UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Fault Injection Attacks on Cryptographic Devices

Diploma Thesis

Liappi Sotiria

Supervisor: Stamoulis Georgios

June 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Επιθέσεις μέσω Εισαγωγής Σφάλματος σε Συσκευές Κρυπτογραφίας

Διπλωματική Εργασία

Λιάππη Σωτηρία

Επιβλέπων: Σταμούλης Γεώργιος

Ιούνιος 2022

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Fault Injection Attacks on Cryptographic Devices

Diploma Thesis

Liappi Sotiria

Supervisor: Stamoulis Georgios

June 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων	**Σταμούλης Γεώργιος** Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Ευμορφόπουλος Νέστωρ** Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Καρακωνσταντής Γεώργιος** Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Liappi Sotiria
6/7/2022

Diploma Thesis

Fault Injection Attacks on Cryptographic Devices

Liappi Sotiria

Abstract

Nowadays, hardware security seems to have become both a subject of concern in industry and a triggering event for research in academic field. A Hardware Trojan, the malicious circuit that is inserted in the integrated circuit during manufacturing process, can alter the operation and purpose of the chip. A great percentage of the hardware attacks are related to cryptographic devices that are inextricable part of our current life.

In this thesis, Advanced Encryption Standard (AES), a specific encryption standard is presented in order for the reader to understand the basics of cryptography. Afterwards, fault injection attacks on cryptographic devices and some basic countermeasures for this kind of attacks are discussed.

Keywords:

Hardware Trojan, Hardware Security, Fault Injection Attack, Cryptography, AES

Περίληψη

Στη σημερινή εποχή, η ασφάλεια υλικού μοιάζει να αποτελεί τόσο ένα θέμα ανησυχίας στη βιομηχανία όσο και ένα έναυσμα για έρευνα στον ακαδημαϊκό τομέα. Ένα Hardware Trojan, το κακόβουλο δηλαδή κύκλωμα το οποίο τοποθετείται στο ολοκληρωμένο κύκλωμα κατά την κατασκευαστική διαδικασία, δύναται να μεταβάλει τη λειτουργία όπως και τον σκοπό του chip. Μεγάλο ποσοστό των επιθέσεων σε υλικό σχετίζονται με συσκευές κρυπτογραφίας οι οποίες είναι αναπόσπαστο κομμάτι της σύγχρονης καθημερινότητας.

Στη διπλωματική αυτή, παρουσιάζεται το AES, ένα συγκεκριμένο πρότυπο κρυπτογράφησης με σκοπό τη βασική κατανόηση της κρυπτογράφησης. Έπειτα, αναφέρονται ποικίλες επιθέσεις μέσω εισαγωγής σφάλματος σε συσκευές κρυπτογραφίας καθώς και μερικές μέθοδοι προστασίας από αυτές.

List of Figures

Table of Contents

Abbreviations

AES	Advanced Encryption Standard
COTS	Commercial off-the shelf
CPA	Complete Pipeline Architecture
EM	Electromagnetic
FIA	Fault Injection Attack
HT	Hardware Trojan
IC	Integrated Circuit
OS	Operating System
PPA	Partial Pipeline Architecture
RSA	Rivest–Shamir–Adleman

Chapter 1 Introduction

As the years pass by, 21st-century technology and inventions are of great importance in people's everyday lives. On this account, the menace of cyber-attacks is higher than ever. In the beginning of the 1980s, the first software attacks made their appearance. At that moment, the underlying hardware had been a source of peacefulness. However, during the last decade or two, the complexity of Integrated Circuits (ICs) has increased. It was about 2008 when a crucial malfunction in Syrian radar was observed that might have been deliberately activated using a "back door" placed within a commercial off-the-shelf (COTS) microprocessor [1]. More specifically, the design, fabrication, and distribution of electronics have resulted in a revolution throughout the industry. As the sector transitions to a global business model, new attack vectors emerge and untrusted entities are likely to invade in the IC's life cycle.

But what exactly can someone consider as a hardware security threat? Due to the high cost of ICs development procedure, many semiconductor industries delegate the fabrication to a third-party foundry. In that stage, someone with devious intentions is probable to put on additional hardware in the IC. This additional hardware is used to be called a "Hardware Trojan". A Trojan aims to take advantage of possible vulnerabilities of the electronic material such as smart cards, steal authentication passwords, or exploit data of high importance, for instance, crucial government information.

1.1 Subject of Diploma Thesis

This thesis aims to explicate fault injection attacks (FIAs), one of the main hardware security threats, and especially on cryptographic devices that are of high concern nowadays. More precisely, some of the main types of FIAs are presented as well as the impact they have on chip's function and lifetime. Thenceforth, all these attacks is possible to be detected with many algorithms that have been developed and are sufficient in a very high rate.

1.2 Chapters’ Organization

In Chapter 2, the basics of Hardware Trojans (HTs), Fault Injection Attacks and Cryptography, are introduced, for the reader to have a deeper comprehension of the subject. Chapter 3 provides some examples of FIAs that were applied on AES FPGA implementation and represents the main purpose of this thesis which is to indicate the vulnerability of an IC and how easy it is or not to take advantage of a chip. In Chapter 4 countermeasures against this kind of attacks are presented and finally, in Chapter 5 the conclusions are drawn and further improvements of chips’ security are proposed.

Chapter 2 Preliminaries

2.1 Introduction

As HTs keep emerging, new attacks are developing and end up being a major source of worry for electrical circuit researchers and designers. FIAs especially are a kind of hardware invasion that generally targets straight to the physical structure of the IC. Under those circumstances, methods of this category are very effective and less detectable than others, if are operated delicately. The main targets of FIAs are smart cards and cryptography microprocessors due to the fact that they transfer high-value data that the attacker aims to take advantage of for malicious proceedings. Another thing worth reporting is that encryption and decryption procedures are mainly based on the Advanced Encryption Standard(AES) whose implementation is usually known for everybody who is involved in this field. As a result, when someone knows the structure of the circuit, is way easier to invent new HTs to place on it.

2.2 Hardware Trojans Classification

As mentioned in [3], Wang, Tehranipoor, and Plusquellic were the first that established a comprehensive categorization of HTs. Since there are numerous configurations about chips' functions, Trojans' classification is distinguished into three major types based on their physical, activation, and action features. This taxonomy, which is described extensively below, covers the fundamental properties of HTs and is beneficial for understanding and assessing the basics of various detection techniques and systems.
In the category of physical features, several hardware implementations of HTs are described. Beginning by the type subcategory, Trojans are divided into functional and parametric ones. More specifically, the parametric class comprises HTs that are physically realized by modifying existing wires or/and logic. On the contrary, the functional class includes Trojans that are implemented by adding or removing transistors or gates. Furthermore, the size category takes into consideration the number of components that have been added, destroyed, or altered on the chip. Proceeding on the distribution category, worthy of saying is that outlines the actual position of the HT on the chip. Last but not least, the structure category describes the situation in which an attacker is obliged to reconstruct the layout in order to implant a Trojan. In this case, modifications are possible to result in a modified positioning of the components and even more of the complete design.

The factors that induce a Trojan to be triggered and perform its harmful purpose are referred to as activation characteristics. In more detail, these features, as Fig. 1 shows, are classified as externally activated, for instance by an antenna or a sensor that might interface with the outside environment, and internally activated which are differentiated between always-on and condition-based. “Always-on” indicates that the HT is always active and has the ability to interrupt any operation of the chip whenever it is feasible. It is a common practice for this category, the Trojan to be inserted in sparsely used nodes or paths to be less detectable. On the other side, Trojans of the condition-based subcategory are idle and are only triggered when a specific condition is satisfied. Temperature, voltage, humidity, or electromagnetic (EM) interference are some parameters that the output of a sensor can detect and might be used to activate the device. An internal logic state, a certain input pattern or the number of inputs, and generally whatever an adversary has come up with, are some instances of additional conditions.

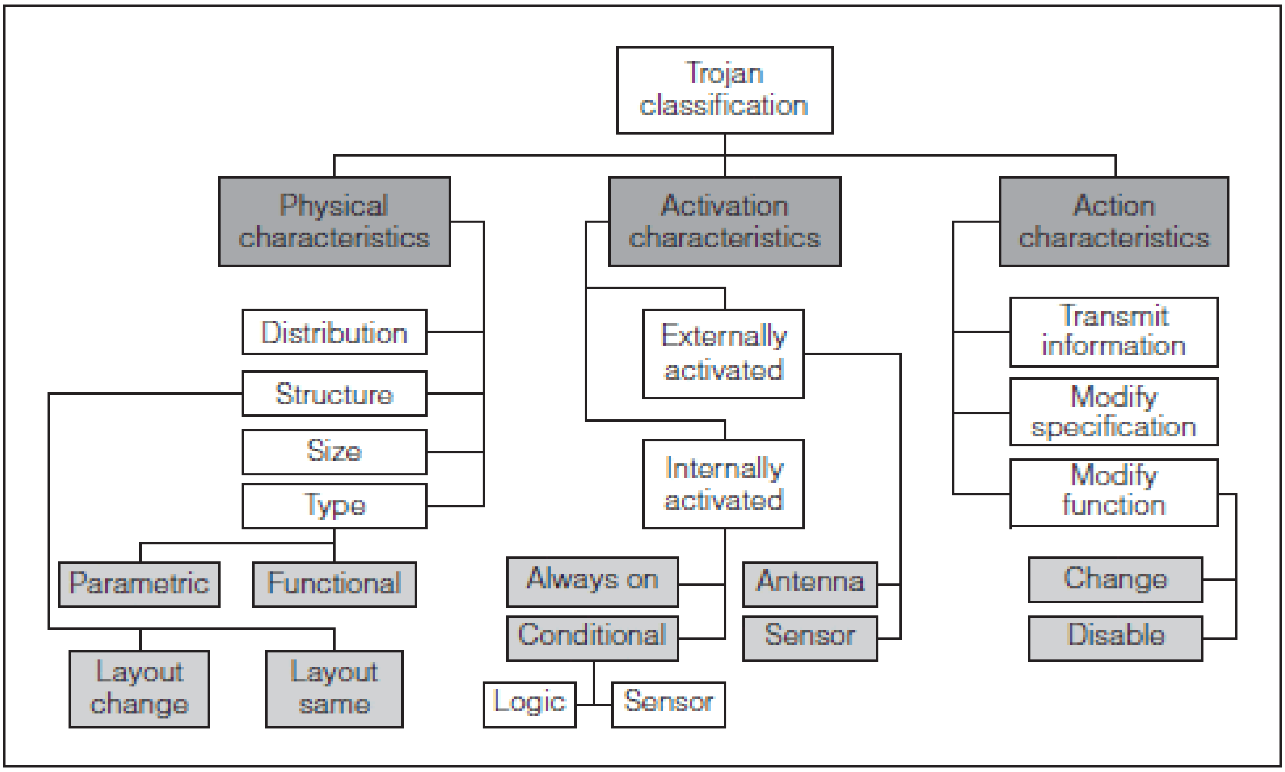


Fig. 1: Trojan Classification [13]

The sorts of disruptive activity introduced by the HT are distinguished by its action characteristics. Modify function, modify specification and transmit information are the three categories that a Trojan’s behavior is divided into. Modify function refers to adding, deleting, or bypassing the logic of the circuit in order to alter the chip’s function. The modify specification class discusses HTs that aim their assaults on moderating crucial parameters such as power consumption, throughput, or latency. At last, in the transmit information are included the Trojans that convey critical information to an adversary, which is and the main concern of this thesis.

2.3 Cryptography

2.3.1 What is Cryptography?

"Cryptography" as a term, has its origin in Greece and precisely translates into "secret writing". It was before the development of digital communications that cryptography was mostly utilized by the military for espionage purposes as [4] refers. With the advancements in technology succeeding one another, the Internet is one of the main media for achieving people's communication and information transfer. Although it is very easy in use and the abilities that it provides are countless, many dangers are lurking, with data interception being the main among others. This is where cryptography is needed the most. Using cryptographic algorithms in software and hardware systems can assist in rendering messages incomprehensible to everybody except the intended receiver. Cryptography is distinguished into two parts: encryption and decryption. The real, authentic data (plaintext) need to be transformed into a text that is not recognizable by anybody. This transformation is called encryption and the product of this procedure ciphertext. When ciphertext arrives at its receiver, decryption helps to convert it into the original data, so as to achieve the successful communication between sender and receiver.

2.3.2 Types of Cryptography

Cryptography is classified into two main types: secret-key cryptography and public-key cryptography.

In secret key cryptographic algorithms, encryption and decryption procedures employ the same key for communication. On this account, secret cryptography is also called "symmetric". In order to communicate efficiently, it is necessary for both the sender and the receiver to have at their disposal the same key. Worthy of saying is that Data Encryption Standard (DES) is one of the most famous secret key cryptographic algorithms that AES succeeded after years.

In public-key cryptography, on the other hand, named "asymmetric" otherwise, two keys are utilized. One for encryption and another one for decryption. Moreover, two keys are available for each user. There is one public key that all users have access to it and a private one that is kept secret. These two are connected to each other via mathematical relations. It is customary, the public key to operate encryption and the private one the decryption process. An instance of an algorithm of this category is Rivest–Shamir–Adleman (RSA) which is taught in many universities worldwide.

When someone researches cryptography, they can discover that there is a connection between the two types of cryptography discussed above. Symmetric encryption uses some session keys that somehow must be encrypted. This key distribution is successfully achieved by public-key encryption.

Considering the advantages of each type compared to the other, as they are mentioned in [3], secret key cryptography is faster and its key is considered more strong. Moreover, the conversion between the plaintext and the ciphertext is contemplated to be more straightforward with not much information in between. However, in public-key cryptography, there is much less key handling in session, there are multiple levels of safety because encryption and decryption are operated with different keys and furthermore, each user has two keys available.

An example diagram for better comprehension of cryptography is shown in Fig. 2, depicting a typical secure file transfer. In the beginning, the client receives from the server the public key during public key cryptography. The client, then, encrypts the session key using public key and when the message is sent to the server, the session key is decrypted using its private key. Finally, the client encrypts the file to be sent using the session key and the session key is used from the server to decrypt the message of the file.

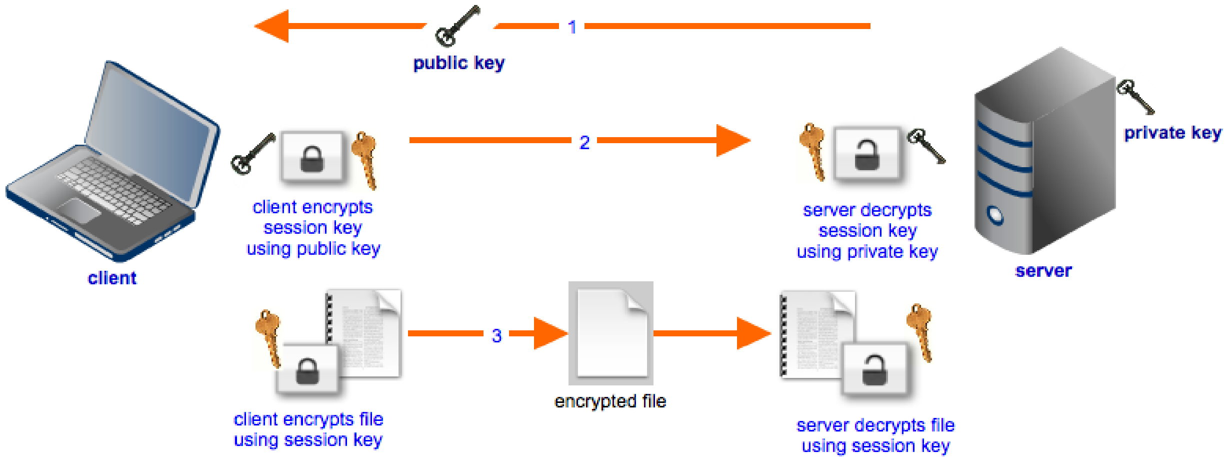

1
public key
client
2
client encrypts session key using public key
server decrypts session key using private key
private key
server
3
client encrypts file using session key
encrypted file
server decrypts file using session key

Fig. 2: Secure File Transfer Protocol [2]

Now that the basics of cryptography have been introduced, it is time to move forward to delve into the most commonly used algorithm for encryption and decryption, the AES.

2.3.3 Advanced Encryption Standard

AES is divided into three versions: AES-128, AES-192, and AES-256. Each of these numbers corresponds to the length of the key, in bits, that is used for encryption and decryption. One of the main differences between these editions is the number of rounds that need to be made for the algorithm to be completed. In particular, as this number goes hand in hand with the size of the key, when the length of the key is getting bigger so does the number of rounds. Especially, for 128, 192, 256-bit length keys need to be implemented 10, 12, and 14 rounds respectively. In this thesis, the AES-128 algorithm steps will be introduced. The procedure for the other two is similar with some adjustments.
The main flow of the AES algorithm is shown in Fig. 3 below.

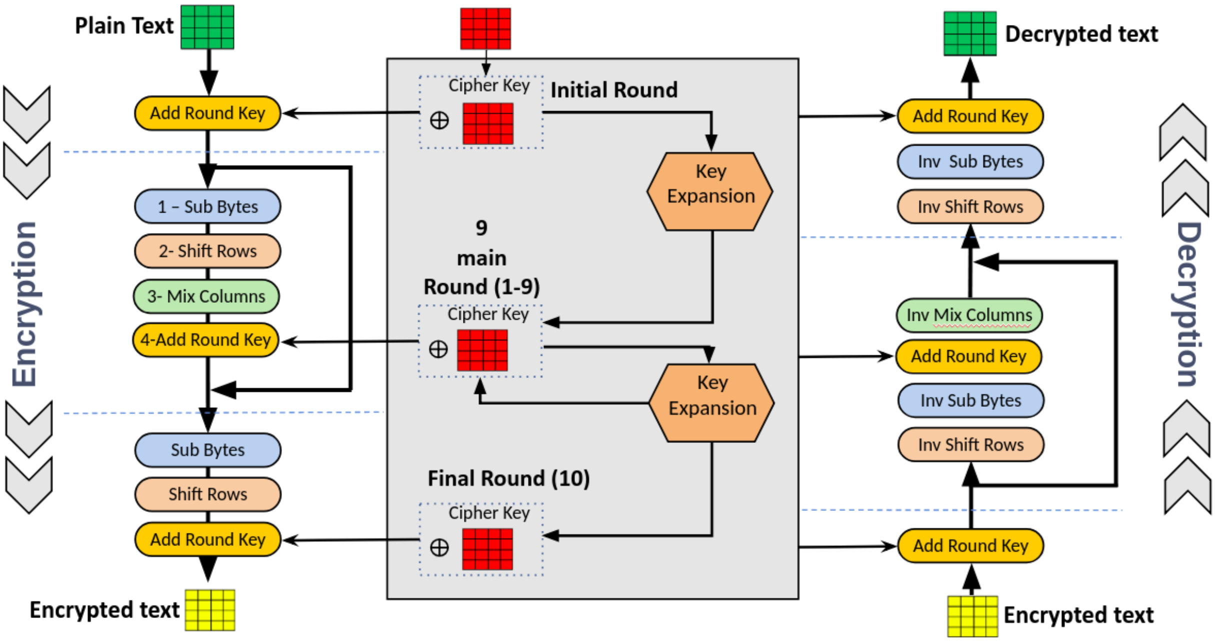


Fig. 3: AES algorithm [2]

First of all, the plaintext is transformed into a two-dimensional array with 1 byte in each cell. The way that this array is filled is vertically, Each column is defined as a word.
It is easy to observe that before the first round, there is a pre-round transformation in which Add Round Key method is applied. After that, the first round begins and the Sub Bytes, Shift Rows, Mix Columns, and Add Round Key procedures are repeated for ten rounds, except for the final round where Mix Columns is dismissed.

Add Round Key

In this transformation, a round key, there is further information for it below, with length 128 bits are XORed with the 128 bits of the data bit by bit (Fig. 4). In the initial step, the round key is considered as the cipher key. If Add Round Key takes place in the tenth round, the output data is the ciphertext.

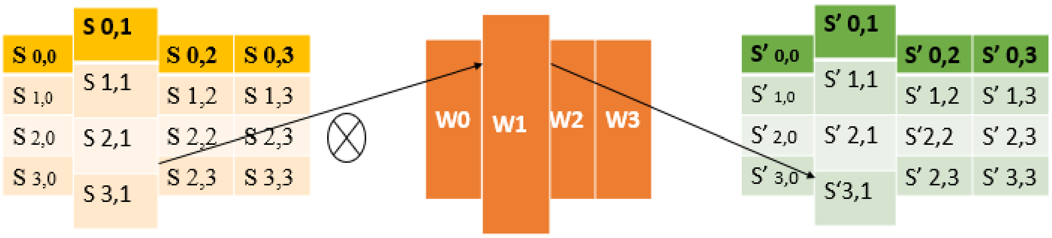
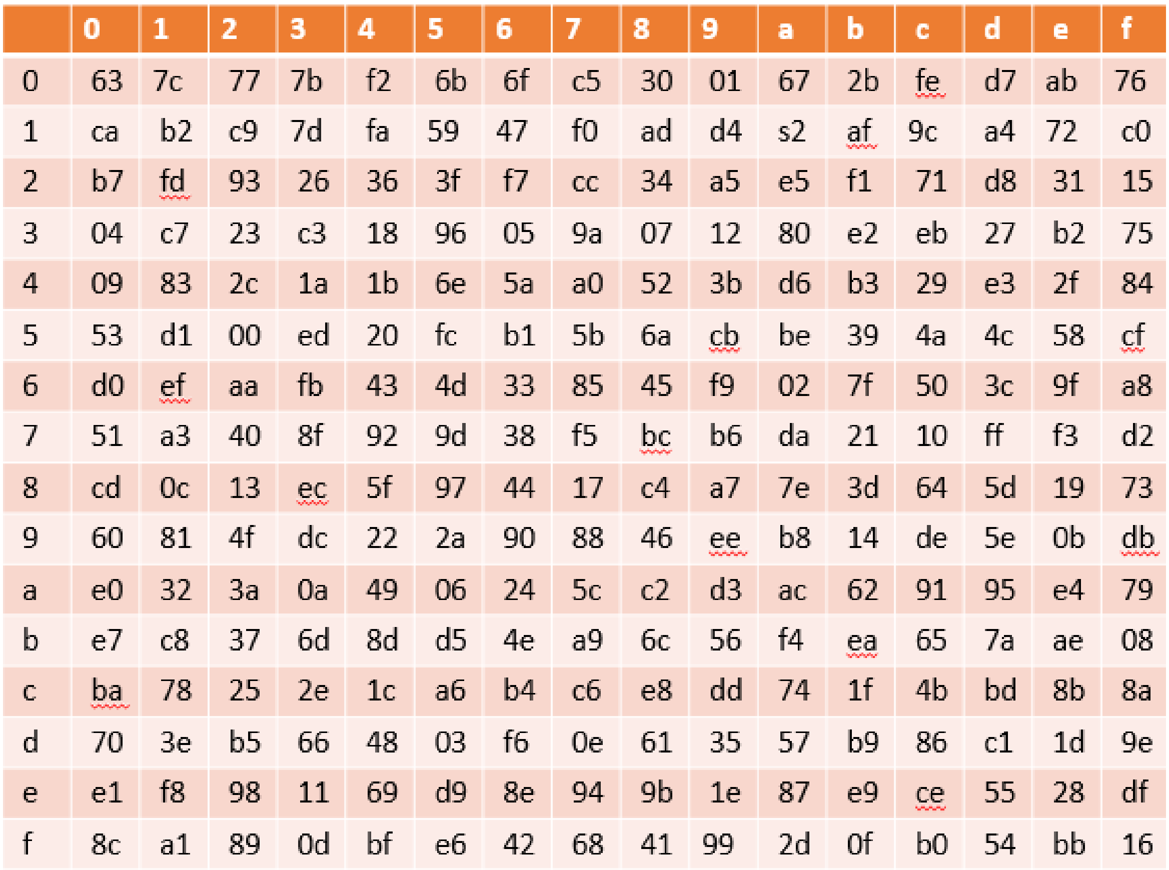


Fig. 4: AddRoundKey [2]

Byte Substitution (SubBytes)

A standard lookup table, called Rijndael S-box (Fig. 5) is used for bytes' substitution. Practically each byte of the text is substituted by the respective byte of the S-box. The initial 4 bits of the byte can be considered as row index while the last 4 as the column index. In this way, we can extract the value from S-box (Fig. 6). For instance, if the byte for substitution is h'a8, h'c2 will take its place.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	b2	c9	7d	fa	59	47	f0	ad	d4	s2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig. 5: Rijndael S-box [2]

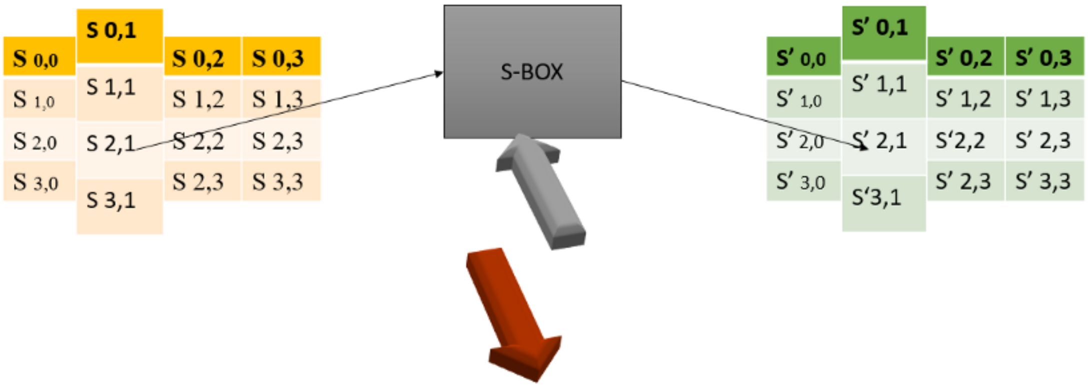


Fig. 6: SubBytes [2]

Shift Rows

In this step, rotations to the left are operated, depending on the row index. The first row remains at its position. The second one is rotated one position, the third row two positions, and the fourth three positions to the left (Fig. 7).

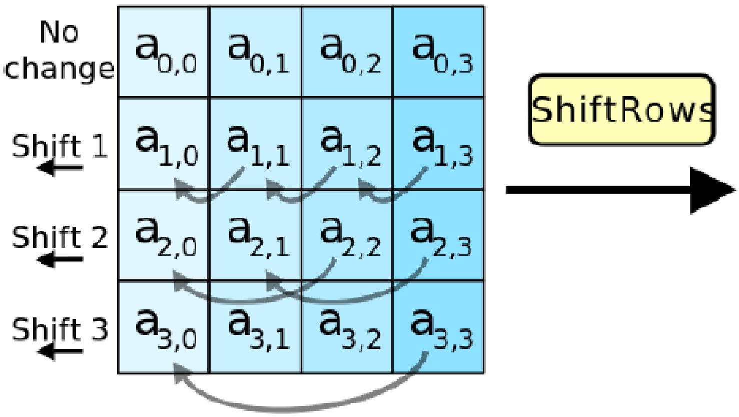


Fig. 7: ShiftRows [2]

Mix Columns

Each word is transformed by being XORed with the standard state matrix as seen in Fig. 8. It is important to be repeated that this step is not applied in the last round of the algorithm.

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Fig. 8: MixColumns [2]

Key Schedule

The key schedule is the procedure for the generation of the round keys. Specifically, the initial cipher key in each round of encryption is modified to a round key. Considering that each key consists of four words, the final array of the keys will consist of 44 words, 4 for the cipher key and 40 divided equally for the next rounds. Each word of the array is linked with its previous one and the one four positions back. For all the words whose index in the array is not a multiple of 4, only an XOR operation between the previous word and the word 4 positions earlier. Words in positions that are a multiple of 4 are following some more complex steps. In the beginning, the previous word is rotated by one byte to the left. Then, each byte is substituted using the S-box table and the result of it is XORed with the word 4 positions earlier plus a round constant (Fig. 9).

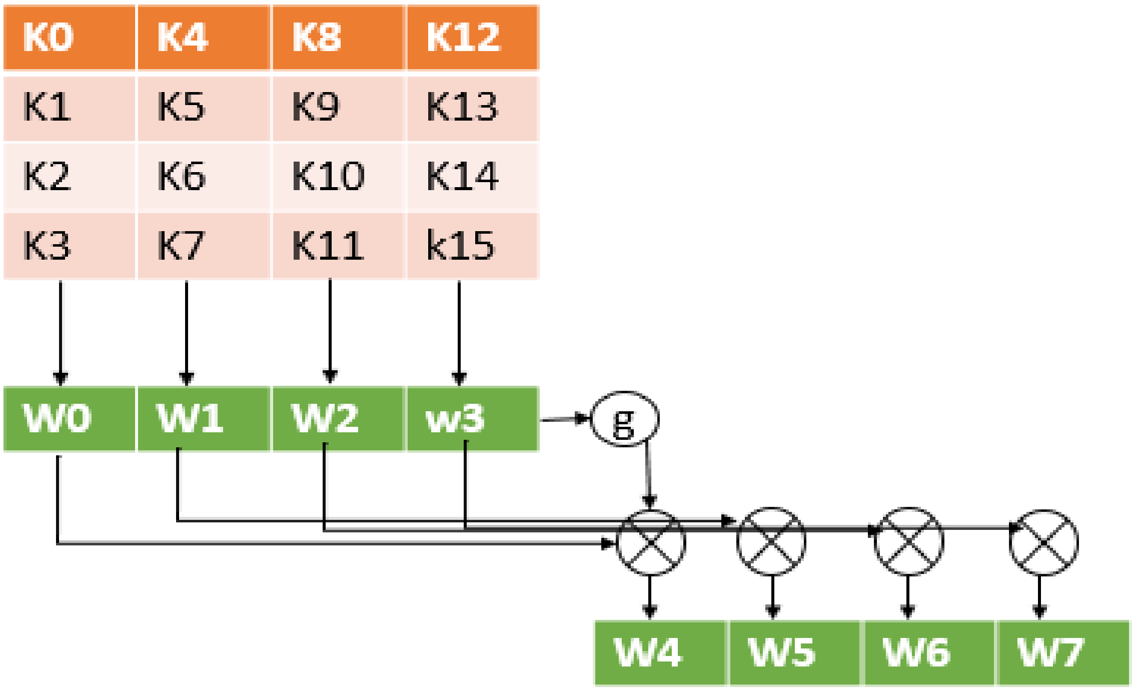


Fig. 9: Key Schedule [2]

These were all the steps for the encryption process. Performing the steps in reverse order, namely, first Add Round Key, Mix Columns, and next Shift Rows and Byte Substitution is the whole decryption. The only thing that demands attention is that although being extremely closely linked, encryption and decryption algorithms must be performed independently because of the reverse order of the steps that might lead to conflicts.

2.4 Fault Injection Attacks

After the basics of cryptography have been described, the time has come to display some of the main kinds of FIAs on cryptographic devices. The main purpose of these attacks is either to exploit the secret key to have access to the communication messages that are exchanged, or to reduce the life expectancy of the chip. But let's discuss a classification of FIAs before proceeding with further analysis.

2.4.1 Fault Injection Methods Classification

The first distinction of FIAs is in active and passive attacks. An adversary performs an active attack when the chip is operated deliberately outside of its regular operating conditions. When the chip operates within normal specification but the intentions are to find leakage of secret information, the attack is called passive.

Another categorization of fault injections is according to the percentage of physical invasion on the chip. Explicitly, non-invasive attacks do not alter the physical layout of the chip, semi-invasive ones involve decapsulation, i.e removing the outer layer of the device, and last, fully-invasive attacks entail delayering and sometimes inquiring the surface of the chip [14].

2.4.2 Fault Injection Techniques

As [5] refers, some of the most important fault injection techniques are clock and voltage glitching, electromagnetic interference, light and laser usage, and Focused Ion Beams.

Clock glitching, which is a non-invasive method to inject faults, is related to handling the clock signal in a way that setup or/and hold times are controverted.

Voltage glitching, respectively, involves tampering with the power supply of the chip. One example of this method is operating the chip with a used-up power supply. In this case, transistors with a voltage threshold higher than the voltage supply, are never get activated resulting in occasional failures in the device, as some parts of the chip are never triggered [5].

In respect to electromagnetic fields and laser interference, they cause transient faults due to voltage or current pulses that lead to the malfunction of the chip.

The method that could be considered as the most accurate FIA is the Focused Ion Beams. Ions at low and high beam currents are used in order to represent the physical image of the chip. In this way, modification of trace paths by adding or removing circuits and more often simple wires, and redirection of signals are the main purposes of FIB [5].

Presentation and analysis of some of the above methods take place in the following chapter.

Chapter 3 Fault Injection Techniques for Cryptographic Devices Attacks

3.1 Clock and Voltage Glitch Attacks

For the purpose of understanding power/voltage and timing attacks, it would be better if a small introduction to timing constraints came about. First of all, in every IC there is a clock signal that is employed for the internal processes to be synchronized. The right data can be passed through the registers if and only if there is no timing violation. When talking about timing violations, there are some restrictions for the successful data and the clock signals they have to be provided at. To be more specific, every signal has its setup and hold time. The setup time is defined as the time period for which the signal must be stable before a change in clock's edge to guarantee the robustness of the operation. Respectively, the hold time is described the same as setup time but with only difference that the time constraint is after the clock's edge. Fig. 10 depictures the violations(b,c) taking place in a D flip-flop and a normal signal operation(a).

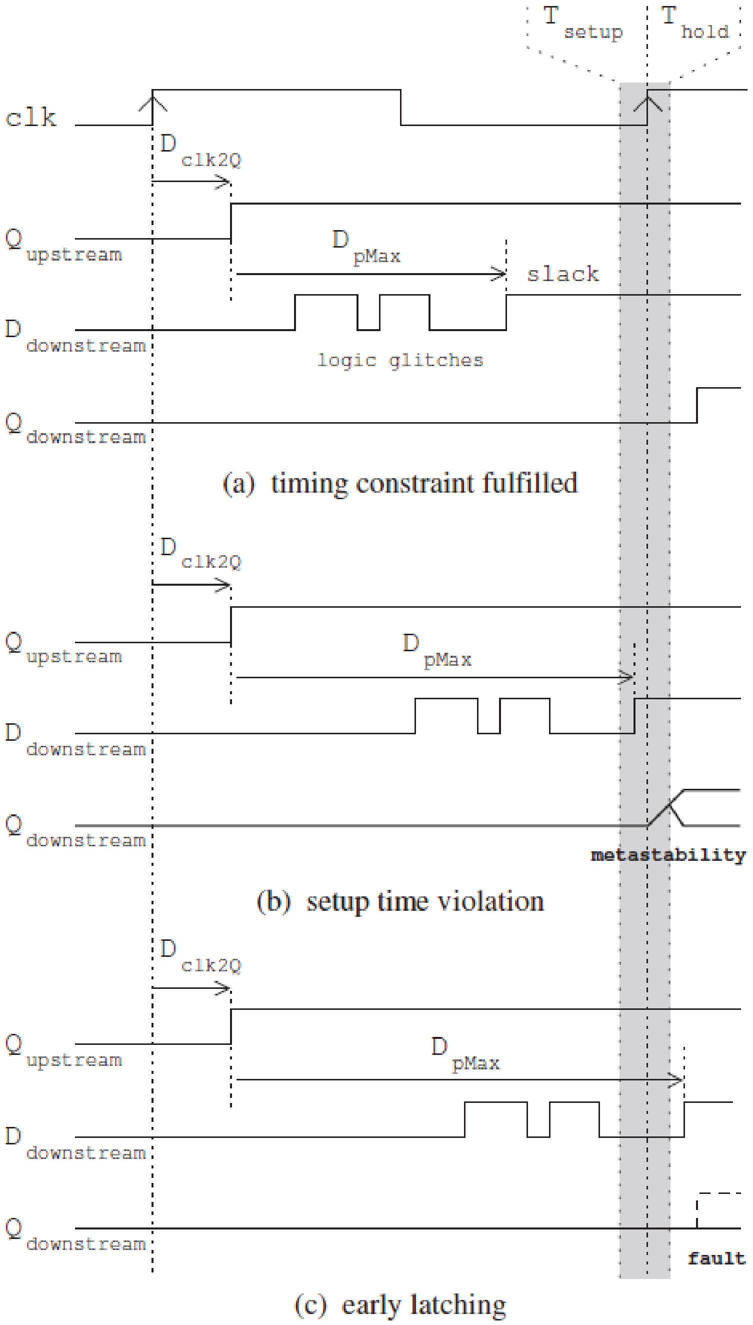


Fig. 10: Timing constraint fulfilled or violated [10]

While studying power and clock glitch attacks, authors in [10] came up with the idea of comparing these two attacks since both of them are taking advantage of timing violations. To the extent of the experiment's completion, the appropriate set up was used. That is an AES test chip that implements the known algorithm, a board to be based on and two voltage pulse regulators(Agilent 814A and Picosecond 10,300B). The test procedure carried out as follows. For known plaintext and key, a clock glitch pulse generator tried to attack at every AES round, except the first due to its small data propagation time. For a standard voltage(1.2 V), the critical times namely the time for the attack to steal the data in each round are reported in Fig. 11. It is easily observed that all of these times are lower than the clock period(10ns).

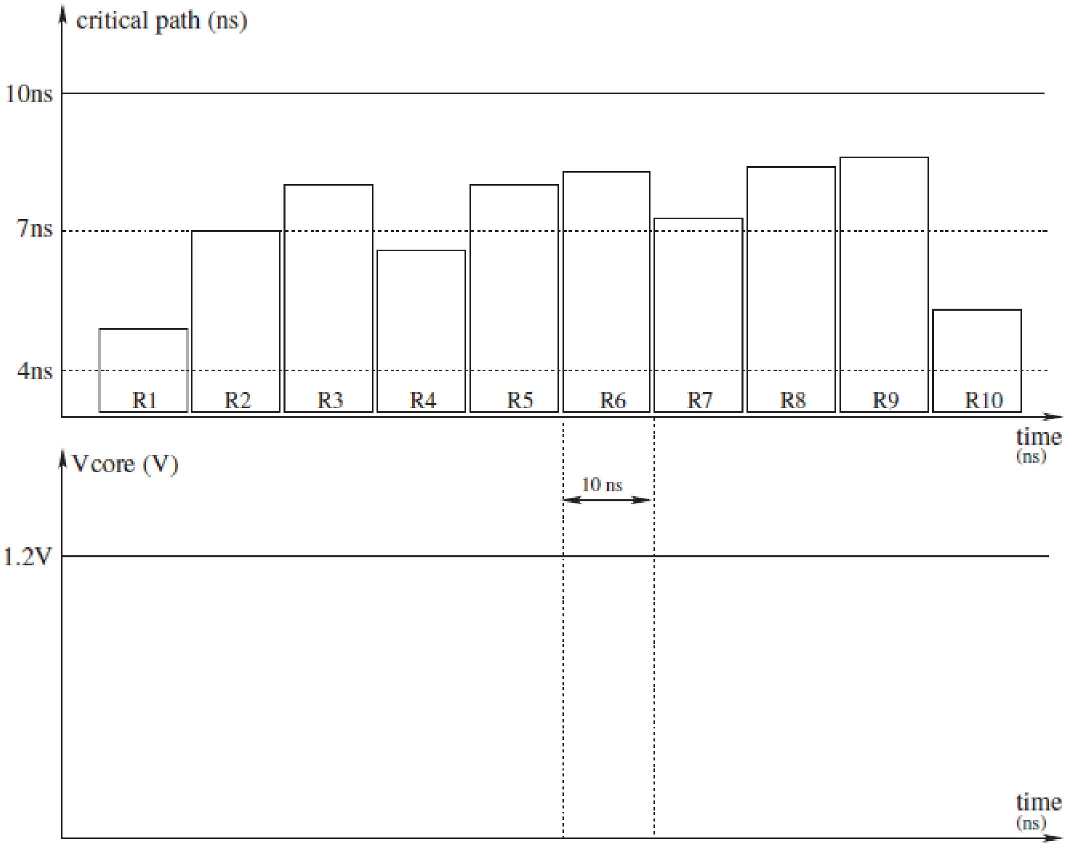


Fig. 11: Critical paths in every AES round [10]

In addition, when the generators utilized to generate and a DC pulse, it underpowers the circuit. As a result, the critical times are increased(Fig. 12).

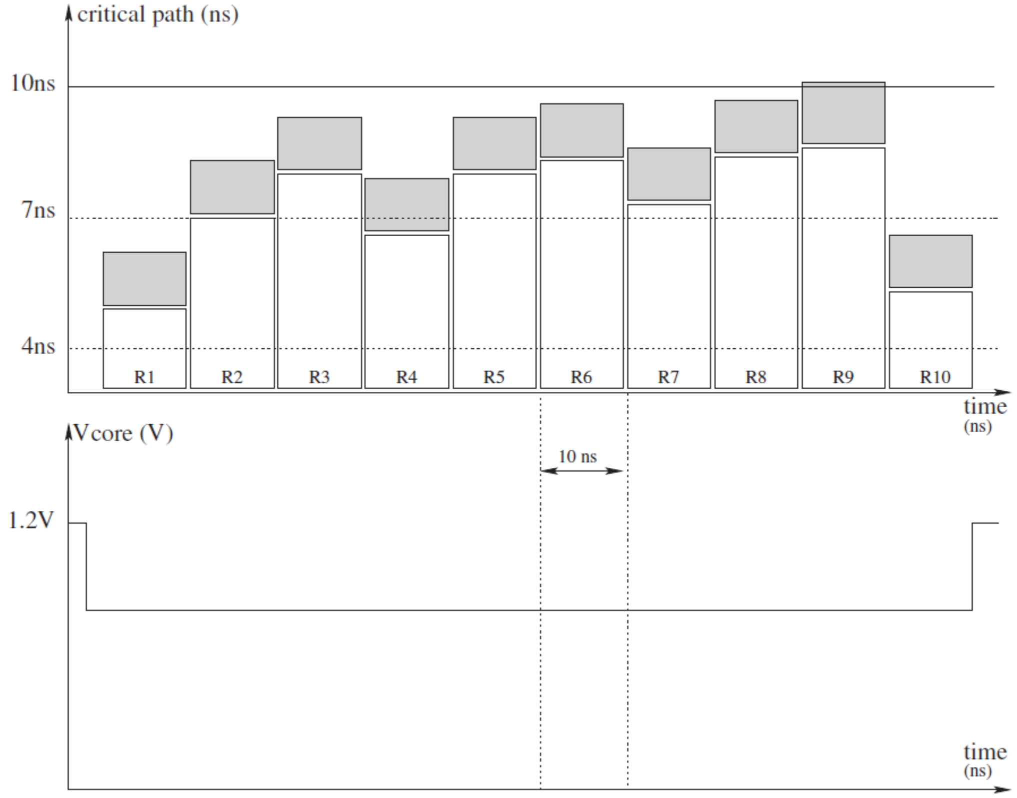


Fig. 12: Critical paths in every AES round with underpowering [10]

As conducted by the authors, for an actual comparison of voltage and clock glitches attacks, testing became under negative power supply, namely power supply that is more negative in polarity than the ground of the circuit. The reader easily observes that the power supply that induced in the 3rd and 6th AES round(Fig. 13 and 14 respectively) lead to increased critical times that exceed the clock period.

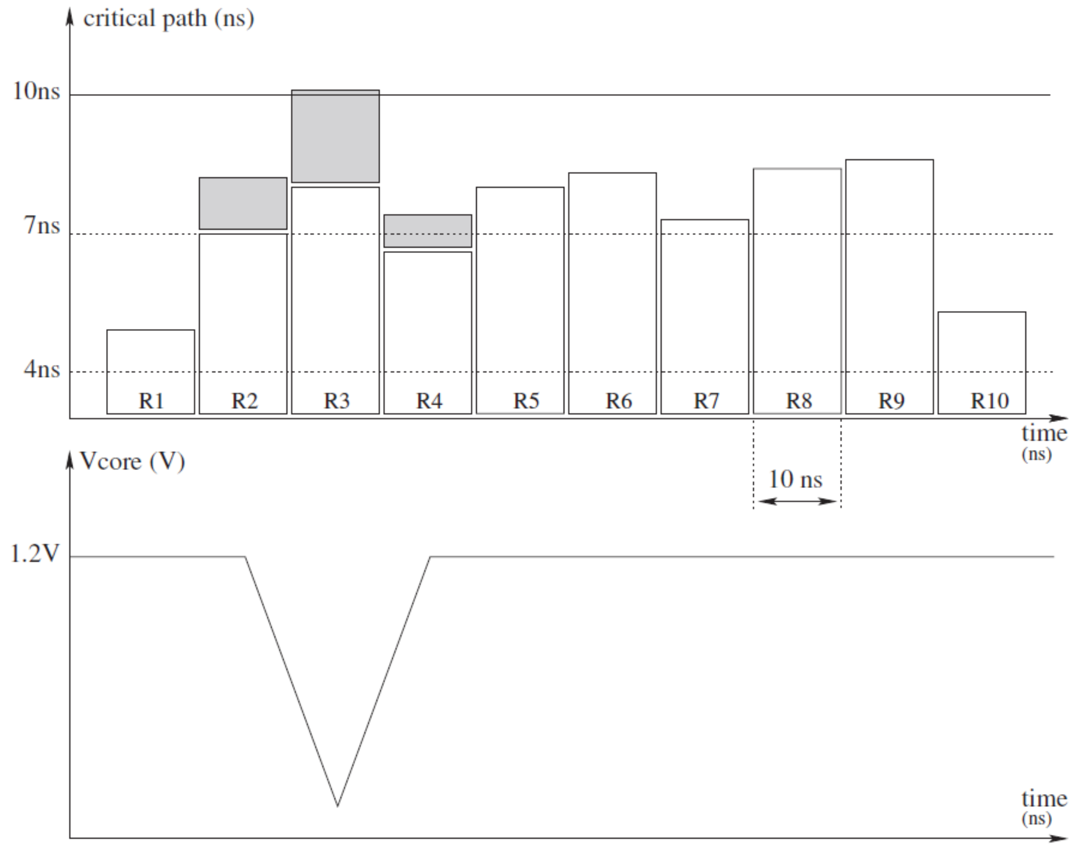


Fig. 13: Results of power glitch on 3rd round [10]

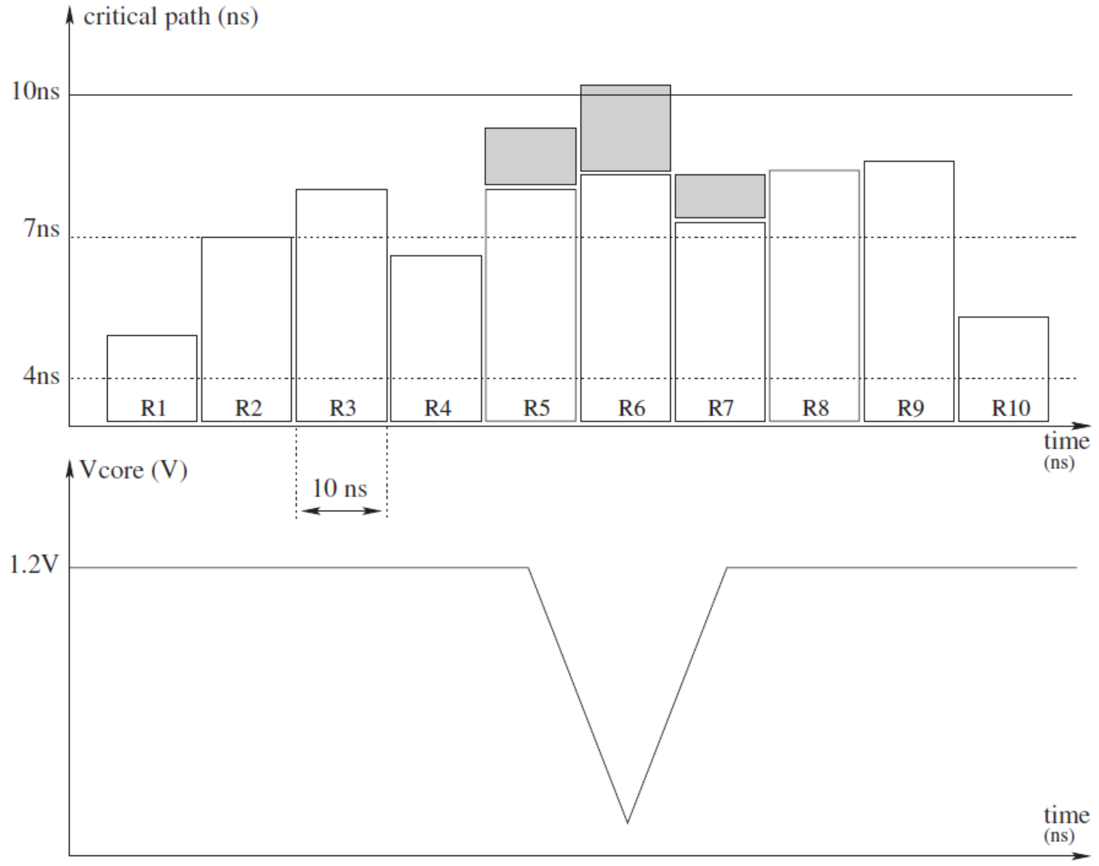


Fig. 14: Results of power glitch on 6th round [10]

For the accomplishment of their research, authors used a library with 1000 plaintext-key sets. For the first pair for each data, the first round of the encryption was free of faults so as to reveal the ciphertext that was used for comparison in the next rounds. The AES algorithm was operated for all the sets for both clock and power glitches doing repeated comparisons between the actual ciphertext and the one that was expected to be taken. In conclusion, the results showed that clock and voltage glitches have similar to identical effect on the operation of the chip. Specifically, a 70% percentage indicated that faults were identical in both cases and in the other 30% there were some timing violations of the nearest critical path and some injections took place in rounds that were next to the target one.

3.2 Optical fault injection attack using laser beams

As optical FIAs are categorized in semi-invasive attacks, the chip preparation that is necessary in order the attack to be successful is the minimum possible. The access is provided either from the front or from the back side of the chip. It is worth noting that invasions in front side are more difficult because transistors are placed at this side.

As the Fig. 15 shows, the light wavelength differs depending of the target side. Silicon in back side must be semi-transparent so the preferable wavelength is necessary to be above 1000nm on average. On the other side, front side operations could be executed with a variety of wavelengths with shorter wavelengths to be prefered. This is due to the fact that the increased energy levels allow transistors to switch more easily. Authors in [6] selected the values for front and back side laser wavelengths to be 808nm and 1064nm respectively.

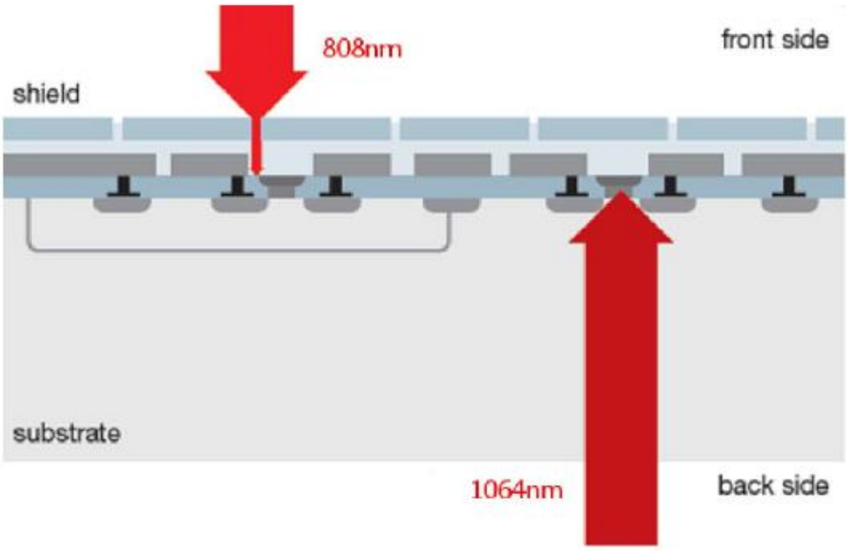


Fig. 15: Front and back side attacks [6]

In the same paper [6], two experiments are presented. They operate optical FIAs in two different secure cards and the results are noticeable.

The first card is programmable and isn't equipped with an Operating System(OS). The program that is performed in the card, requires a two-phase PIN verification. The user is able to gain access only if both verifications are succeeded. Regarding the hardware, a decapsulation of the back of the chip is needed to gain optical access at its substrate.

Then, by selecting the 1064nm laser that introduced above, fault injections are performed in little locations of the chip. When there is a negative response on the output, the candidate vulnerable regions are decreased. Now it's time to find the timings at which the laser beams should be inserted. For doing so, authors considered 100 clock cycles, with each cycle to be 1μs, and for every time period a laser beam was operated. On the 72th μs, the first successful PIN verification happened. For the second verification, the same process is followed and the second required time is 33 μs. Remarkable is the fact that changing time by 1μs, the FIA fails.

In the second case, the card that is used is programmable, has OS and some extra hardware for optical faults detection. The access is quite easy as the front side is transparent and there are no shields protecting it. In this card there is an accelerator for a specific encryption standard which is the main target of the attack. The setup depicted in Fig. 16 is utilized aiming in power signals measurement when an encryption process is in action.

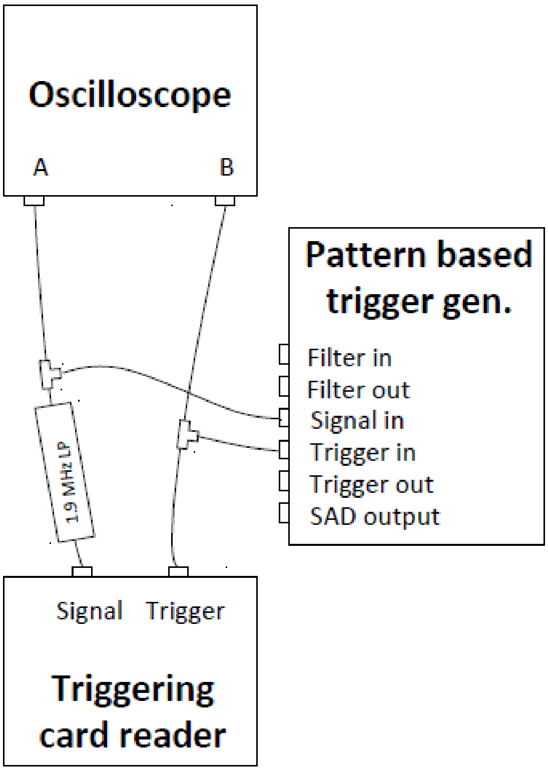


Fig. 16: Equipment [6]

The results are presented in next Fig.

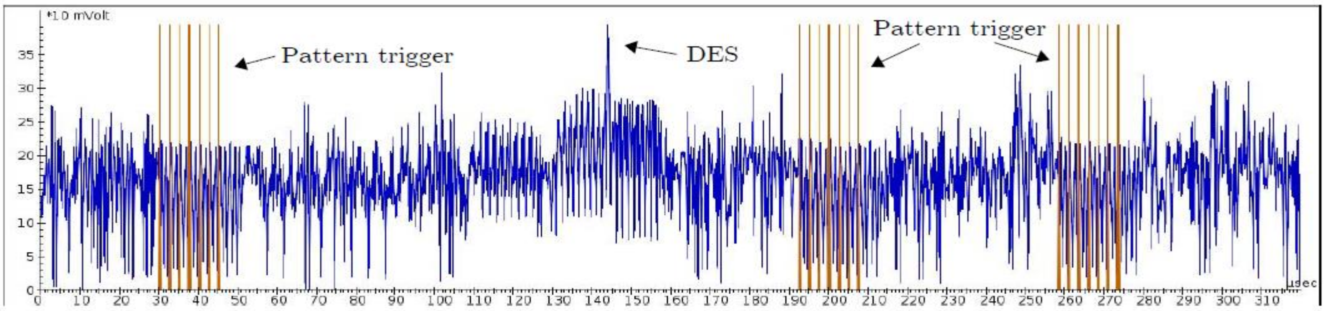


Fig. 17: Results of simulations [6]

The reader can observe that the DES is operated at a time equal to 144 µs where the waveform gets its higher value. However, there is another pattern about 144µs earlier that seems to be unique and is established as the pattern trigger for the execution to begin. The setup used for this procedure is shown below.

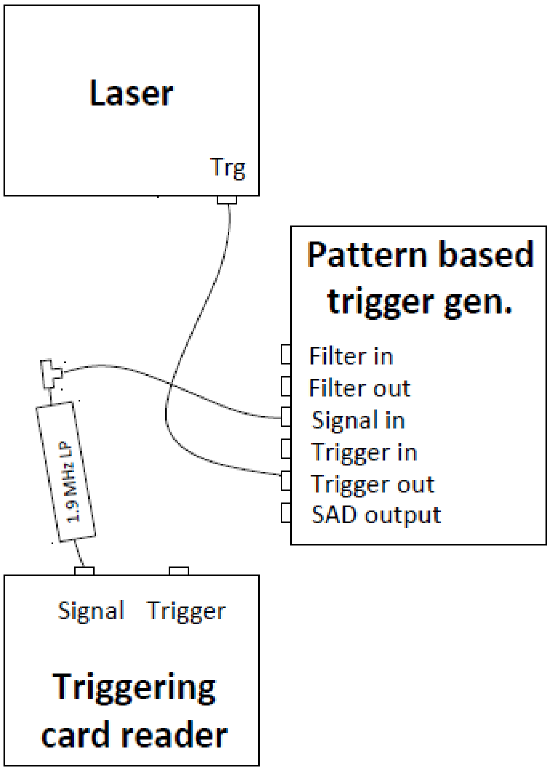


Fig. 18: Laser attack equipment [6]

With these tools, it’s easy to scan the chip to find the best combination between time and location. The results are indicators for the best possibilities for the laser attack to succeed. As shown in Fig. 19, the orange/red areas are more vulnerable to the laser beams in comparison to the green/blue ones.

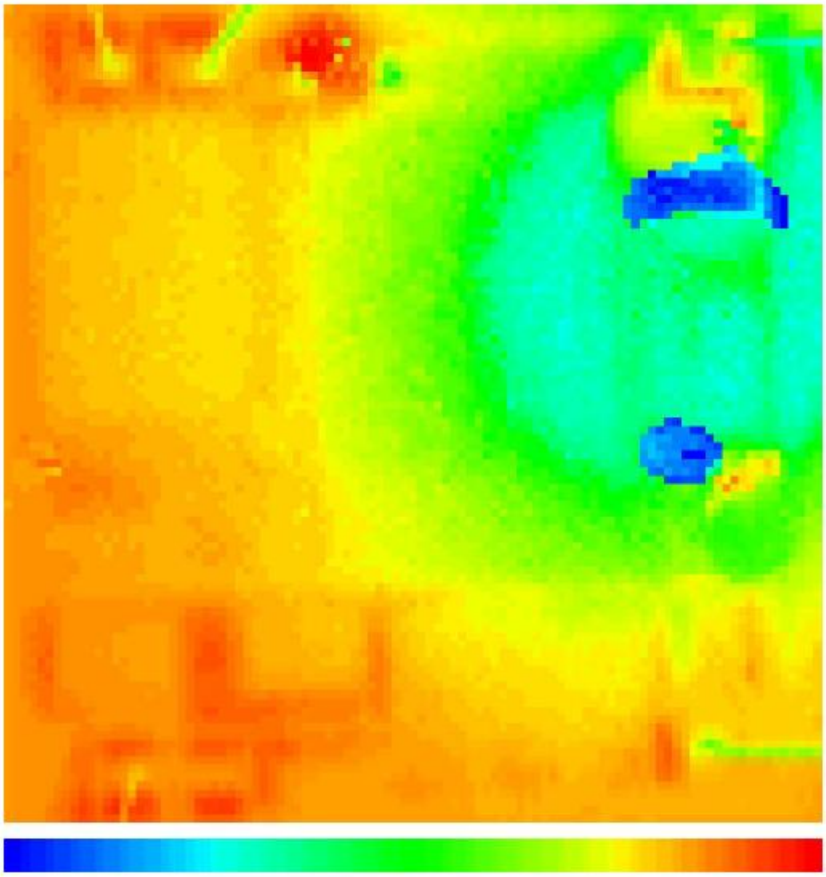


Fig. 19: Optical Laser Beam Current [6]

3.3 Electromagnetic Fault Injection Attack

Smart cards, and generally any kind of cryptographic chips, are partitioned in various sections. One of these sections is the cryprograchic coprocessor which consists and the most crucial part. Taking advantage of the EM channel and the EM emissions, someone is able to gain important information and utilize it for his own benefit as [7] introduced first. As [8] refers, an adversary might target either analogue or digital modules of the chip. In the first case the invasion occurs using a continuous wave of sinusoidal form aiming in disturbing power operation conditions. In the second case, the EM signal is more impetuous and its destination is a clock signal so as to alter the module’s logic. The equipments used for each case are shown respectively in the two images below.

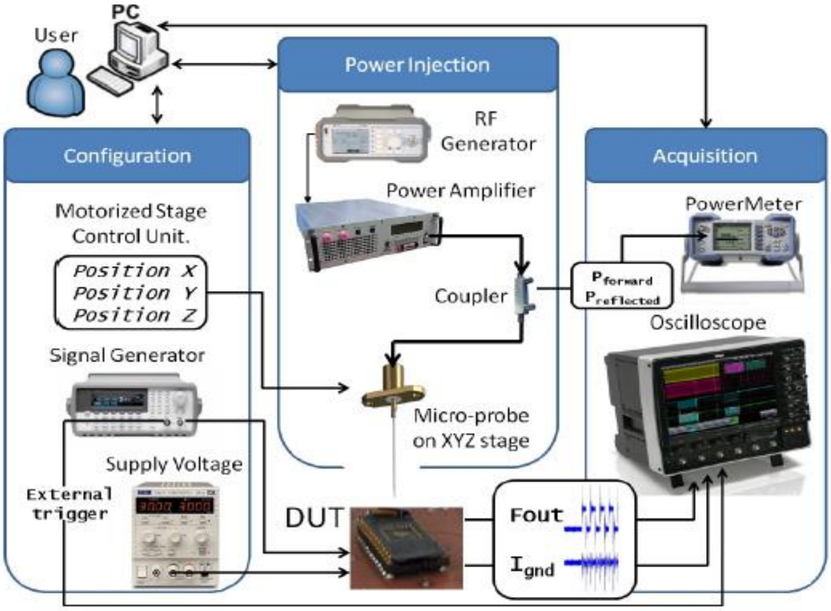


Fig. 20: Direct Power Injection Platform [8]

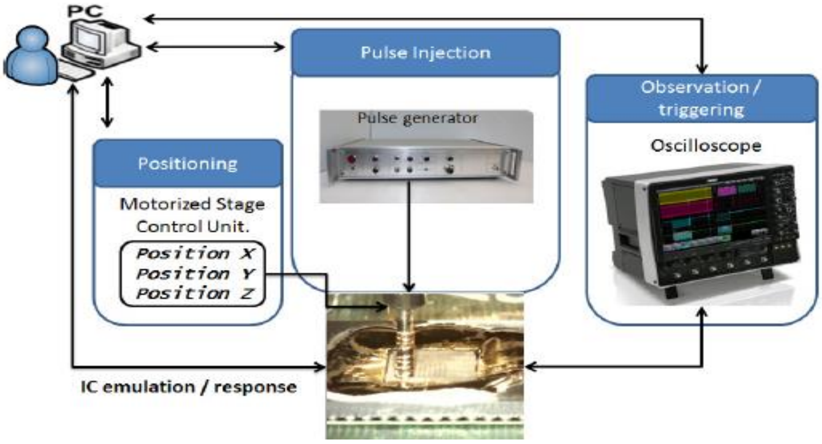


Fig. 21: EM Pulse Injection Platform [8]

It is apparent that in the case of digital EM intervasions, the equipment that is utilized is way more simple.

Altough EM FIAs target various purposes modules such as Ring Oscillators, Random Number Generators, and General Purpose CPUs, in this section it is worthy of mentioning an attack in an AES [9] that was implemented in a Xilinx FPGA and consists of the basic modules that are mentioned in Chapter 2. The invasion took place by subjecting the whole surface of the chip in EM waves with deviation step equal to the probe's diameter. As a result, a total of 900 spots each implanted 1000 times during the last round of the encryption process. In the next figure the results are shown. It is easy to claim that the most crucial faults are injected in areas that are also having crucial role in the encryption process(e.g. Round Execution and Key Expansion).

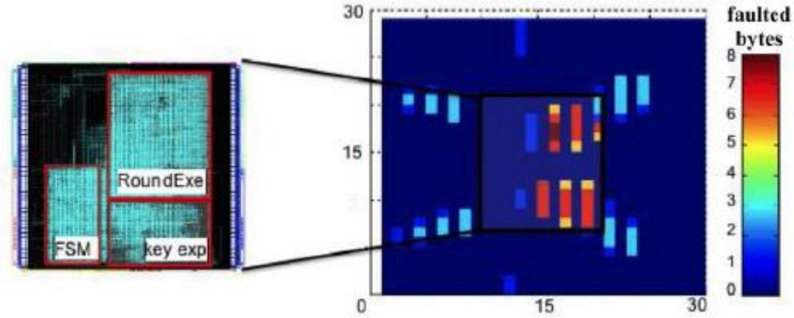


Fig. 22: Faulted Bytes [8]

Chapter 4. Countermeasures

After discussing some FIAs of great consequence on the regular functions of crypto chips, the time is ripe for examining intelligent ways for protecting AES processors against every possible malicious intrusion.

4.1 Re-encryption stage added on the AES algorithm

Writers of [11] proposed a slight modification in the encryption and the decryption process of the AES algorithm. This idea is implemented by dividing the algorithm in two parts. In the first part, Bytes Substitution and Shift Rows procedures take place while the second part is responsible for the Mix Columns and Add Round Key operations. This division aims to provide a reassurance of the invasion or not of any fault. Moreover, it is more easy for the tester to check half of the utilized modules every time and detect the actual location of the fault. As Fig. 23 shows, with this technique, there are notable savings in time as the fault detection modules operate parallel with the encryption algorithm. Precisely, when encryption takes place in the first half AES round, re-encryption is operated in the second one and vice versa. Every half round is done in one clock cycle and with the beginning of a new clock cycle there is the alteration between encryption and re-encryption. The required clock cycles for this process to be completed are 22. This number, at first sight may seem very large but considering the cost for checking the chip in a different moment, is very beneficial.

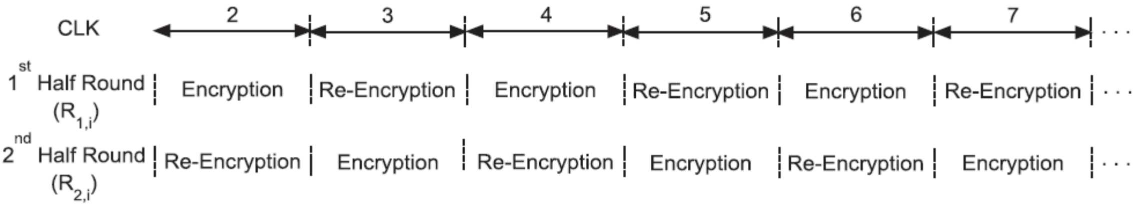


Fig. 23: Rounds of encryption and re-encryption [11]

For further facilitation, authors proposed a different hardware implementation of each half round. The ShiftRows transformation has no effect on the SubBytes outputs, which are defined by a LUT, since it executes only a cyclical left shifts of the state's last three rows. SubBytes and ShiftRows are then utilizing the same module. (Fig. 24)

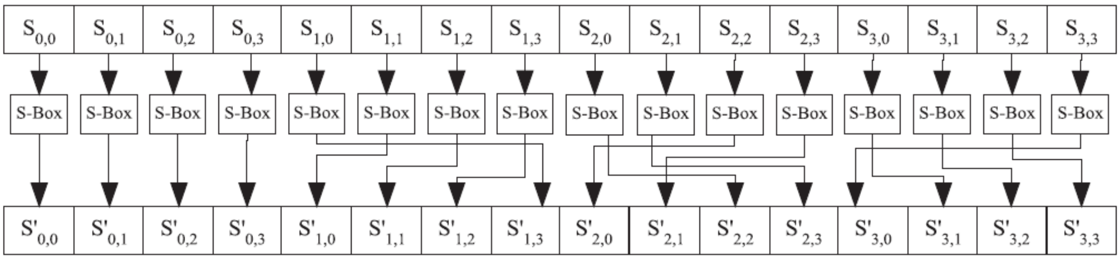

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$ $S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$ $S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$ $S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
S-Box
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$ $S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$ $S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$ $S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig. 24: S-box and Shift Rows Combination [11]

For the second half round and MixColumns, the array in Fig. 8 is used which can transformed in the next equations.

$$\begin{aligned}
s'_{0,j} &= \left(02 \cdot s_{0,j}\right) \oplus \left(03 \cdot s_{1,j}\right) \oplus s_{2,j} \oplus s_{3,j} \\
s'_{1,j} &= s_{0,j} \oplus \left(02 \cdot s_{1,j}\right) \oplus \left(03 \cdot s_{2,j}\right) \oplus s_{3,j} \\
s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus \left(02 \cdot s_{2,j}\right) \oplus \left(03 \cdot s_{3,j}\right) \\
s'_{3,j} &= \left(03 \cdot s_{0,j}\right) \oplus s_{1,j} \oplus s_{2,j} \oplus \left(02 \cdot s_{3,j}\right)
\end{aligned} \qquad (1)$$

Using the fact that 03 = 02 XOR 01, the set of equations (1) is written as

$$\begin{aligned}
s'_{0,j} &= 02 \cdot \left(s_{0,j} \oplus s_{1,j}\right) \oplus s_{1,j} \oplus s_{2,j} \oplus s_{3,j} \\
s'_{1,j} &= s_{0,j} \oplus 02 \cdot \left(s_{1,j} \oplus s_{2,j}\right) \oplus s_{2,j} \oplus s_{3,j} \\
s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus 02 \cdot \left(s_{2,j} \oplus s_{3,j}\right) \oplus s_{3,j} \\
s'_{3,j} &= s_{0,j} \oplus s_{1,j} \oplus s_{2,j} \oplus 02 \cdot \left(s_{3,j} \oplus s_{0,j}\right)
\end{aligned} \qquad (2)$$

where 02 in hardware is implemented by 3 XOR gates and in the Fig. 25 is symbolized as xtime function.

Finally, the MixColumn is combined, in a common component, with the AddRoundKey operation which XOR-es the state array after MixColumn with the round key that is generated at the key generation phase.

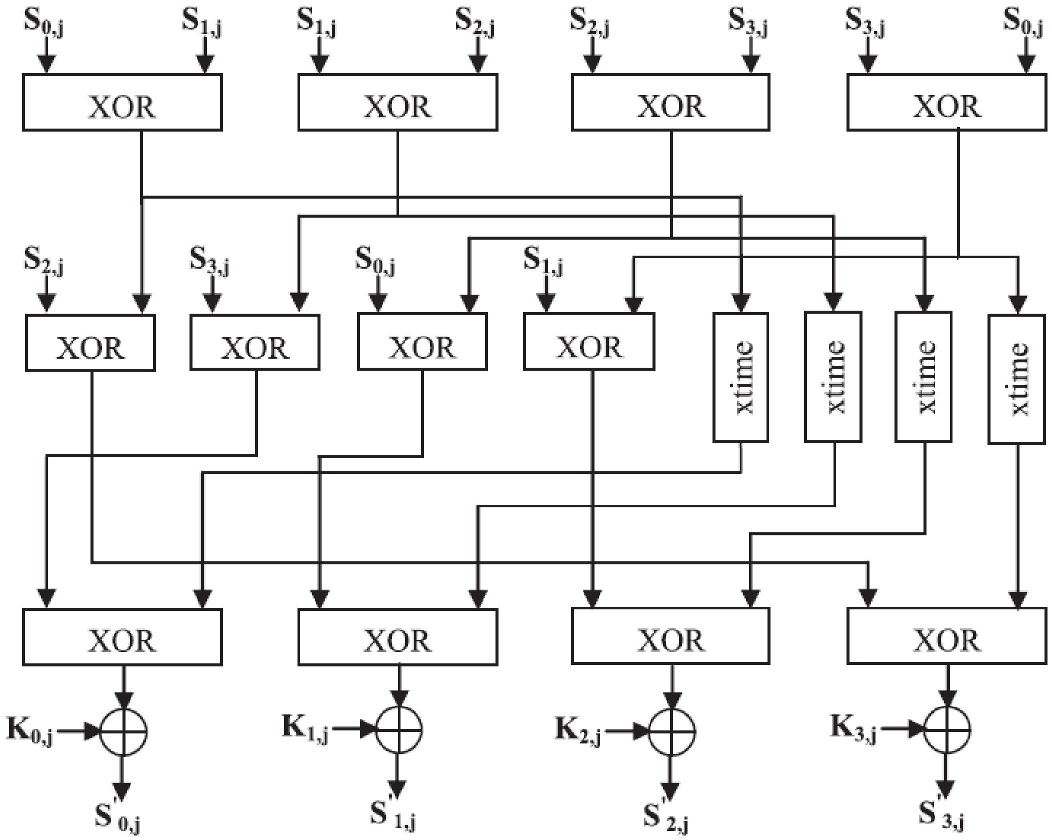


Fig. 25: MixColumn and AddRoundKey [11]

4.2 High throughput fault-resilient AES architecture

As a means of achieving the specified end of a high throughput design for all kind of faults detection, including the permanent ones, a parallel architecture is proposed[12]. For this purpose, 16 S-Boxes are utilized, each for every byte of the state (Fig. 26). Then, the results of the 4 S-Boxes are combined and Shift Rows takes place in an internal procedure as in Section 4.1. Before the following steps (Mix Columns and Add Round Key) a pipeline register is applied with the use of some comparators in order to compare the value of the expected ciphertext with the value of the real one in that phase of the algorithm which is the most crucial for faults injection.

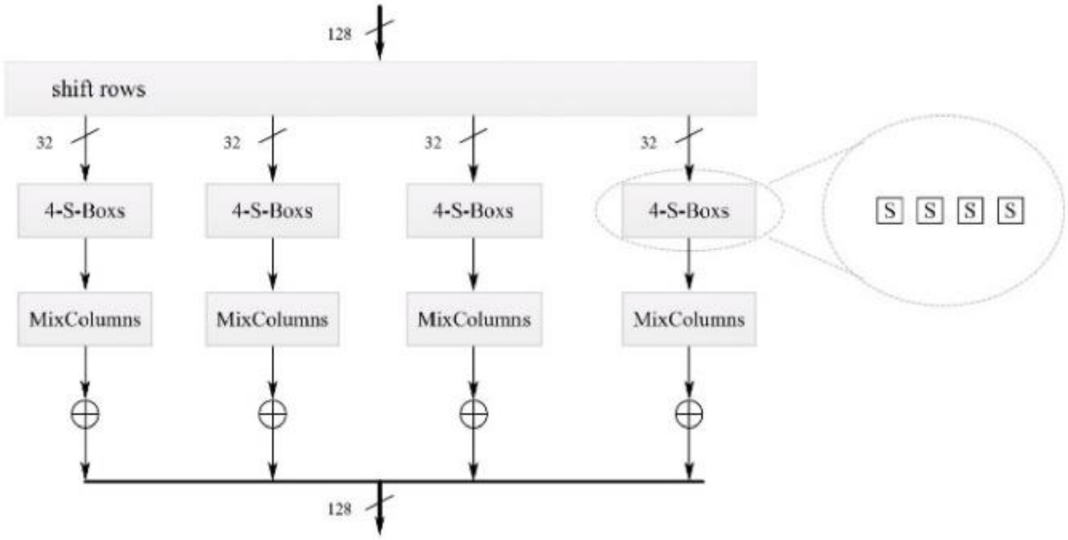
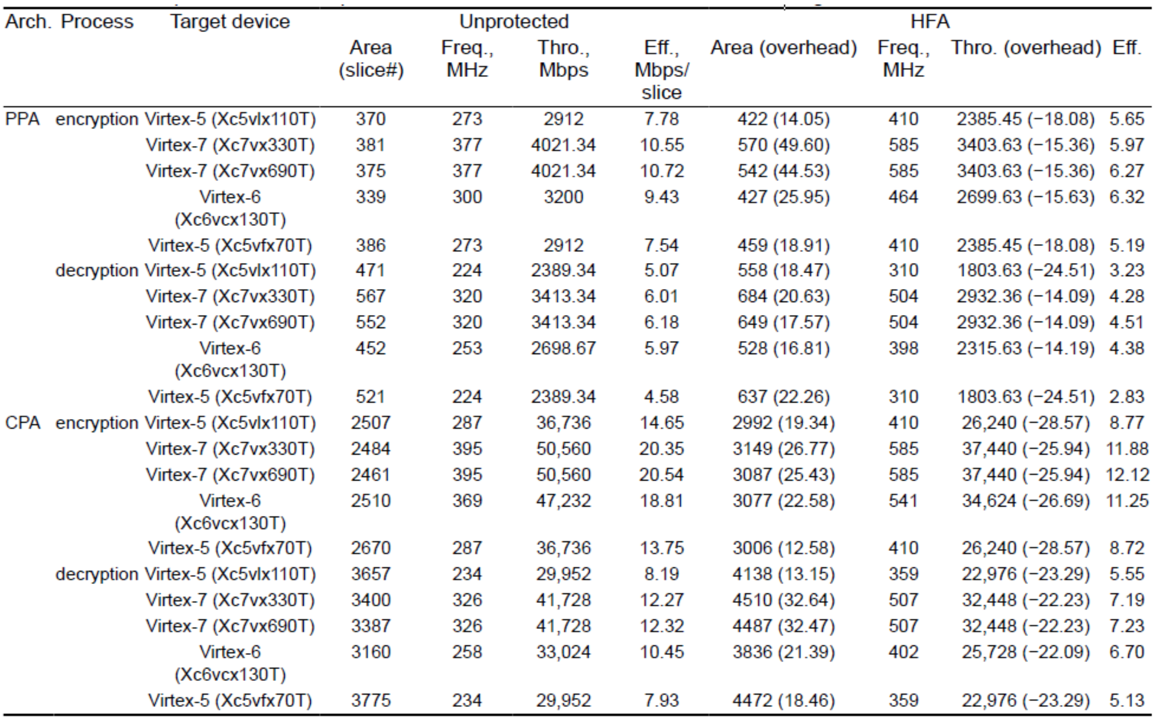


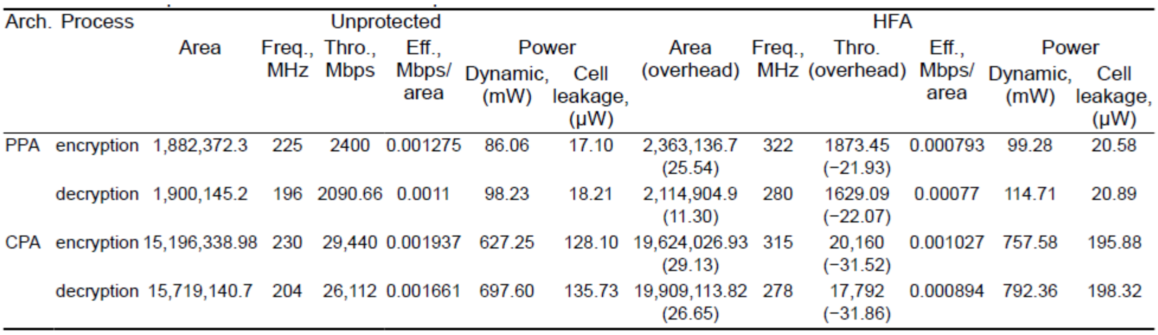
Fig. 26: Parallel Architecture [12]

By implementing this design on different FPGAs (Fig. 27) and ASICs (Fig. 28), the following results come out.



Arch.	Process	Target device	Unprotected				HFA			
			Area (slice#)	Freq., MHz	Thro., Mbps	Eff., Mbps/ slice	Area (overhead)	Freq., MHz	Thro. (overhead)	Eff.
PPA	encryption	Virtex-5 (Xc5vlx110T)	370	273	2912	7.78	422 (14.05)	410	2385.45 (−18.08)	5.65
		Virtex-7 (Xc7vx330T)	381	377	4021.34	10.55	570 (49.60)	585	3403.63 (−15.36)	5.97
		Virtex-7 (Xc7vx690T)	375	377	4021.34	10.72	542 (44.53)	585	3403.63 (−15.36)	6.27
		Virtex-6 (Xc6vcx130T)	339	300	3200	9.43	427 (25.95)	464	2699.63 (−15.63)	6.32
		Virtex-5 (Xc5vfx70T)	386	273	2912	7.54	459 (18.91)	410	2385.45 (−18.08)	5.19
	decryption	Virtex-5 (Xc5vlx110T)	471	224	2389.34	5.07	558 (18.47)	310	1803.63 (−24.51)	3.23
		Virtex-7 (Xc7vx330T)	567	320	3413.34	6.01	684 (20.63)	504	2932.36 (−14.09)	4.28
		Virtex-7 (Xc7vx690T)	552	320	3413.34	6.18	649 (17.57)	504	2932.36 (−14.09)	4.51
		Virtex-6 (Xc6vcx130T)	452	253	2698.67	5.97	528 (16.81)	398	2315.63 (−14.19)	4.38
		Virtex-5 (Xc5vfx70T)	521	224	2389.34	4.58	637 (22.26)	310	1803.63 (−24.51)	2.83
CPA	encryption	Virtex-5 (Xc5vlx110T)	2507	287	36,736	14.65	2992 (19.34)	410	26,240 (−28.57)	8.77
		Virtex-7 (Xc7vx330T)	2484	395	50,560	20.35	3149 (26.77)	585	37,440 (−25.94)	11.88
		Virtex-7 (Xc7vx690T)	2461	395	50,560	20.54	3087 (25.43)	585	37,440 (−25.94)	12.12
		Virtex-6 (Xc6vcx130T)	2510	369	47,232	18.81	3077 (22.58)	541	34,624 (−26.69)	11.25
		Virtex-5 (Xc5vfx70T)	2670	287	36,736	13.75	3006 (12.58)	410	26,240 (−28.57)	8.72
	decryption	Virtex-5 (Xc5vlx110T)	3657	234	29,952	8.19	4138 (13.15)	359	22,976 (−23.29)	5.55
		Virtex-7 (Xc7vx330T)	3400	326	41,728	12.27	4510 (32.64)	507	32,448 (−22.23)	7.19
		Virtex-7 (Xc7vx690T)	3387	326	41,728	12.32	4487 (32.47)	507	32,448 (−22.23)	7.23
		Virtex-6 (Xc6vcx130T)	3160	258	33,024	10.45	3836 (21.39)	402	25,728 (−22.09)	6.70
		Virtex-5 (Xc5vfx70T)	3775	234	29,952	7.93	4472 (18.46)	359	22,976 (−23.29)	5.13

Fig. 27: FPGA results [12]



Arch.	Process	Unprotected						HFA					
		Area	Freq., MHz	Thro., Mbps	Eff., Mbps/ area	Power		Area (overhead)	Freq., MHz	Thro. (overhead)	Eff., Mbps/ area	Power	
						Dynamic, (mW)	Cell leakage, (μW)					Dynamic, (mW)	Cell leakage, (μW)
PPA	encryption	1,882,372.3	225	2400	0.001275	86.06	17.10	2,363,136.7 (25.54)	322	1873.45 (−21.93)	0.000793	99.28	20.58
	decryption	1,900,145.2	196	2090.66	0.0011	98.23	18.21	2,114,904.9 (11.30)	280	1629.09 (−22.07)	0.00077	114.71	20.89
CPA	encryption	15,196,338.98	230	29,440	0.001937	627.25	128.10	19,624,026.93 (29.13)	315	20,160 (−31.52)	0.001027	757.58	195.88
	decryption	15,719,140.7	204	26,112	0.001661	697.60	135.73	19,909,113.82 (26.65)	278	17,792 (−31.86)	0.000894	792.36	198.32

Fig. 28: ASIC results [12]

The main difference between the Partial Pipeline Architecture (PPA) and the Complete Pipeline Architecture (CPA) is that in PPA there only two stages of the above-refered pipelining with the results of the second stage to be feed in the first stage and this operation is repeated for each round of encryption or decryption. Contrary to this, CPA is similar to PPA with the additional feature that there is a pipelining stage that interconnects two consecutive rounds(Fig. 29 and 30 respectively).

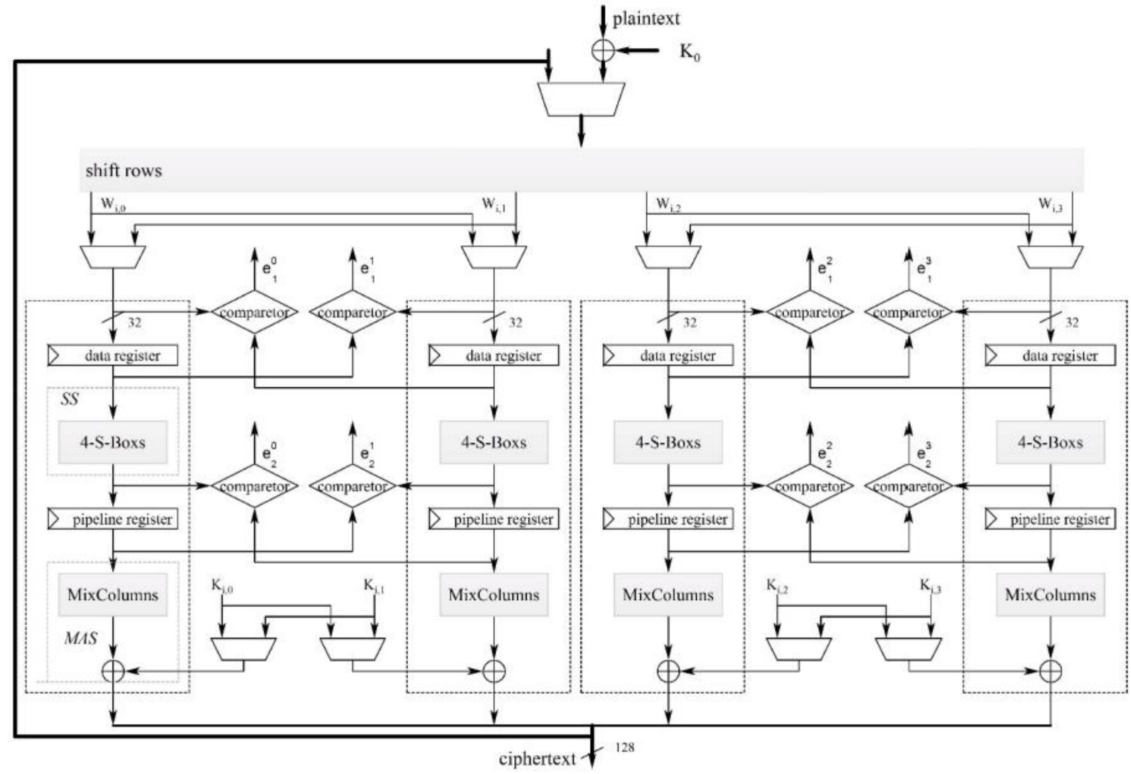


Fig. 29: PPA [12]

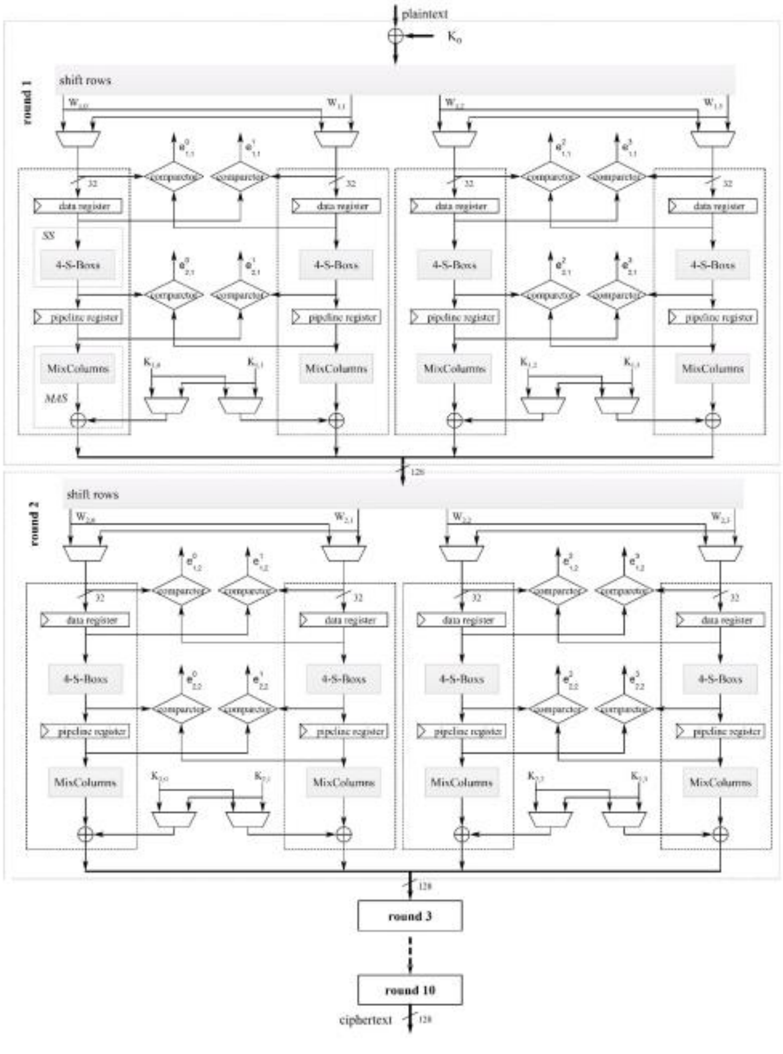


Fig. 30: CPA [12]

Chapter 5 Observations

In Chapter 3 some highly effective techniques for fault injection attacks were mentioned. In the first one, the adversary inserts a glitch in the clock chain aiming at passing wrong data to the encryption process so the receiver of the message doesn't get the right ciphertext. Similar to the clock glitch, writers of the specific research, try to insert a power glitch using the same data set that was used in their previous experiment. Comparing these two methods, an important conclusion is drawn. In a percentage of about 70%, the exact same ciphertext is exported and the same time channels are utilized. This result provides a great advantage for further research in both faults injection and detection.

The next method presents an optical attack using laser beams. Two experiments are taking place, one in the front side and one in the backside of the chip. With the appropriate laboratory equipment and simulations, the most vulnerable and important coordinates and modules, in a larger scale, are revealed. In these modules, the attack is more possible to be achieved.

In the last one, in a similar way as above, an EM invasion is introduced. The most crucial areas of the chip are again found and indicate where the fault should be take place.

Subsequently, we discussed two main ways of detecting fault injections, one relative to adding a re-encryption stage in the algorithm and another one using a high-throughput architecture.

In the first case, each round of the original AES algorithm is divided in two rounds. The first half round consists of SubBytes and ShiftRows routines and the second round of MixColumns and AddRoundKey. According to the researchers that proposed this method, when the first half round operates encryption, the second half round is responsible for the re-encryption, in which the comparison of the two ciphertexts happens in order to find differences and faults by extension, and vice versa.

The use of a high-throughput architecture of the algorithm consists another medium on the faults detection field. Within this approach, there are two subsections. A partial pipeline architecture which inserts a pipeline register and comparators between ShiftRows and MixColumns and a complete pipeline architecture in which pipeline registers connect the rounds, are utilized.

Chapter 6 Conclusion

Among the various challenges that chip manufacturing faces, HTs have emerged the last decades and consist one of the main threats. Significantly, the last few years, the topic has came up in the surface and more and more researches have been arised.

While HTs seem to appear in various forms, one of the most serious ones is FIAs that aim in the physical layout of the chip. Clock and power glitches, laser beams and electromagnetic interference are some instances that are presented in this thesis. By taking advantage of the physical vulnerability of the device, any adversary can induce with the expedient way a fault that alters its operation or steals the cryptography key and succeeds access to the secret messages that are transferred in it.

Despite the brief appearance of HTs, some efficient techniques for their detection have been proposed as referred in the Chapter 4. These methods modify some steps of the AES algorithm and achieve to detect any HT with a success rate over 98%. On this account, they consist a reference and inspiration for further research to face any hardware threat conclusively.

References

[1] Xiao, Kan, et al. "Hardware trojans: Lessons learned after one decade of research." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 22.1 (2016): 1-23.

[2] AES algorithm and its Hardware Implementation on FPGA- A step by step guide | by Gourav Saini | Medium

[3] Tehranipoor, Mohammad, and Farinaz Koushanfar. "A survey of hardware trojan taxonomy and detection." *IEEE design & test of computers* 27.1 (2010): 10-25.

[4] Atreya, Mohan, "Introduction to Cryptography," pp. 1-7.

[5] Dutertre, Jean-Max, et al. "Review of fault injection mechanisms and consequences on countermeasures design." *2011 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*. IEEE, 2011.

[6] Bozzato, Claudio, Riccardo Focardi, and Francesco Palmarini. "Shaping the glitch: optimizing voltage fault injection attacks." *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2019): 199-224.

[7] Quisquater, Jean-Jacques, and David Samyde. "Electromagnetic analysis (ema): Measures and counter-measures for smart cards." *International Conference on Research in Smart Cards*. Springer, Berlin, Heidelberg, 2001.

[8] Maistri, Paolo, et al. "Electromagnetic analysis and fault injection onto secure circuits." *2014 22nd International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2014.

[9] Dehbaoui, Amine, et al. "Electromagnetic transient faults injection on a hardware and a software implementations of AES." *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2012.

[10] Zussa, Loic, et al. "Power supply glitch induced faults on FPGA: An in-depth analysis of the injection mechanism." *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*. IEEE, 2013.

[11] Mestiri, Hassen, et al. "A high-speed AES design resistant to fault injection attacks." *Microprocessors and Microsystems* 41 (2016): 47-55.

[12] Sheikhpour, Saeide, Ali Mahani, and Nasour Bagheri. "High throughput fault-resilient AES architecture." *IET Computers & Digital Techniques* 13.4 (2019): 312-323.

[13] X. Wang, M. Tehranipoor, and J. Plusquellic, ''Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions,'' Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 08), IEEE CS Press, 2008, pp. 15-19.

[14] Van Woudenberg, Jasper GJ, Marc F. Witteman, and Federico Menarini. "Practical optical fault injection on secure microcontrollers." *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2011.