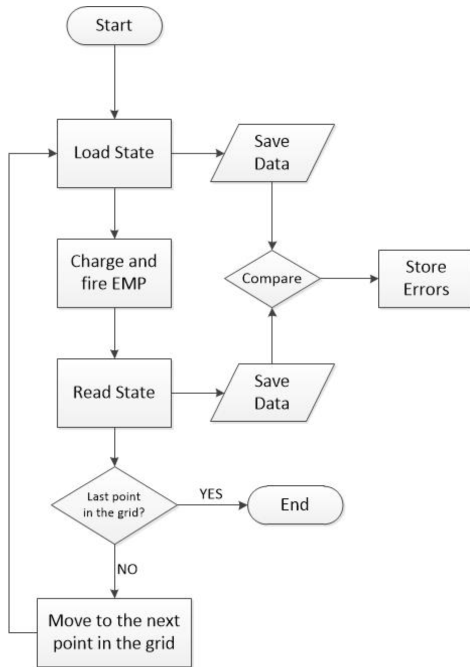

Fig. 2. Cartography Steps.

this area in an evenly spaced grid of points with the desired resolution (the higher the resolution, the more information will be collected, but also the more time and resources will be consumed). Once these parameters have been selected, the cartography extraction can be started. As shown in Fig. 2, firstly the DuT is loaded with the desired specific data, these data are stored for future use. Once the loading is finished, the EMP device is charged and fired at the current position and the new state of the device is read. A comparison is made and the difference between the state after and before the EMP insertion is stored. Lastly, if the current point in the grid is not the last one, the XY-table moves the device to the next point in the grid, if it is the last one, then the operation is finished.

Once all the data are collected, maps can be generated showing the zones with more likelihood of inserting an error in the device. If it is accessible, the information about the specific affected zones inside the device can be represented as well.

IV. Case of study and results

To evaluate the presented methodology, the following experimental setup has been carried out (see Fig. 3). As an EMP generator, the CHIPSHOUTER from NewAE is used [6]. The SRAM included in the CW521 board, better known as Ballistic Gel, has served as DuT [10]. More specifically, the AS6C3216A-55TIN is a low power CMOS SRAM organized as 2,097,152 words by 16 bits or 4,194,304 words by 8 bits; is well designed for low power applications, and particularly well suited for battery back-up nonvolatile memory application. The ZABER ASR100B120B XY-table is used for positioning [11]. Finally, a PC (i5, 64GB RAM, Windows 10) is used to

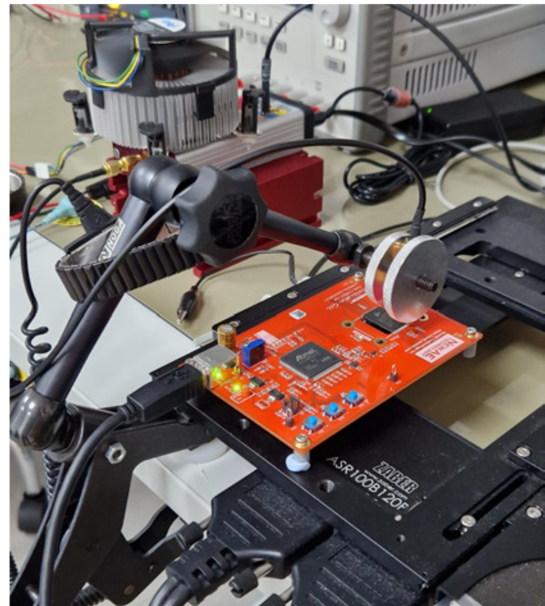

Fig. 3. Articulated arm with the ChipSHOUTER probe tip pointing to the SRAM.

automate/control the whole experiment and instruments with Python, processing the collected data with MATLAB.

According with the operation described in the previous section (see Fig. 2) the whole area of the SRAM module is covered and analysed (18 mm long, 12 mm wide). The probe tip of the ChipSHOUTER is placed at approximately 1 mm above the package of the SRAM. The selected resolution is 0.9 mm/step, generating a cartograph of 20 x 14. Python scripts were used to control and establish the communication with the instruments and the device.

Several experiments were conducted, varying the data loaded into the SRAM and the voltage applied to the EMP.

- All the bits are set to logic 0 - EMP voltage at 450V
- All the bits are set to logic 1 - EMP voltage at 450V
- Random sequence of bits - EMP voltage ranging from 350V to 500V in 50V steps

Depending on the selected grid according to the size of the Dut and the steps selected for the experiment, the time for the data acquisition varies. In the proposed experiment, with a cartograph of 14 x 20 and steps of 0.9mm, the execution time of data acquisition is 30 minutes. A big amount of valuable data are recovered from these attacks, as explained below.

The fault injection zone maps (see Fig. 4 a), c), e), g)) represent the total number of bits changed in each of the points where an EMP was injected in the SRAM. The faulty bit maps (see Fig. 4 b), d), f), h)) show the location of the bits in the SRAM array that were changed, at least once, during all the EMP injection process.

In the cases where the SRAM was programmed with all zeros and with all ones, the fault injection zones maps for both (Fig. 5(a) and Fig. 6(a)) are very similar. Same happens with the faulty bit location maps (Fig. 5(b) and Fig. 6(b)). By zooming in the top left corner of the faulty bit map, where

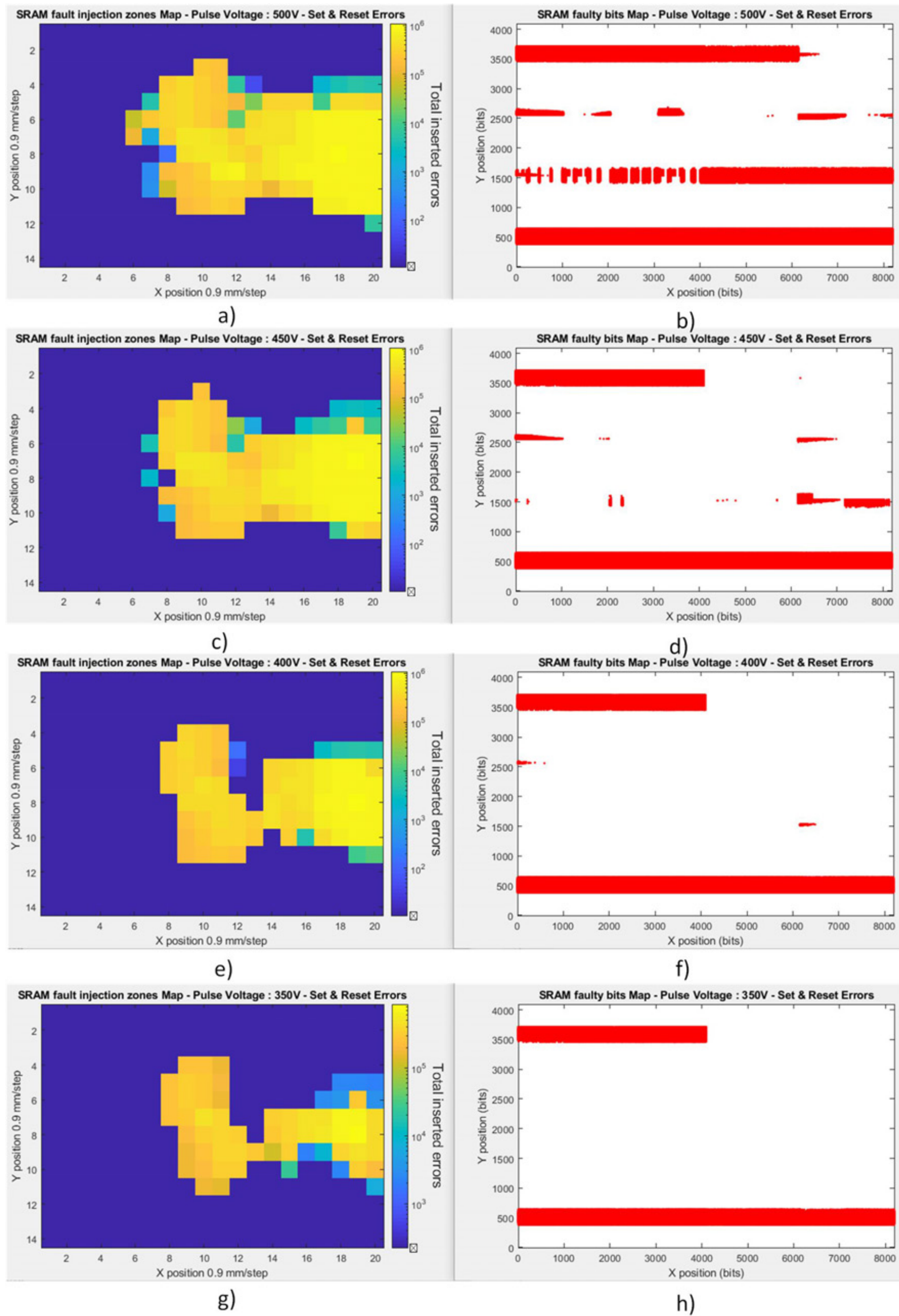

Fig. 4. Fault injection zones map (left) and location of the changed bits (right) - EMP at 500V - SRAM programmed with random sequence of bits.

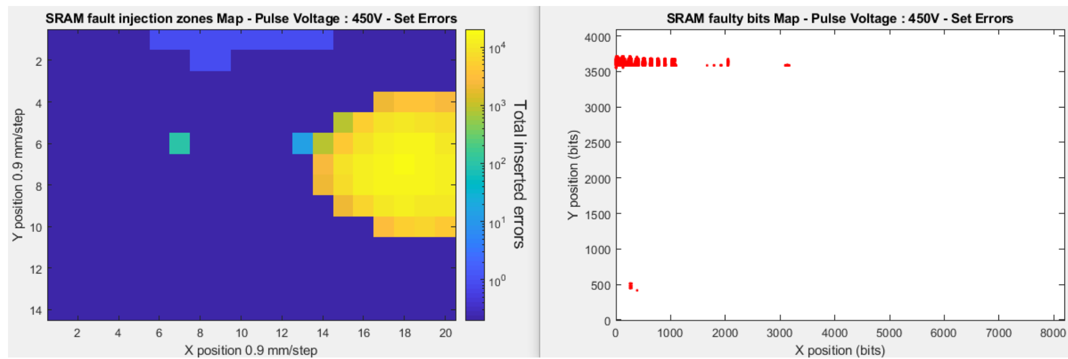

Fig. 5. Fault injection zones map (left) and location of the changed bits (right) - EMP at 450V - SRAM programmed with all zeros.

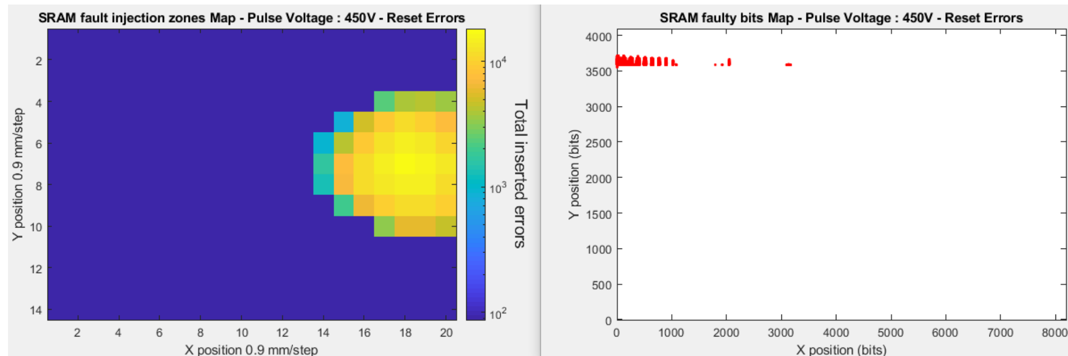

Fig. 6. Fault injection zones map (left) and location of the changed bits (right) - EMP at 450V - SRAM programmed with all ones.

most errors occurred, and representing how many times each bit was changed to an erroneous value (Fig. 7) it can be seen a pattern of columns which seem to be of the same height and evenly spaced in the horizontal axis. Fig. 7 is the map for the all zeros configuration, but virtually the same is observed for the all ones setting.

A similar behaviour could be expected when programed the SRAM with a random sequence of bits, but looking at Fig. 4, many differences can be observed in both maps. The dependence of the number of errors and the size of the fault injection zone on the voltage is revealed as well. It seems that the same fault injection zone in the maps of Fig. 5 and Fig. 6 is also present in the random bits configuration, but a new area appears at the centre of the SRAM too. The higher the voltage, the bigger these zones are and the more errors there will be overall.

In the faulty bit maps, two horizontal stripes seem to be present in all of them and, as the voltage is increased, new zones appear, outlining what appears to be two more horizontal stripes. Similar to the behaviour of the column pattern shown before in the all-zeros and all-ones cases, these horizontal lines appear to be equally spaced as well, this time along the vertical axis. Fig. 8 shows the same faulty bit map as Fig. 9 (500V case), but in this case the error occurrence during all the experiment is taken into account. It can be observed how the bottom stripe has a higher error occurrence than the top one. Also the two intermediate stripes have a much lower error rate (note the logarithmic scale in the colourbar).

By zooming in parts of the top and bottom stripes, some interesting results can be observed. In Fig. 9 the left part of the top stripe is shown. A square/rectangular pattern is seen along with the same column pattern that appeared in Fig. 7 for the all zeros/ones case. Fig. 10 represents a zoom in the right part of the bottom stripe. A pattern of rectangles slightly different from the previous one can be observed, along with a central line and some scattered points up and down.

Taking into account the way a SRAM is implemented typically in an integrated circuit, these kind of results could be expected. Although the layout of the DuT is unknown, it is expected the presence of cell in banks along rows and columns. This would explain the stripe patterns seem (both the horizontal and the vertical), maybe because the EMP is propagating a signal along the SRAM data/address buses. Also the square/rectangle pattern could be due to a ”bank-wise” organization where chunks of cells are arranged in a particular bank. Another interesting result is the non-correlation of the EMP injection zone and the bit pattern observed. This is more noticeable in the all zeros/ones maps (Figures 5 and 6). This could be due to injecting the EMP at the address decoder and forcing it to propagate a signal along the buses of the SRAM (as stated before). Knowing the layout of the die, it could be easier to analyse these results with higher accuracy.

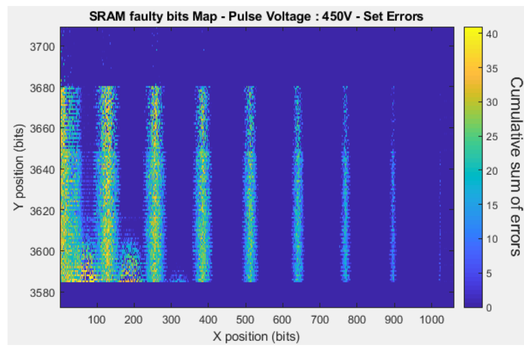

Fig. 7. Cumulative sum of errors per bit, zoom at top left corner - EMP at 450V - SRAM programmed with all zeros.

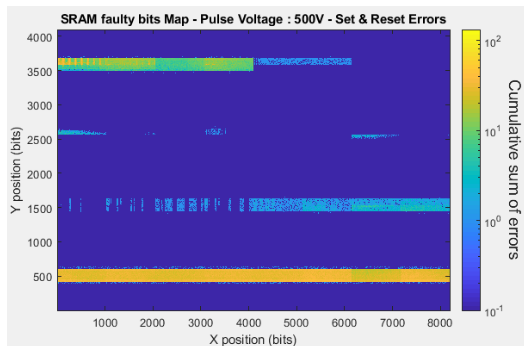

Fig. 8. Cumulative sum of errors per bit, whole map - EMP at 500V - SRAM programmed with random sequence of bits.

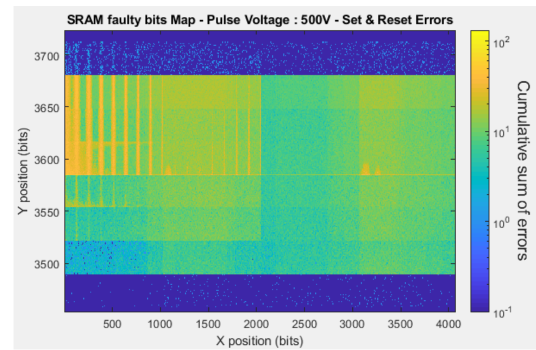

Fig. 9. Cumulative sum of errors per bit, zoom at top left corner - EMP at 500V - SRAM programmed with random sequence of bits.

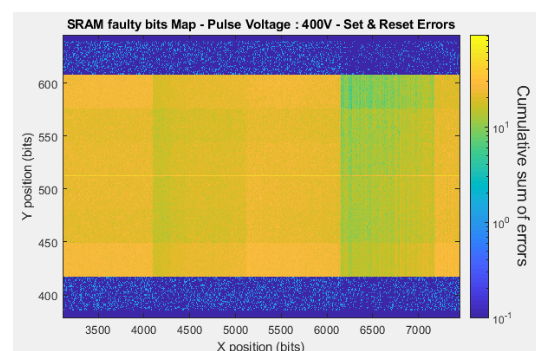

Fig. 10. Cumulative sum of errors per bit, zoom at bottom right corner - EMP at 400V - SRAM programmed with random sequence of bits.

V. Conclusions

This paper presents a experimental cartography generation methodology to carry out automatic EMFI attacks. The presented methodology allows to improve the efficiency in fault injection attacks, showing the areas where the circuit presents greater vulnerabilities against the EM disturbances, allowing to focus the attacks on those points.

The developed setup uses the ChipSHOUTER tools from NewAE for the electromagnetic field generation, an XY-table for the precision positioning of the tip, and a SRAM as a demonstration vehicle. Results show that following the steps of the proposed methodology, this setup allows to determine the areas where the circuit is more vulnerable and allows to know the places where the attacks must be performed in order to obtain successful fault injections, reaching up to been able to inject one bit single fault in the SRAM.

Acknowledgment

This work has been funded by SPIRS (Secure Platform for ICT Systems Rooted at the Silicon Manfacturing Process) Project with Grant Agreement No. 952622 under the European Union's Horizon 2020 research and innovation programme. Authors want to thank Programa Operativo FEDER 2014-2020 and Consejería de Economía, Conocimiento, Empresas y Universidad de la Junta de Andalucía under Project US-1380823, and project Grant PID2020-116664RB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] Z. Kazemi, M. Fazeli, D. Hely and V. Beroulle, "Hardware Security Vulnerability Assessment to Identify the Potential Risks in a Critical Embedded Application," in *IEEE 26th Int. Symp. on On-Line Testing and Robust System Design (IOLTS'20)*, Napoli, Italy, 2020, pp. 1–6.

[2] Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proc. of the International Cryptol. Conference (CRYPTO'99), 1999, pp. 388–397.

[3] Bar-El, H.; Choukri, H.; Naccache, D.; Tunstall, M.; Whelan, C. The Sorcerer's Apprentice Guide to Fault Attacks. *Proc. IEEE*, vol. 94, 2006, 370–382.

[4] J. J. Quisquater and D. Samyde, "Eddy current for magnetic analysis with active sensor," in *Proceedings of eSMART*, 2002, 185–194.

[5] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Reliab.*, vol. 5, num. 3, 2005, 305–316.

[6] NewAE Technology Inc. (2019). CW520-ChipSHOUTER. https://www.newae.com/products/NAE-CW520. Last accessed May 2023.

[7] P. Maurine, "Techniques for em fault injection: Equipments and experimental results," in *FDTC*, 2012, pp. 3–4.

[8] F. Majeric, E. Bourbao, and L. Bossuet, "Electromagnetic security for SoC," in *23rd IEEE International Conference on Electronics Circuits and Systems (ICECS)*, 2016, pp. 265–268.

[9] M. Dumont, M. Lisart and P. Maurine, "Electromagnetic Fault Injection : How Faults Occur," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2019, pp. 9–16.

[10] NewAE Technology Inc. (s.f.). Ballistic gel. ChipSHOUTER User Manual. https://rtfm.newae.com/ChipSHOUTER/Ballistic%20Gel/. Last accessed May 2023.

[11] ZABER Technologies ASR-100B120B-T3A XY Table datasheet. "https://www.zaber.com/api/assets/ASR100B120B-T3A-Datasheet.pdf". Last accessed May 2023.