IST-2002-507932

ECRYPT

European Network of Excellence in Cryptology

Network of Excellence

Information Society Technologies

D.VAM.4

Electromagnetic Analysis and Fault Attacks: State of the Art

Due date of deliverable: 31 May 2005
Actual submission date: 31 May 2005

Start date of project: 1 February 2004 Duration: 4 years

Lead contractor: UCL Crypto Group

Revision 3

Project co-funded by the European Commission within the 6th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Electromagnetic Analysis and Fault Attacks: State of the Art

Editor
François-Xavier Standaert (UCL)

Contributors
Lejla Batina (KUL), Elke De Mulder (KUL), Kerstin Lemke (RUB),
Stefan Mangard (GRAZ), Elisabeth Oswald (GRAZ), Gilles Piret (UCL).

31 May 2005
Revision 3

The work described in this report has in part been supported by the Commission of the European Communities through the IST program under contract IST-2002-507932. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Contents

Chapter 1

Introduction

Since their introduction in the late nineties, side-channel and fault attacks have attracted significant attention within the cryptographic community. These kind of attacks belong to implementation attacks which exploit the additional leakage that is caused by the physical nature of cryptographic devices. This additional leakage consists of physical observables (as timing, power consumption and electromagnetic (EM) emanation) in case of side-channel cryptanalysis and it consists of erroneous computations in case of fault analysis. The efficiency and simplicity of these cryptanalytic methods was amazing at that time. The cryptographic research that traditionally focussed on mathematical cryptanalysis has consequently also moved to this new research area of implementation attacks. A large body of theoretical and experimental work has been carried out in order to analyze the possibility to defeat cryptographic implementations using physical mechanisms. As a consequence of these threats, a number of countermeasures have also been proposed.

Regarding side-channel attacks, the timing leakage is probably the simplest one to understand. It is also the threat that can be the most easily counteracted. Because of that, it was not of relevance for this report. This report also does not cover power analysis in great detail. This decision is motivated by the fact that the measurement set-ups used for power analysis are now widely known.

On the other hand, there are only a few publications on measurement set-ups used for EM emanation. It is reported in earlier works that the nature of EM emanation is manifold and not well understood, yet. In addition, the countermeasures for power analysis and electromagnetic analysis are commonly assumed to be similar, though evidences are missed until now. Similarly, practical research on fault analysis is currently mainly driven by manufacturers, and not by the cryptographic research community. We aim to join the theoretical scenarios that are invented by academia with the practical experience that is reported in the scientific literature.

In general, while a number of experiments have been proposed and demonstrated the strength of these attacks, numerous questions remain about what is actually feasible by a knowledgable opponent. The look for optimal post-processing tools and the need of efficient countermeasures is certainly as critical. In this report, we consequently analyze the state of the art of fault and electromagnetic attacks. The choice of these two types of techniques was mainly motivated by the fewer number of publications in the field and the need of further research.

In the first part of the report, electromagnetic analysis attacks are described and illustrated through the measurement setups of two ECRYPT partners, namely KU Leuven and TU Graz. A number of techniques are presented for the measurements of electromagnetic radiation and for the post processing of these measurements. One section is devoted to the usual models in use for the analysis and implementation of such attacks. Countermeasures against electromagnetic analysis are finally briefly described.

The second part of the report describes fault attacks on the basis of previously published works. First, the different types of (known) physical sources of faults are listed and defined. The models in use for the devices, adversaries and faults are also introduced. Then, the algorithmic consequences of faults are analyzed in the context of a number of cryptosystems. Finally, a list of protections are suggested.

Chapter 2

Classification of attacks

Traditionally, the mathematical cryptanalysis assumes that the cryptographic device is an abstract machine that allows only the input and output data of the cryptographic algorithm to be used for cryptanalysis. But, in reality other attacks are possible if the adversary has physical access to the cryptographic device or at least to the near-by environment. These are *Implementation Attacks* which target the cryptographic device itself. These attacks can be *Active Attacks* which range from changing the environmental conditions to the physical penetration of the cryptographic device. Another class of attacks acts in a passive way, just by observing the inherent leakage of the cryptographic device. *Passive Attacks* are even more dangerous as they do not leave any damage to the cryptographic device that can be recognized later on. Passive Attacks just use the cryptographic device in its intended environment and can obtain cryptographic keys by leaked information. This additional information used can be the power consumption of the device, electromagnetic radiation, timing information on the cryptographic service or error messages obtained. Note, that combinations of active and passive approaches are possible, e.g. an active attack can be a preparation step for a passive attack. In Figure 2.1 we present our general picture of Implementation Attacks by starting with distinguishing active and passive attacks first.

2.1 Active Attacks

Active attacks target the physical security of the device. We distinguish three kinds of active attacks:

- Non-invasive attacks. Changes towards extreme environmental conditions put the cryptographic device under physical stress which may lead to an erroneous behavior of the device. Malfunction can be caused e. g. by short-time pulses in the supply voltage or by freezing down the environmental temperature. Though there is a certain risk of a permanent destruction of the target circuit, generally non-invasive environmental attacks do not leave specific damage to the cryptographic device. The first scenarios of Fault Analysis make use of non-invasive approaches.
- Semi-invasive attacks. As the first step for opening of the cryptographic device, the package material has to be removed. If this is already sufficient for the specific type

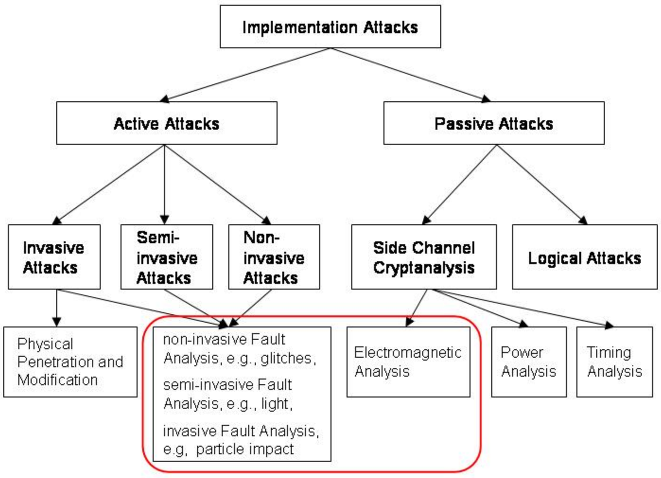
Implementation Attacks
Active Attacks
Passive Attacks
Invasive Attacks
Semi-invasive Attacks
Non-invasive Attacks
Side Channel Cryptanalysis
Logical Attacks
Physical Penetration and Modification
non-invasive Fault Analysis, e.g., glitches,
semi-invasive Fault Analysis, e.g., light,
invasive Fault Analysis, e.g, particle impact
Electromagnetic Analysis
Power Analysis
Timing Analysis

Figure 2.1: Classification of Implementation Attacks. The red frame encloses Fault Analysis and Electromagnetic Analysis that are covered in this survey.

of attack we denote as semi-invasive. The costs of the necessary equipment are still moderate and the removal of the packaging can be done in a standard laboratory.

- Invasive attacks. Direct connections are made (i. e. to an internal bus line) to read out security relevant data within the cryptographic device. Nowadays, active attacks as physical probing and physical manipulation demand for semi-conductor equipment which is available in specialized laboratories only.

Note that a physical secure device does not offer external interface functions to the program and data memory in the end-user environment, e.g., debugging interfaces which are still available at standard microcontrollers.

2.2 Passive Attacks

We distinguish two kinds of passive attacks:

- Side Channel Cryptanalysis. It makes use of the inherent physical leakage of the cryptographic device as an additional information obtained for cryptanalysis. The tools needed for the measurement of this additional information are still in a low to moderate budget range. Side channel techniques are covered in Chapter 3 with a strong focus on electromagnetic analysis.
- Logical attacks. Logical Attacks make use of the external logical functions of the cryptographic device and look for specific software or protocol bugs that can be exploited. Direct access to the cryptographic device is not necessary. Therefore, only software tools are needed. Logical Attacks as the famous Bleichenbacher attack on the RSA encryption standard PKCS#1 are outside of the scope of this report.

Chapter 3

Electromagnetic Analysis (EMA)

There are several partners within the ECRYPT consortium that have developed the know-how and setups to conduct electromagnetic analysis (EMA). In this chapter we discuss some of these setups that are exemplary for the others. These setups are similar to the setups used in previous work [78], [38] and [5]. With these setups we have also verified the results of the previous work. Consequently, we elaborate on the state-of-the-art in EMA by using our own results in the followings sections, instead of referring to previous work only.

First, the previous work will be discussed in Sect. 3.1. Sect. 3.2 presents these measurement setups in detail. We sketch the models that are currently used to perform EMA in Sect. 3.3. Results of attacks that have been conducted are presented in the 3.4 and 3.5. In Sect. 3.6 we discuss different post processing techniques and we conclude the chapter by listing useful countermeasures in Sect. 3.7.

3.1 Previous Work

Electromagnetic analysis exploits information that leaks through the electromagnetic field that is produced by a device. It is well known that the US government has been aware of electromagnetic leakage since the 1950's. The resulting standards are called TEMPEST; partially declassified documents can be found in [74]. The first published papers are work of Quisquater and Samyde [78] and the Gemplus team [38]. Quisquater and Samyde showed that it is possible to measure the electromagnetic radiation from a smart card. Their measurement setup consisted of a sensor which was a simple flat coil, a spectrum analyzer or an oscilloscope and a Faraday cage. Quisquater also introduced the terms Simple EMA (SEMA) and Differential EMA (DEMA). The work of Gemplus deals with experiments on three algorithms: DES, RSA and COMP128. They observed the feasibility of EMA attacks and compared them with power analysis attacks in favor of the first. Namely, EM emanation can also exploit local information and, although more noisy, the measurements can be performed from a distance. This fact broadens the spectrum of targets to which side-channel attacks can be applied. Of concern are not only smart cards and similar tokens but also SSL accelerators and many other cryptographic devices.

According to Agrawal *et al.* there are 2 types of emanations: intentional and unintentional [3, 5]. The first type results from direct current flows. The second type is caused by various couplings, modulations (AM and FM), etc. The two papers mentioned above deal exclusively with intentional emanations. To the contrary, the real advantage over other side-channel attacks lies in exploring unintentional emanations [3, 5]. More precisely, EM leakage consists of multiple channels. Therefore, compromising information can be available even for differential power analysis (DPA) resistant devices which can be detached from the measurement equipment.

Besides carefully exploring all available EM emanations an attacker can also focus on a combination of two or more side-channels. Agrawal *et al.* defined these so-called multi-channel attacks in which the side-channels are not necessarily of a different kind [4]. For example, they discussed combined power and EM analysis but also multi-channel DPA attacks. The latter uses a CMOS leakage model and the maximum-likelihood principle for performing and analyzing. Another example of a multi-channel attack is introduced by Walter and Thompson in [94]; they were the first to combine power and timing analysis.

Mangard also showed that near-field EM attacks can be conducted with a simple hand-made coil in [68]. Besides that he showed that measuring the far-field emissions of a smart card connected to a power supply unit also suffices to determine the secret key used in the smart card. Carlier *et al.* showed that EM side-channels from an FPGA implementation of AES can be effectively used by an attacker to retrieve some secret information in [25]. They worked close to the FPGA and in this way were able to get rid of the effects of other computations made at the same time.

Up to now, most papers on EMA applied similar techniques as power analysis while apparently much more information is available to be explored. It is likely that future work will also deal with combinations of EMA with other side-channel attacks.

3.2 EM Measurement Setups

Measurement setups for power and EM attacks are very similar in practice. In both attacks, the power dissipation of a module is recorded with a digital oscilloscope while the module performs a cryptographic operation. The only difference is the probe that is used for the attack.

Figure 3.1 shows a block diagram of the reference measurement setups. The setups consist of three main components: the cryptographic module, a digital oscilloscope and a standard PC. The PC provides some data input for a cryptographic operation to the attacked module. While the module performs this operation, the digital oscilloscope records the side-channel output of the module. Depending on which side-channel is exploited, different probes are used for the oscilloscope. After the cryptographic operation is completed, the power trace that has been recorded during this operation is transferred from the oscilloscope to the PC.

Different types of oscilloscopes can be used. There are stand-alone oscilloscopes that are connected to a PC via a GPIB cable or a LAN interface, and there are PCI cards that provide the same functionality as an oscilloscope but are directly inserted in a PC. The advantage of PCI-cards over the stand-alone oscilloscope is that the storage transfer into the memory of the PC performing the attack works significantly faster via the PCI bus than via the GPIB cable.

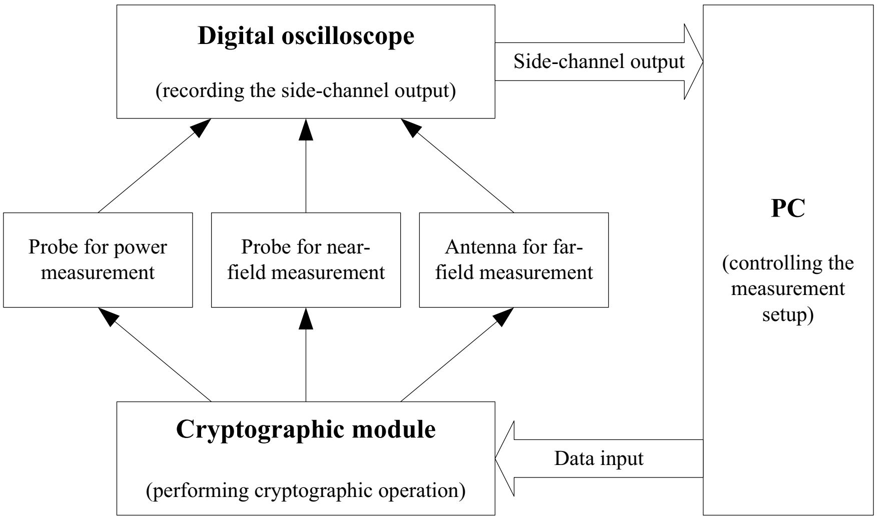
Digital oscilloscope
(recording the side-channel output)
Side-channel output
PC
(controlling the measurement setup)
Probe for power measurement
Probe for near-field measurement
Antenna for far-field measurement
Cryptographic module
(performing cryptographic operation)
Data input

Figure 3.1: Block diagram of the measurement setups used for power and EM attacks.

This greatly reduces the time that is needed for an attack. This performance gain becomes important if a high number of traces is gained or, alternatively, if many storage-intensive traces are recorded.

However, the typical stand-alone oscilloscopes have certain advantages as well. They often have a higher bandwidth and a better sensitivity than the PCI-cards. Detailed specifications of two different types of oscilloscopes are shown in Table 3.1.

Table 3.1: Specification of typical oscilloscopes used for power and EM attacks.

	LeCroy LC584	GaGe CompuScope 82G
Type of oscilloscope	Standalone device	PCI card
Interface to PC	GPIB	not necessary
Acquisition memory	2 MB	16 MB
Resolution	8 bit	8 bit
Maximum sample rate	8 GS/s	2 GS/s
Maximum bandwidth	1 GHz	400 MHz
Input impedance	10 MΩ, 11 pF or 50 Ω	1 MΩ, 25 pF or 50 Ω
Smallest input range	16 mV	200 mV

There are also different types of probes available. We have employed an active differential probe for contact-based power measurements and passive probes for the trigger signals. The specifications of these probes are shown in Table 3.2. Some of the probes that have been used for EM attacks have been self-made and therefore no detailed specification is available for them. These EM probes are discussed more detailed in Sect. 3.2.1.

In all measurement setups, a standard PC has been used to provide input data to the attacked cryptographic module and to perform the analysis of the recorded power traces.

To simplify the analysis special trigger signals can be implemented in the cryptographic software if possible.

Table 3.2: Specification of typical probes used for power and EM attacks.

	LeCroy AP034	LeCroy PP005
Type of probe	active, differential	passive
Bandwidth	1 GHz	500 MHz
Attenuation	1:1	1:10
Input capacitance	0.85 pF differential	11 pF
DC input resistance	2 MΩ	10 MΩ

The following subsections discuss different EM attack setups in detail. We show measurement setups for EM attacks in the near and in the far field. It is important to point out that the goal of the measurement setups was to prove that the attacks are possible and to get a better understanding for attacks in practice. The setups have not been optimized for characteristics such as low noise and high bandwidth.

3.2.1 Measurement Setups for EM Attacks in the Near Field

We have conducted attacks based on measuring the electromagnetic field surrounding a device. Every current flowing in a device affects the electromagnetic field. Hence, similar to the power dissipation, the electromagnetic field also depends on the data that is processed by a device.

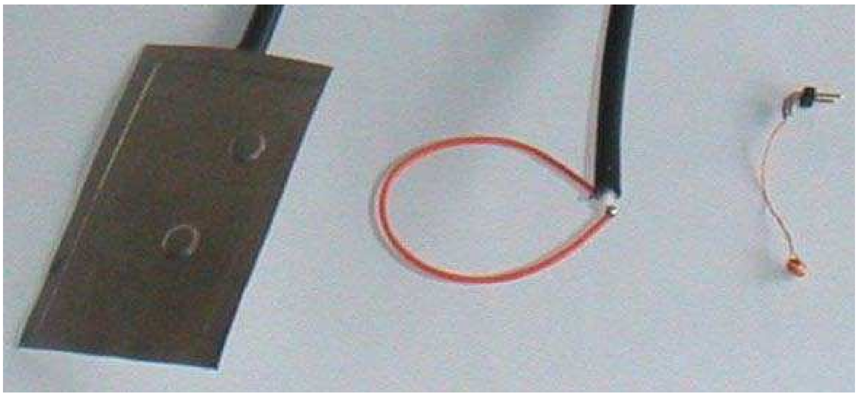

Figure 3.2: Probes used for EM attacks in the near field.

This property can be exploited either in the near or in the far field. The near field is the electromagnetic field in the immediate surrounding of a device, while the far field starts at a distance of several meters.

Measurement setups for attacks in the near field are very similar to setups used to analyze the electromagnetic compatibility (EMC) of a device. The EMC of a device is usually determined by measuring the electromagnetic emissions according to the IEC standard 61967 [50]. In this standard, different methods for measuring the near field of a device are specified.

However, building a measurement setup according to [50] is quite expensive. Therefore, we have used simpler setups. As demonstrated in [78] and [38], we have manually placed simple self-made coils (see Figure 3.2) close to the attacked device in order to measure the electromagnetic field.

Although the probes shown in Figure 3.2 look quite primitive, we have been able to perform successful differential EM attacks using them. The leftmost probe has been inspired by probes as they are usually employed for measurements of the electrical field. The probe consists of a metal plate that has been connected to the inner conductor of a 50 Ω coaxial cable.

The other two probes are simple coils that have been inspired by probes as they are usually employed for measurements of the magnetic field. While the probe in the middle is also based on a 50 Ω coaxial cable, the rightmost probe is used in combination with the differential probe AP034.

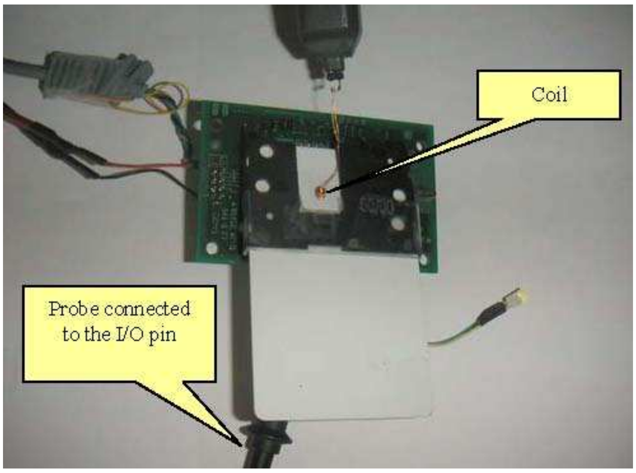

Figure 3.3: Measurement setup for the EM attacks on smart cards in the near field.

Of course, the signal that is received with an EM probe strongly depends on the position of the probe in relation to the attacked device. In fact, the position is more critical for smaller probes. Hence, in particular the rightmost probe shown in Figure 3.2 needs to be placed carefully.

Figure 3.3 shows a photo of a measurement setup based on this probe. The position of the probe was chosen such that the amplitude of the received signal was maximized. This strategy has also been used in attacks using the other probes. This setup also allows to make power analysis attacks by simply attaching a differential probe to a resistor that has been inserted in the power line of the reader.

Figure 3.4 shows a photo of a measurement setup based on this probe for a microcontroller. The microcontroller-board was self-made. The microcontroller that is used is a standard 8051-compatible microcontroller without any specific features. The board also allows to make power measurements by connecting a differential probe to the resistor that can be seen at the bottom of the photo.

Figure 3.5 shows a photo of a measurement setup based on a loop antenna and an FPGA. The probe was placed around the FPGA to ensure that the amplitude of the signal was maximized.

In all attacks conducted in the near field, a passive probe has been used to provide a trigger for the oscilloscope. Only the EM field was measured using an EM probe.

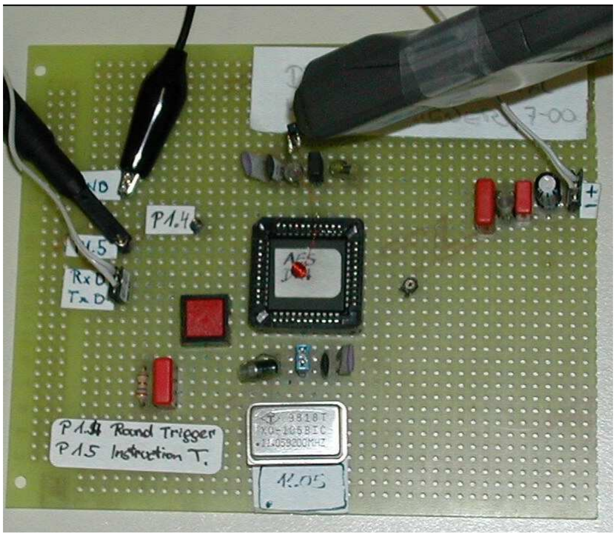

Figure 3.4: Measurement setup for EM attacks on microcontrollers in the near field.

3.2.2 Measurement Setups for EM Attacks in the Far Field

Exploiting the electromagnetic emissions of a device in the far field is usually more difficult than conducting attacks in the near field. The reason for this is that in the far field the emissions of the attacked device are typically buried in a lot of noise. A significant amount of interference is caused by radio signals and by the radiated emissions of other electronic devices that are in the reception area of the antenna used by the attacker.

The big challenge of EM attacks in the far field is therefore to detect the compromising emissions of a device among the many other signals that are received with the antenna used for the attack. Hence, the ideal equipment for an EM attack in the far field would be a superheterodyne receiver that can be tuned to a wide range of frequencies and which has a variable bandwidth.

However, such a receiver has not been available in the context of this research. Consequently, we have built a simpler measurement setup. The goal of this setup was to show that even the far field of a device contains enough information to perform successful EM attacks. In order to proof this point, we have conducted experiments in a room that was shielded against external electromagnetic radiation up to 1 GHz. The room was a Faraday cage with reflecting iron walls.

A smart card in a smart card reader and a corresponding power supply unit have been placed into the room (see Figure 3.6). The far-field emissions of this arrangement have been measured with a biconical antenna that was also in the same room (see Figure 3.7). The distance between the smart card and the antenna was several meters.

We connected the antenna (frequency range: 30 MHz to 200 MHz) via a 30 dB wideband amplifier to the digital oscilloscope—no filtering was done between the antenna and the oscilloscope. During all attacks conducted in the far field, a probe was connected to the I/O port of smart card. This probe was used as trigger for the oscilloscope.

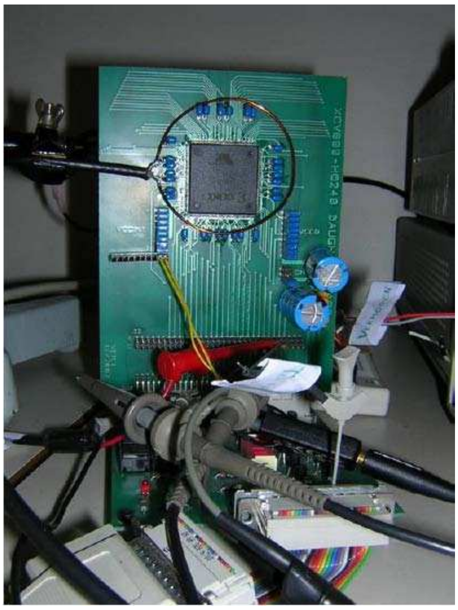

Figure 3.5: Measurement setup for EM attacks on FPGAs in the near field.

3.3 Models for EM Attacks

In this section we report on the models that we used to predict the EM radiation of the attacked device.

The current that flows during the transition of the output of a CMOS gate, causes a variation of the electromagnetic field surrounding the chip that can be monitored by for example inductive probes which are particularly sensitive to the related impulse. The electromotive force across the sensor (Lentz' law) relates to the variation of magnetic flux as follows [82]:

$$V = -\frac{\mathrm{d}\phi}{\mathrm{d}t} \quad \text{and} \quad \phi = \iint \vec{B} \cdot d\vec{A},$$

where V is the probe's output voltage, ϕ the magnetic flux sensed by probe, t is the time, $\vec{B}$ is the magnetic field and $\vec{A}$ is the area that it penetrates.

Maxwell's equation based on Ampère's law relates the magnetic field to their origin:

$$\vec{\nabla} \times \vec{B} = \mu\vec{J} + \epsilon\mu\frac{\delta\vec{E}}{\delta t},$$

where $\vec{J}$ is the current density, $\vec{E}$ is the electrical field, ϵ is the dielectric permittivity and μ is the magnetic permeability.

Two broad categories of EM emanations can be distinguished: direct emanations and indirect emanations [5]. The first category results from intentional currents. The last category of emanations originates of coupling between different components in the device and includes amplitude modulation and angle modulation, which are explained in Sect. 3.6.2.

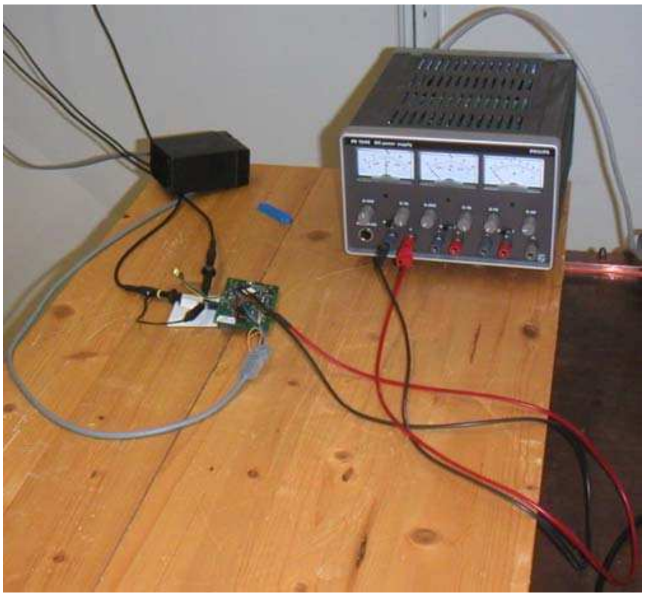

Figure 3.6: Measurement setup for EM attacks on smart cards in the far field.

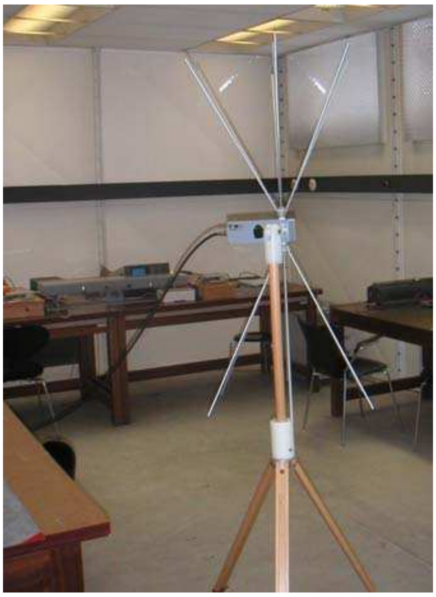

Figure 3.7: Antenna used for EM attacks in the far field.

3.3.1 The EM Leakage of a Typical Microcontroller

It has turned out that the characteristics of the power dissipation leakage of the microcontrollers (and smart cards) that we investigated have certain similarities to the characteristics of the EM leakage; both are heavily related to the Hamming weight of the processed data. In [6] they use another model for some components; in those the power dissipation of a bit influences the power dissipation of the bit before and after it, so they refine the model with this info.

Figure 3.8 shows the power dissipation of the microcontroller (see Figure 3.4) during the execution of a MOV instruction with different operands. It is clearly visible in this figure that the amplitude of the last peak is different for different operands. The amplitude depends linearly on the Hamming weight of the operand involved in the MOV instruction. In fact, the same also holds for other instructions of the microcontroller.

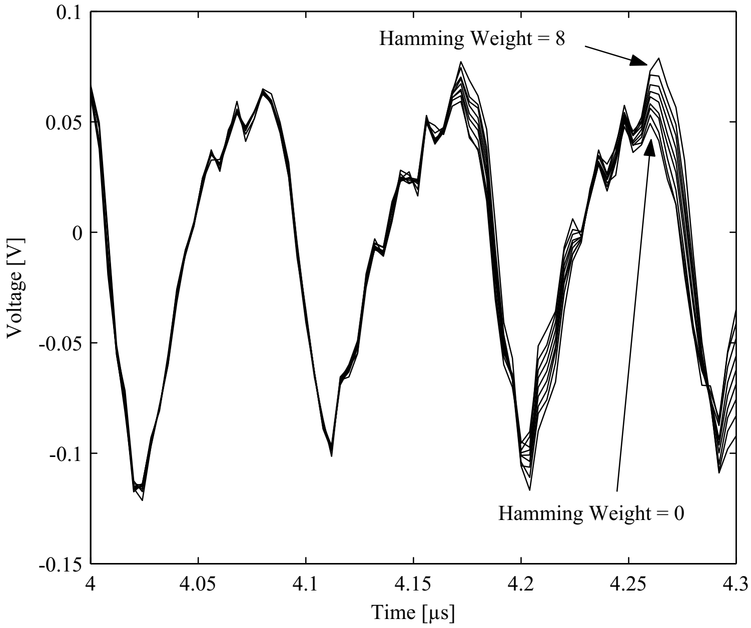

Figure 3.8: The voltage drop along the resistor of the power measurement setup depends linearly on the Hamming weight of the value that is transferred over the bus of the microcontroller.

This type of leakage is observed because the microcontroller is based on a pre-charged bus. Every time an operand of an instruction is transferred over the bus, the microcontroller leaks the Hamming weight of the operand.

The Hamming weight leakage of both devices is not restricted to contact-based power measurements. It can also be exploited in the near and in the far field. However, the measurements of the electromagnetic field usually contain more noise. Consequently, the average of several power traces needs to be calculated in order to determine the Hamming weight of the value that is transferred over the bus.

An interesting difference between contact-based measurements and measurements of the

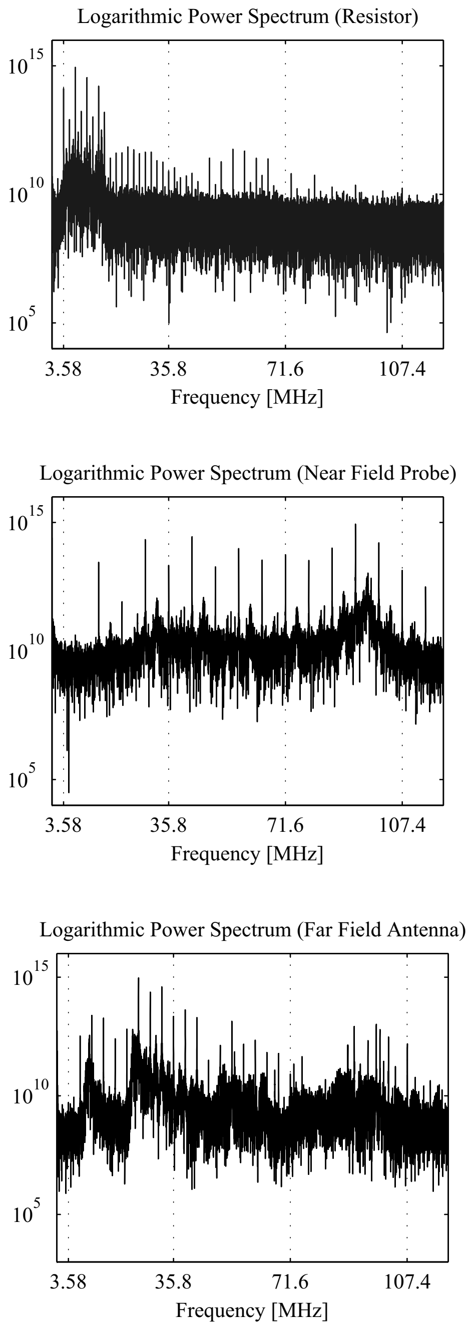

Figure 3.9: The power spectra of measurements conducted based on a resistor, a near field probe and an antenna.

electromagnetic field is presented in Figure 3.9. This figure shows the power spectra of measurements that have been conducted based on a resistor (see Figure 3.3), a near field probe (see Figure 3.3) and an antenna (see Figure 3.7). The spectra have been calculated based on fast Fourier transformations (FFTs) of the corresponding power traces. In all scenarios, the power dissipation of a smart card executing an AES-128 encryption has been recorded. The smart card was clocked with 3.58 MHz.

In all spectra shown in Figure 3.9, high peaks occur at the clock frequency and its harmonics. This can be explained as follows. Every time the clock signal rises from low to high, a lot of switching activities in the combinational circuits of the smart card are initiated. Hence, a high power dissipation occurs on every positive clock edge. This power dissipation signal is a broadband signal and therefore high peaks also occur at the harmonics of the clock signal.

The bandwidth of the probe or the antenna that is used in an attack determines which part of the spectrum is exploited. The Hamming weight is actually leaked by the side bands of most harmonics. In fact, this is why all three attack scenarios lead to similar results although the bandwidths of the measurement setups are completely different.

Measurements that are based on the insertion of a resistor into the ground wire of the power supply are limited to rather low frequencies (see the first plot of Figure 3.9). Most parts of the signal that is measured with the resistor are below 20 MHz. This is different for attacks conducted in the near field. In this scenario, harmonics of the clock frequency can be measured even above 100 MHz. This is also the case when an antenna in the far field is used.

It is important to point out that in all of our attack setups no explicit filtering of the input signal of the oscilloscope was performed. Filtering would enable attacks on carefully chosen parts of the spectrum and it would also enable attacks in higher frequency ranges. Actually, researchers at IBM have shown in [5] that harmonics in the range of several hundred MHz can be exploited, if a receiver is plugged between the antenna and the oscilloscope.

Based on our measurement setups, we have been able to exploit the side-channel leakage of cryptographic devices in the baseband. However, the leakage of the attacked smart card and the microcontroller has already been significant in this frequency band. Both devices leak the Hamming weight of the data that is transferred over the processor bus. This leakage can be exploited by contact-based power measurements and by measurements in the near and in the far field.

3.3.2 The EM Leakage of a Typical FPGA

The power dissipation characteristics of typical FPGAs have some notable differences to the power dissipation characteristics of typical microcontrollers. While many microcontrollers show that their power (or EM) leakage is due to the Hamming weights of the processed data, typical FPGAs show leakage characteristics that are more related to the switching activity within the circuit. Nevertheless, it was also observed that there are again similarities in the power dissipation and the EM leakage; the leakage is mainly due to the switching activity, however, EM measurements seem to be noisier than power measurements.

Figure 3.10 and 3.11 show this similar behavior. The two traces depicted in those figures show the leakage of an elliptic curve point multiplication that was implemented on the FPGA

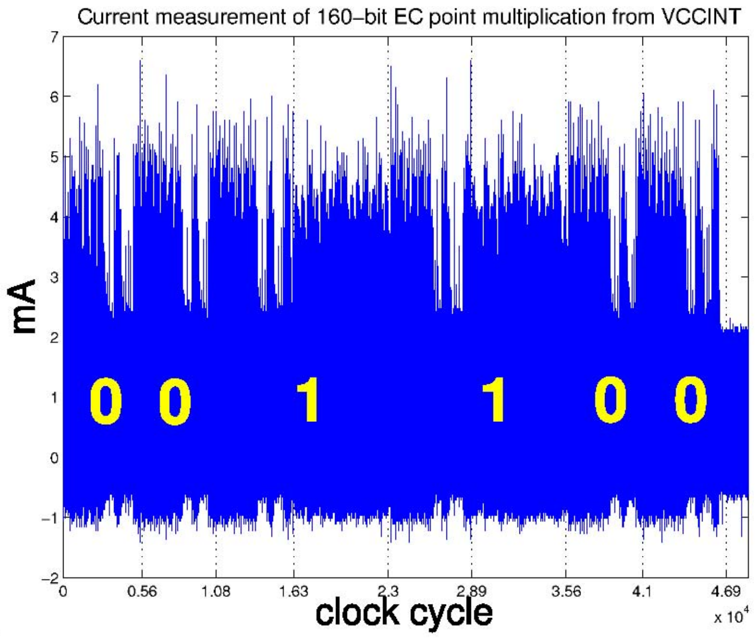

Figure 3.10: The power dissipation trace of an elliptic curve point multiplication measured on the FPGA setup

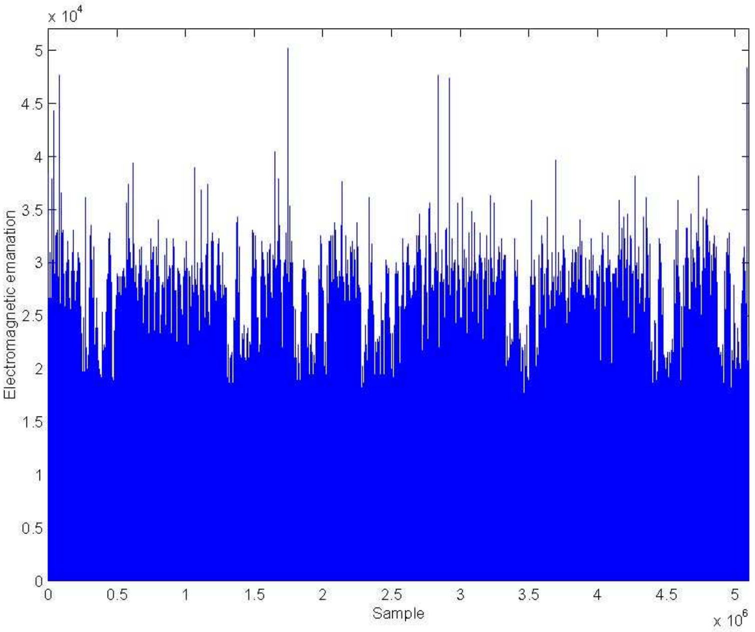

Figure 3.11: The EM trace of an elliptic curve point multiplication measured on the FPGA setup.

setup that we presented in Figure 3.5. The upper trace shows that power dissipation trace of one execution of the double-and-add algorithm. The lower trace shows the EM trace of a similar execution. In the lower trace, the key 100110 was used. It is clearly visible that in this FPGA setup, the same leakage is measured in the EM field as in the power dissipation.

3.4 EM Attacks

Two types of electromagnetic analysis attacks are distinguished. In a *simple electromagnetic analysis* (SEMA) attack, an attacker uses the side-channel information from one measurement directly to determine (parts of) the secret key. In a *differential electromagnetic analysis* (DEMA) attack, many measurements with varying input/output are used.

3.4.1 Simple EM Attacks

A simple electromagnetic analysis attack (SEMA) uses only one measurement and extracts information from the raw data or demodulated data. Simple analysis works because different operations consume different amounts of power. Also, it has been shown that the electromagnetic radiation of some instructions on smart cards directly leak information about operands in contrast to the power measurements, [5].

The following example is an attack on an elliptic curve multiplication over $GF(p)$. For more information see [71, 62, 19].

It can be derived from Fig. 3.11 that the key used during this measurement of an elliptic curve multiplication is 100110, because of conditional branching depending on the key in the algorithm. There is difference in radiation between the operations performed when the key-bit is 0 or 1.

In Fig. 3.11 raw data is used, but some techniques can be used to get a more clear image of the measurement. Fig. 3.12 shows an AM demodulated electromagnetic radiation trace of an operation in the used algorithm. This AM demodulation is done with a spectrum analyzer which could be placed in the measurement chain in front of the digitizing oscilloscope. Instead of using a spectrum analyzer to demodulate, a (radio) receiver or software could be used. A well known carrier is the clock frequency.

3.5 Differential EM Attacks

In a differential electromagnetic analysis attack (DEMA) a significant number of measurements and some statistical techniques are used to extract information. In DEMA, an attacker uses a hypothetical model of the attacked device which is used to predict several values for the electromagnetic radiation of a device. The quality of this model is dependent on the knowledge of the attacker. The predictions are compared to the real, measured electromagnetic radiation of the device. Comparisons are performed by applying statistical methods on the data. A differential analysis attack works because current dissipation is dependent on the value of operands in an execution, which leads to different electromagnetic fields.

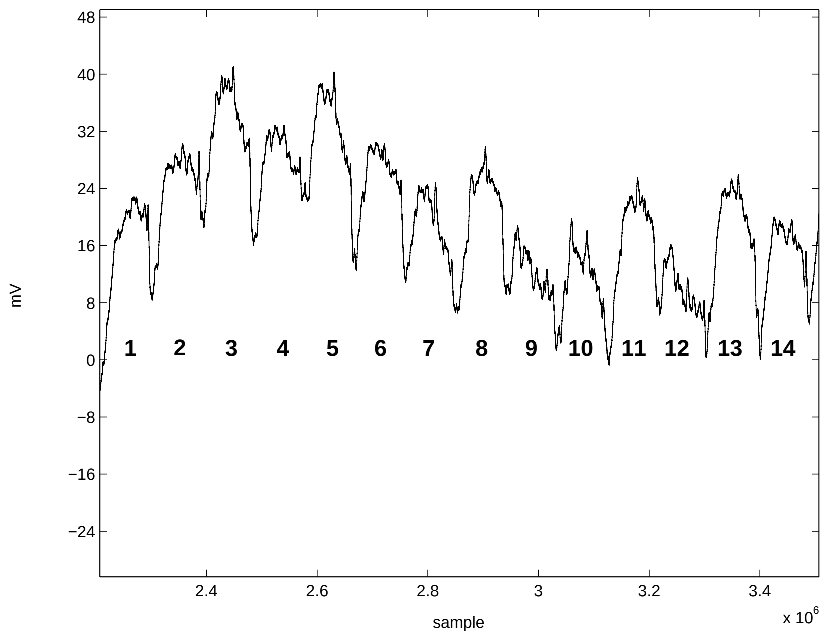

Figure 3.12: AM demodulated electromagnetic radiation trace of a 160-bit EC point addition

All differential analysis attacks start in the same way, first the attacker has to choose a point of attack, next the attacker collects measurements at the point of attack with different inputs but with the same key. After that the attacker guesses a key and uses the model to compute the predictions for the measurements; the last step involves using one of the techniques described underneath to find out if the key hypothesis was correct.

3.5.1 Correlation Analysis

In this analysis, the correlation can be measured with the Pearson correlation coefficient [31]. Let $t_i[j]$ denote the $i\,th$ measurement data at time j and T the set of measurements. Let p_i denote the prediction of the model for the ith measurement and P the set of such predictions. Then we calculate

$$C(T,P) = \frac{E(T \cdot P) - E(T) \cdot E(P)}{\sqrt{Var(T) \cdot Var(P)}} \qquad -1 \leq C(T,P) \leq 1 \,. \tag{3.1}$$

In Eq. (3.1), $E(T)$ denotes the expected (average) measurement data of the set of measurements T and $Var(T)$ denotes the variance of the set of measurements T. Note that T and P are said to be uncorrelated, if $C(T,P)$ equals zero; otherwise, they are said to be correlated. If their correlation is high, *i.e.*, if $C(T,P)$ is close to $+1$ or -1, it is assumed that the prediction of the model, and thus the key hypothesis, is correct.

3.5.2 Distance of Mean Test

A distance of mean test begins by running the cryptographic algorithm for N random values of input. For each of the N inputs, I_i, a discrete time side-channel signal, $t_i[j]$, is collected and the corresponding output, O_i, may also be collected. The side-channel signal $t_i[j]$ is a sampled version of the side-channel output of the device during the execution of the algorithm that is being attacked. The index i corresponds to the I_i that produces the signal and the index j corresponds to the time of the sample. The $t_i[j]$ are split into two sets using a partitioning function, $D(\cdot)$: $t_0 = \{t_i[j]\,|D(\cdot) = 0\}$, $t_1 = \{t_i[j]\,|D(\cdot) = 1\}$.

The next step is to compute the average side-channel signal for each set:

$$\begin{array}{ll} A_0[j] & = \frac{1}{|t_0|} \sum_{t_i[j] \in t_0} t_i[j] \\ A_1[j] & = \frac{1}{|t_1|} \sum_{t_i[j] \in t_1} t_i[j] \, , \end{array} .$$

By subtracting the two averages, a discrete time differential side-channel bias signal, $B[j]$, is obtained: $B[j] = A_0[j] - A_1[j]$.

Selecting an appropriate D function results in a differential side-channel bias signal that can be used to verify the guessed part of the secret key. A peak will appear at the exact moment of prediction.

3.5.3 Maximum Likelihood Test

If $t_i, i = 1, \dots, L$ indicates L independent sets of measured signals and if $H_k, k = 1, \dots, K$ represents K equally likely hypotheses on some property of these signals, then the maximum likelihood hypothesis test decides in favor of H_k if

$$k = \underset{1 \le k \le K}{argmax} \prod_{i=1}^{L} p(t_i | H_k) \, .$$

If we have two hypotheses, we will choose for hypothesis H_1 if

$$\prod_{i=1}^{L} p(t_i | H_1) \ge \prod_{i=1}^{L} p(t_i | H_0) \, .$$

Assume that t_i is a vector of length n and that for all hypotheses the signal has a multivariate Gaussian distribution; under these conditions and taking the natural logarithm of the previous formula, we get that we choose H_1 when

$$\sum_{i=1}^{L} ((t_i - \mu_{H_0})^T \Sigma_{H_0}^{-1} (t_i - \mu_{H_0}) - (t_i - \mu_{H_1})^T \Sigma_{H_1}^{-1} (t_i - \mu_{H_1})) \ge L(\ln |\Sigma_{H_1}| - \ln |\Sigma_{H_0}|)$$

When this theory is used in the traditional distance of mean test and by using a void hypothesis H_v, which is a random sorting into the 0-bin and the 1-bin, we will decide in favor of H_1 if $M_{H_1} \ge M_{H_0}$ with

$$M_{H_i} = \frac{(\mu_{H_i} - E[\mu_{H_v}])^2}{V[\mu_{H_v}]} - \frac{(\mu_{H_i} - E[\mu_{H_i}])^2}{V[\mu_{H_i}]} - \ln(\frac{V[\mu_{H_i}]}{V[\mu_{H_v}]}) \qquad (3.2)$$

at the correct point in time. The following maximum likelihood estimators are used for the expected values of the mean and the variance: $E[\mu_H] = \mu_H$ and $V[\mu_H] = \frac{\sigma^2_{H,0}}{N_0} + \frac{\sigma^2_{H,1}}{N_1}$ with $\mu_H = \mu_0 - \mu_1$, the difference of the mean of the 0-bin and the 1-bin, $\sigma^2_{H_i}$ the variance of the i-bin and N_i the number of elements in the i-bin.

Fig. 3.13 shows the result of a distance of mean test on an ECC algorithm and Fig. 3.14 shows the result of a correlation analysis attack on the same algorithm.

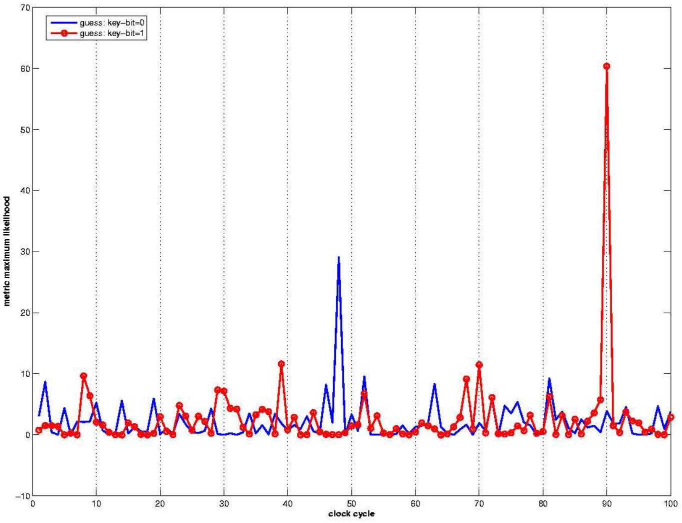

Figure 3.13: Result of a distance of mean test, the peak shows the exact moment of prediction and confirms the correct key hypothesis.

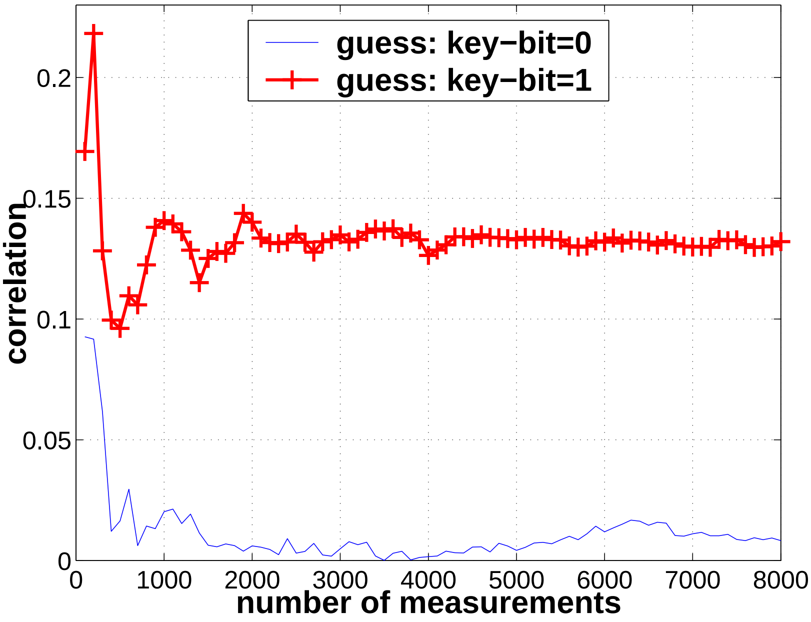

Figure 3.14: Result of a correlation analysis test, the upper curve shows the right key guess, the lower a false key guess

3.6 Post Processing Techniques

3.6.1 Discrete Fourier Transform

The discrete Fourier transform (DFT) can be used in order to find the clock frequency or to find any frequency information in the measured trace. Let x be a complex series with N

samples of the form $x = x_0, x_1, \ldots, x_{N-1}$ where x_i is a complex number. The series outside the range 0, $N-1$ is extended N-periodic, that is, $x_i = x_{i+N}$ for all i.

The discrete Fourier transform (DFT) of x is denoted as X; it also has N samples. The forward transform is defined as

$$X_n = \frac{1}{N}\sum_{i=0}^{N-1} x_i e^{-jk2\pi n/N} \; for\, n = 0 \cdots N-1. \tag{3.3}$$

3.6.2 Demodulation Techniques

There are different ways to superpose an analog signal on a carrier. More specific, a modulated signal (bandpass signal) can be written as $s(t) = Re(g(t)e^{j\omega_c t})$ where $\omega_c = 2\pi f_c$ and f_c is the carrier frequency. To obtain the desired modulated signal $s(t)$ the appropriate modulation mapping function $g[m(t)]$ has to be applied with $m(t)$ the analog signal. For amplitude modulation $g(t) = A_c[1 + m(t)]$. For angle modulation $g(t) = A_c e^{j\theta(t)}$. Phase modulation (PM) and frequency modulation (FM) are special cases of angle-modulated signaling. For PM, $\theta(t) = D_p m(t)$ with D_p the proportionality constant, in words it means that the phase is directly proportional to the modulating signal. For FM, the phase is proportional to the integral of $m(t)$, so that $\theta(t) = \int_{-\infty}^{t} m(\sigma)d\sigma$ where D_f is the frequency deviation constant D_f.

3.7 Countermeasures

As EMA attacks are still under intensive development the countermeasures are an active research area. It is obvious that Power Analysis (PA) countermeasures can also be applied to EMA. However, because the EM signal contains more information than the power signal, other techniques have to be applied to defeat EMA attacks. With respect to the goal they try to achieve, countermeasures can be divided into two types. The first one tries to lower the signal intensity while the second one attempts to reduce the information contained in the signal. The first type is mainly already included in the design strategy of most silicon vendors. An example of the first type is shielding. Examples of the second type are asynchronism, dual-line logic, etc. [78, 38]. Sometimes countermeasures are divided into software and hardware countermeasures.

The main software countermeasures against PA attacks are as follows. *Time randomization* is given in [47, 56, 66, 69, 75, 46, 52, 13]. In this type of countermeasure, operations occur during random intervals of an execution. *Permuting the execution* is given in [45]. *Masking techniques* are shown in [7, 44, 42, 92, 91].

The main hardware countermeasures against PA attacks are as follows. *Increasing the measurement noise* is done by using a random number generator [33]. *Power signal filtering* was proposed to be added to the implementation [84, 32]. *Novel circuit designs* are given in Sect. 3.7.1. *Detachable power supplies* was given in [84]. *Securing algorithm at the logic level* was proposed in [89]. This method employs logic gates with a power dissipation, which is independent of the data signals and therefore the technique removes the foundation for

DPA. Asynchronous circuits can be used too [37, 72]. Usage of reversible logic in order to reverse the computation which returns the consumed energy during the computation back to the circuit was considered in [41]. It has to be stressed that most of them are not tested against EMA yet.

Here we look into that problem in the following sections on all levels of abstraction, which are depicted in the security pyramid in Figure 3.15. Each abstraction layer represents specific modeling, design and implementation issues that must be covered for secure system operation [81]. Many of the proposed countermeasures are also mentioned in [78].

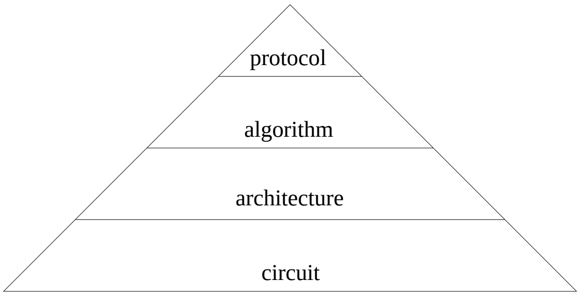

Figure 3.15: Security pyramid showing the different levels of abstraction.

3.7.1 Circuit Level Countermeasures

The power dissipation of a circuit depends on the output transitions of the gates. If these transitions depend on secret information, the security of the implementation can be compromised. Hence, to make a circuit resistant against attacks that use the power supply as direct or indirect information, the power dissipation should be independent of the secret information. The circuit-level approaches to achieve this are the following: design for low dissipation and new logic styles. The latter can be divided into two categories: custom logic styles and standard logic styles. Custom logic styles are only applicable to custom ASIC implementations. Standard logic styles combine standard cells from existing libraries into new standard cells. Hence, they can be used for FPGA implementations as well as standard cell ASIC implementations. Tiri *et al.* developed SABL [58], which is a custom logic style, and WDDL [90], which is a logic style consisting of standard cells. Besides these there are other DDLs, DyCML, ...

When a design is made for low dissipation, the measurements will be harder to make and more stress will be put on the amount of measurements or the measurement equipment. Also, the shrinking trend in the silicon industry causes a natural reduction of the emanated field.

3.7.2 Countermeasures at The Architecture Level

Quisquater and Samyde discuss noise generators and the metal layers used in chips in order to reduce the radiated field [78]. Mostly used materials for this purpose are aluminium or copper. They also mentioned the Faraday cage. Such a construction would possibly block compromising electromagnetic radiation but it would be very difficult to apply this idea on smart card platforms. Asynchronous processors would affect an attacker's task in aligning DEMA traces. Namely, synchronism in processors makes differential attacks possible because it makes the statistical combination of several curves effective. However, this approach is not so easy to handle and affects design strategies tremendously. A Faraday cage around the device would stop the signals to leak to the outside. Although effective, this countermeasure is not feasible in practice for every case. Another proposed countermeasure is a modification of the chip [78]. In [30], the authors use a new type of architecture that makes it harder for an attacker to mount an EMA. The basic idea is to design an architecture with independent cells that randomly proceed.

3.7.3 Algorithm/Protocol Level Countermeasures

Protection at this level is considered to be cheaper as these are also mainly software countermeasures. One of the most straightforward methods to help resist electromagnetic (or power) analysis attacks is to use randomization in the time domain. If the operations are randomly shifted in time, then statistical analysis of the power dissipation signals can be more difficult. Many researchers have noted the benefits of timing randomization as a countermeasure [70, 26, 34], but they have also cautioned that attackers might be able to remove such randomization. Other important software countermeasures include all types of masking techniques [45, 44, 42, 92].

Another suggestion mentioned by various authors is splitting the sensitive data into two balanced bytes to keep the Hamming weight of all processed data constant [70]. Thus, if Hamming weight information is the cause of leakage, then an attacker will not learn anything useful since all bytes have exactly the same weight. Unfortunately, data balancing is not very effective against all types of DEMA attacks (*i.e.*, DEMA attacks against address bus data) and is also not effective if the processor leaks Hamming distance information.

Chapter 4

Fault Attacks

Fault Analysis is an active attack against the implementation of security modules. In the context of cryptanalysis, fault analysis aims to disturb the computation of a cryptographic algorithm in such a way that an erroneous result is obtained. By applying mathematical cryptanalysis these erroneous results can be used to extract cryptographic key material.

We summarize the relevant results of fault analysis that have been achieved, yet. Our main approach is to put forward a state-of-the-art survey of fault analysis that considers physical techniques for fault induction, models used for attack scenarios, methods for cryptanalytic postprocessing, and state-of-the-art countermeasures. This chapter is organized as follows. In Section 4.1 we reconsider physical effects that can be applied to induce a fault. Section 4.2 discusses models regarding their underlying assumptions of the implementation of cryptographic devices and their security services, the fault characteristics and the adversarial capabilities. Section 4.3 presents cryptanalytic methods that have been proposed for block ciphers, stream ciphers as well as asymmetric primitives. Our survey is finished by evaluating countermeasures in Section 4.4.

4.1 Techniques for Fault Insertion

4.1.1 Definitions

We start with a definition of a *fault* and quote from Wikipedia [1]:

> **Definition:** A fault is defined as an abnormal condition or defect at the component, equipment, or sub-system level which may lead to a failure. According to Federal Standard 1037C, the term fault has the following meanings:
>
> 1. An accidental condition that causes a functional unit to fail to perform its required function.
> 2. A defect that causes a reproducible or catastrophic malfunction. A malfunction is considered reproducible if it occurs consistently under the same circumstances.

3. In power systems, an unintentional short-circuit, or partial short-circuit, between energized conductors or between an energized conductor and ground. A distinction can be made between symmetric and asymmetric faults.

Faults occur accidentally, e.g., due to wear lifespan of components. The sensitivity towards faults and its frequency can be enhanced by external means. We define such external means as *fault insertion techniques*, or, alternatively as *fault induction techniques.*

Fault induction aims to cause an interference within the physical implementation and to enforce an erroneous behavior of the implementation.

We consider environmental based attacks and particle based attacks. Environmental attacks include tampering at the external interface, but also rapidly changing electromagnetic (EM) fields or even overheating. Particles used in fault induction are photons emitted by intensive light sources, but also charged or neutral particle beams with non-zero mass. A recent survey on current technologies can be found in [87].

4.1.2 Wear Lifespan

Faulty behavior of semiconductor components can occur because of aging defects in a normal operation environment. Effects that cause aging defects are hot carriers, electromigration, and radiation [73]. Hot carriers are high energetic electrons that cross the potential barrier of the gate oxide due to tunnel effect. These charges can lead to changes in the threshold voltage of gates. Electromigration is caused by high current densities yielding to deformations of the circuit lines. Such effects can have an impact on the wiring resistance and the current leakage. Radiation induced aging is caused again by the insertion of charged particles into the circuit.

4.1.3 Non-invasive Fault Insertion

The importance of protection against environmental fault insertion techniques is already included in the FIPS PUB 140-2, Security Requirements for Cryptographic Modules ([2]): "Deliberate or accidental excursions outside the specified normal operating ranges of voltage and temperature can cause erratic operation or failure of the electronic devices or circuitry that can compromise the security of the cryptographic module."

In this category of non-invasive fault insertion we consider changes of the environmental conditions (e.g., by temperature) and manipulations at the external interfaces of the cryptographic device (e.g., by glitches).

Temperature

Electronic components are specified for a certain range of temperature to guarantee a reliable operation. Faults are more likely to occur outside of this specified range. The effect of 'freezing volatile data memory' (e.g. [95, 85]) can be already observed around temperatures of $-20°C$. [85] reports that the data remanation varies widely, also between devices from the same same type. The main focus of these contributions is based on reading out data memory, even after an activation of a zeroization circuitry. Nevertheless, it can be assumed that freezing may also lead to errors during computations.

Overheating of the chip is an alternative attempt which leads to unforeseen effects, e.g., by the CPU. [12] mentions that random modification of RAM cells can also be caused by heatings. Furthermore it is reported in [12] that the thresholds of read and write temperature do not coincide. A possible exploit of this fact would be to operate the cryptographic device between these two thresholds, e.g., within a temperature range whereat a write operation does not work whereas read operations are not affected.

It is important to note that temperature effects are long-lasting in comparison with the duration of a clock cycle.

Other Physical Effects

The first rumors for possible fault induction techniques after the announcement of the Bellcore attack [22] dealt with microwaves. To the knowledge of the authors a successful approach has never been reported. The exploit of other physical effects, as, e.g., ultrasound has also never been mentioned.

Glitches

Glitches are short-time pulses that are applied at the external interfaces of the cryptographic device during operation. These abnormal conditions can cause faults in the computation as it was reported by [9]. Glitches can be injected, e.g., by glitching the external clock supply. Accordingly to [95]: "By lengthening or shortening the clock pulses to a clocked circuit such as a microprocessor, its operation can be subverted. Instructions or tests can be skipped or generally erratic operation can be induced [9]."

Similarly, by glitching voltage pins to abnormally high or low values for a particular length of time, faults can be induced in the circuit. "The erratic behavior may include the processor misinterpreting instructions, erase or over-write circuitry failing, or memory retaining its data when not desired"[95].

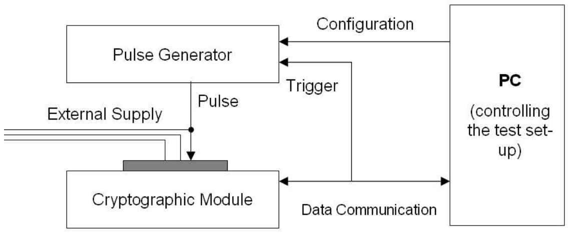

Figure 4.1: Test set-up for fault induction using glitches.

The test set-up (see Figure 4.1) for glitch injection consists of the cryptographic device, a pulse generator and a standard PC (or workstation) for the overall control. The PC initiates the request to the cryptographic module and receives the response of the device. Further, the PC controls the pulse generator. Note, that the cryptographic operation can be monitored *during* fault induction, e.g., by analyzing side-channel information, but this side channel information can not trigger the actual fault induction itself. This means, the pulse generator has to be triggered at a certain delay time with reference to the request that is sent to the cryptographic device. The pulse parameters delay time, amplitude, and duration are varied during testing.

4.1.4 Semi-invasive Fault Insertion

Semi-invasive fault induction requires some preparation of the chip under test. Usually, the packaging of the chip has to be removed first which is, e.g., described in [8].

Optical Fault Induction

In [86] optical fault induction was introduced. Optical fault induction makes use of a photoflash (white light), or alternatively, a laser source. In [86] the authors mounted the photoflash on the video port of a manual probing station and illuminated SRAM memory. The SRAM was shielded with an aperture made from aluminium foil. This aperture allows to expose only one memory cell with the light yielding a precise area resolution. [86] demonstrated that it is possible to change the state of one SRAM cell.

In the visible light spectrum, the basic interaction mechanism of photons with atoms in semiconductor devices is the photoeffect. The photoeffect leads to the absorption of a photon by an atom which in turn emits an electron. Basically, the photoeffect generates free charges at the conductance band. These free charges may form a current that cause errors at the gate level. If electron-hole pairs are generated close to a N-channel or P-channel the gates are especially sensitive [40].

Nowadays, optical fault induction is the most common way to test the sensitivity of smart cards towards faults [40]. Possible parameters are the luminosity of the light, its wavelength (energy), the duration of the emission, the location and the extent of the impact zone. For a more detailed discussion we refer to [40].

Optical fault induction is semi-invasive, as the package of the chip has to be removed. Photographies of test set-ups for optical fault induction can be found, e.g., in [12]. Their laser based set-up uses a microscope to focus the beam.

Electromagnetic Induction

Another semi-invasive attack was presented by [79]. Herein, a miniature coil was produced in which current is injected. The change of the magnetic field induced causes eddy currents in the chip plane. These induced currents overlay on the current data signals which in return provokes faults. For better results the package of the chip should be removed.

In comparison to the optical fault injection, the area resolution achievable by electromagnetic induction is less precise ([80]). This was also confirmed by [40]: the authors see another difficulty in practice, namely that the generation of a high magnetic field requires a high current, which would destroy the wire.

X-ray

Semi-invasive fault induction using X-rays would be another alternative. To the knowledge of the author, there are no publications until now. A possible drawback might be the availability and handling of the sources. Furthermore, the interaction with matter is different, as the Compton effect becomes more important. In the Compton effect photons are elastically scattered with electrons. Energy is transferred to the electrons and leads to a reduction of the photon's frequency.

Moreover, in [95] it is reported, that X-ray radiation can imprint CMOS RAM.

Infrared Laser

In [95] it is reported that it is possible to read and write storage cells by using an infrared (IR) laser directed through the bulk silicon side of the chip. This is feasible as silicon is transparent at IR frequencies. By special shielding techniques the area that is affected by the IR beam might be small. Recent results are presented in [87], but this work aims at imaging and not at fault induction.

4.1.5 Invasive Fault Insertion

Invasive fault induction requires the depackaging of the chip and additionally the removal of the passivation layer. In most scenarios, the circuit is directly manipulated using microprobing. Using expensive equipment such as focused ion beam workstations even manipulations at deeper metal layers are also feasible.

Invasive fault induction techniques are only rarely considered in the scientific literature. One public reference is [87].

Ions, electrons and neutrons

To the knowledge of the authors, there has been no contribution dealing with ion, electron and neutron radiation in the context of fault analysis. Nevertheless, in [95] it was reported that "energy probes can read or write the contents of semiconductor storage, or change control signals". Especially, the "electron beam of a conventional scanning electron microscope can be used to read, and possible write, individual bits in an EPROM, EEPROM, or RAM" [95]. Impact of heavy energetic particles as cosmic rays is known to cause single event effects ([65, 67]). Such effects have been already studied at the development of semiconductor devices for space and high energy physics.

Active Probes

In [95] it is reported that active probes "can inject signals or information into an active system" assuming that the active probe contacts the internal circuit directly.

Modification

Invasive attacks are mounted internally, within the semiconductor device. Tools for these kind of modifications are, e.g., focused ion beams. By using such tools, complex modifications are feasible which can completely modify the internal construction. In our interpretation such modifications are not considered to be part of fault induction techniques, though they are probably the most efficient tools for microelectronic failure analysis.

4.2 Models

In the last years a variety of fault attack scenarios (see Section 4.3) have been published. In this Section we aim to recapitulate and put forward models that can be used as underlying assumptions for concrete fault attacks. Concretely, we deal with models of the cryptographic device, the nature of faults, and an adversarial model for fault analysis.

4.2.1 Model of Cryptographic Devices

Cryptographic devices may be made up of hardware (circuitry) only as, e.g., ASICs and RFID tags. Moreover, other devices include additionally software components as, e.g., microcontroller based smartcards, but also encapsulated security modules that consist of multiple components. In the latter case a reliable hardware-software co-design is also of interest. In our discussions we aim to include both hardware and software aspects for fault induction. We assume that the implementation is deterministic to a certain extent.[1]

We first discuss our assumptions. We distinguish low-cost devices used in high volume markets and more expensive encapsulated cryptographic devices that contain multiple components and are deployed for special purposes.

Low-Cost Devices

Typically, low-cost devices such as smartcards and RFID tags do not contain an internal battery and are supplied with energy and clock by a reader device. This has an implication towards tamper responsiveness: as these modules are not permanently powered up, an internal detection of active attacks is only possible if the device is field supplied. We assume that the reader device does not store cryptographic keys, so that the counterpart of cryptographic protocols is a back-end server or a secured terminal which is not available to the adversary.

[1] The implementation may include internal timing jitter as caused by random process interrupts or asynchronous clocking.

The adversary owns a reader device to communicate with the low cost cryptographic device. The external interfaces of the device include the channels used for communications and external supply of voltage and clocking. They are under complete control of the adversary. Computations are only feasible if the device is powered on and the computation can be always interrupted by removing the power supply.

Encapsulated Cryptographic Devices

Multi-chip cryptographic devices that are built accordingly to the requirements of FIPS-140 ([2]) have to fulfill the given requirements on the physical security for the aimed security level. At level 4 (which is the highest level achievable) FIPS-140-2 requires a tamper detection envelope with tamper response and zeroization circuitry. Furthermore, environmental failure protection for temperature and voltage are required. Encapsulated cryptographic devices are equipped with an internal power supply, but can also be supplied externally. The tamper response and zeroization circuitry shall remain operational when cryptographic keys are stored in the device.

Physical Boundaries

According to FIPS 140-2 [2] we introduce the concept of the *cryptographic boundary* that encloses all security relevant and security enforcing parts of an implementation. Additionally, we define a second boundary that we call the *interaction boundary* that is specific for each physical interaction process [64]. If the adversary does not pass the interaction boundary, the physical interaction is not effective at the cryptographic device. The interaction boundary can be an outer boundary of the cryptographic boundary, as, e.g., in case of temperature which affects the entire cryptographic module. Interaction with light is only feasible if a non-transparent encapsulation is partially removed, e.g., the chip is depackaged. Because of the limited range of the interaction, interaction processes using particles with non-zero mass may require the removal of other layers which breaches the cryptographic boundary.

4.2.2 Fault Models

First, [22] introduces three types of faults, as there are:

1. Transient faults,
2. Latent faults, and
3. Induced faults.

Only in the latter case, the adversary is assumed to have physical access to the cryptographic module and apply physical fault injection. [22] described transient faults by giving the example of a certification authority service that “might generate faulty certificates on rare occasions”. Latent faults are bugs in the hardware or software, that are difficult to catch as they occur rarely.

In this report our focus is on fault induction, as this is the state-of-the-art testing. Based on the adversary's capabilities the fault origin might be either local or global. For the *random fault induction* the concrete fault caused is of minor interest for the adversary. In precise fault attacks, it is often extremely important to achieve a concrete fault at both a specific instant and a specific location. This precision of faults is included in the adversarial model in Section 4.2.4.

In this Section we consider the different nature of faults that can be induced.

Faults can either be transient ("soft errors") or permanent ("hard errors"). [12] gives a taxonomy for provisional (transient) faults and destructive (permanent) faults. For fault injection, provisional faults are the method of choice as the implementation remains fully functional under test. Destructive faults affect the chip's implementation permanently. We reproduce the taxonomy from [12] below in a shortened form.

Provisional Faults [12]

- Single Event Upsets (SEUs) are flips in a cell's logical state to a complementary state.
- Multiple Event Upsets (MEUs) are the generalization of SEUs occurring simultaneously.
- Dose Rate Faults are due to several particles whose cumulative effect generates a sufficient disturbance for a fault to appear.

Destructive Faults [12]

- Single Event Burnout faults (SEBs) are due a parasitic thyristor being formed in the MOS power transistors.
- Single Event Snap Back faults (SESs) are due to the self-sustained current by the parasitic bipolar transistor in MOS transistor channel N.
- Single Event Latch-up faults (SELs) are propagated in an electronic circuit by the creation of a self-sustained current with the releasing of PNPN parasitic bipolar transistors in CMOS technology.
- Total Dose Rate faults are due to a progressive degradation of the electronic circuit subsequent to exposure to an environment that can cause defects in the circuit.

Moreover, faults can modify either memory contents of the implementation or the implementation itself. In case of software, a fault modifies the implementation by modifying the executable program code (destructive fault) or the execution of the program code (provisional fault). In case of hardware, either the internal signals are misinterpreted (provisional fault) or, e.g., short-circuits (destructive fault) are caused. Regarding the modification of memory contents there might be a preferred direction of the error, i.e, if the probability to cause a transition from 0 to 1 is significantly different from a transition from 1 to 0. If this is the case we call it a *directional* data fault that is caused by *asymmetric memory* properties.

4.2.3 The Objectives of an Adversary

The security service of the cryptographic device may include a cryptographic algorithm, but also security enforcing services are feasible that work without any cryptographic means. An example for a non-cryptographic mechanism is a human authentication based on a Personal Identification Number (PIN) using an authentication failure counter.

Informally speaking, an adversary is *successful*, if the insertion of faults either i) yields access to a security service without knowledge of the required secret or ii) yields partial information about the secret.

Interfering the Cryptographic Computation

A cryptographic security service aims to provide security objectives as authentication, non-repudiation, integrity or confidentiality by cryptographic means. It is the goal of the adversary to interfere with computations which involve the secret cryptographic key material. For this, it is required that the implementation is not aware of any errors caused and returns a faulty output. Finally, mathematical cryptanalysis is applied to analyze the erroneous output obtained. The faults needed for a specific attack may be caused within a broad range of intermediate results [22] or have to be very precise in time and location (e.g., [77]). Further details on concrete scenarios are outside of the scope of this Section and we refer to Section 4.3.

Bypassing the Security Service

A bypass of a security mechanism is applicable to hardware and software parts, but probably the most obvious bypasses deal with changes of the program execution in software. For instance, a modification of a security state may give more privileged access rights.

Deactivation of the Security Service

Whereas bypassing is applicable to both hardware and software components, deactivation of components especially concerns hardware components. A possible goal would be a deactivation of the random number generator used.

4.2.4 Adversarial Models

Herein, adversarial models are discussed basically from a physical perspective. We denote the adversary by $\mathcal{A}$. By assumption $\mathcal{A}$ has physical access to the physical device $\mathcal{D}$ under attack and can run a high number of instances of a security service $\mathcal{S}$. Each instance is initiated by a request of $\mathcal{A}$ and D finishes after some computational time T returning a response. Moreover, $\mathcal{A}$ acts in an active, adaptive way. $\mathcal{A}$ aims to disturb the intended computation of S by physical means. The means of $\mathcal{A}$ can be manifold based on the attacking time, the technical equipment as well as the grade of knowledge about the implementation of $\mathcal{D}$.

Generally, the main limitations are caused by the technical equipment available. The attacking time might not be that important and the knowledge about the implementation can be improved by analysis. Because of this we distinguish the non-invasive adversary $\mathcal{A}_{non-inv}$, the semi-invasive adversary $\mathcal{A}_{semi-inv}$, and the invasive adversary $\mathcal{A}_{inv}$. Each adversary is able to monitor the effects caused by fault induction using auxiliary means. If necessary, $\mathcal{A}$ applies cryptanalytical methods for a final analysis step.

Moreover, we assume that $\mathcal{A}$ is able to perform multiple fault injections during the computation time T that are bounded by M, wherein M is a small number. These fault injections occur at the times $\{T_1, T_2, ..., T_M\}$ with $0 \leq T_1 \leq T_2 \leq ... \leq T_M \leq T$. Let L be a small number of spatial separated set-ups for fault injection that can be operated in parallel. The distinct fault injections during one invocation of S are numbered as $\mathcal{F}_{l,m}$ with $l \in \{1, ..., L\}$ and $m \in \{1, ..., M\}$.

The information leakage is considered to be partial for each fault insertion, unless a mathematical analysis is available that leaks to a total break once a certain faulty response is received. ([22] gives an example that one faulty result is sufficient in case of a non-secured RSA-CRT implementation.)

Fault induction is a probabilistic process with success rate p. Complementary events with probability $1-p$ are attempts that do not lead to any fault and attempts causing an unintended fault. The success rate p depends on the interaction rate of the physical interaction process with semiconductor materials of the cryptographic device.

If precision is needed, the fault must be injected into the cryptographic device with sufficient resolution in space and time. According to [64], we consider a local fault induction at a certain target with area extension dA and at the depth z with depth extension dz within an homogeneous medium. Let ΔA be the area and Δz be the depth that is affected by a uniform fault induction process. An area related probability p_{Area} =min$\{1, \frac{dA}{\Delta A}\}$ and a depth related probability $p_{Depth}(z) = \frac{\int_z^{z+dz} \eta(z')dz'}{\int_0^{\Delta z} \eta(z')dz'}$ are then defined, wherein $0 \leq \eta(z) \leq 1$ is the transmission of the medium for the interactive particles (e.g., photons) that cause the fault induction in dependency on the penetration depth z.

If precision in time dt is needed given a timing resolution of ΔT for the fault induction process, the time related probability is p_{Time} =min$\{1, \frac{dt}{\Delta T}\}$, else $p_{Time} = 1$. In case of a precise fault $\mathcal{F}_{l,m}$ in space and time p is reduced by a factor of $p_{Area}\ p_{Depth}\ p_{Time}$.

The non-invasive adversary $\mathcal{A}_{non-inv}$

The non-invasive adversary $\mathcal{A}_{non-inv}$ attacks the cryptographic device by using its external interfaces or by changing the environmental conditions. $\mathcal{A}_{non-inv}$ does not breach the cryptographic boundary of the device. Faults that are injected are random and they are not precise, i.e. the target area ΔA and the target depth Δz are given by the dimensions of the physical device. Tools for fault injection consist of standard laboratory equipment and are in a low-budget range. Changes in the environmental condition as overheating are long-lasting yielding a high value for ΔT which in turn gives a very small value for p_{Time}. ΔT for glitches in the external lines can be of high precision so that glitches can yield high values for p_{Time}, but the product of p_{Area} and p_{Depth} is nearly negligible resulting in a very low probability to induce specific errors.

The semi-invasive adversary $\mathcal{A}_{semi-inv}$

The semi-invasive adversary $\mathcal{A}_{semi-inv}$ uses light or electromagnetic radiation as interaction process. The cryptographic boundary remains intact. Note that photons, i.e., EM radiation, emitted by $\mathcal{A}_{semi-inv}$ are allowed to pass the cryptographic boundary. $\mathcal{A}_{semi-inv}$ applies optical fault induction [86] or electromagnetic induction [79]. $\mathcal{A}_{semi-inv}$ is able to perform a local fault injection without directly connecting the device. The achievable local area resolution, i.e. p_{Area}, can be high. For optical fault induction, $\eta(z)$ is reduced exponentially with increasing target depth due to the photo effect. Accordingly p_{Depth} decreases exponentially. Failures in deeper layers are hard to achieve for $\mathcal{A}_{semi-inv}$. $\mathcal{A}_{semi-inv}$ can achieve high values for p_{Time}.

The invasive adversary $\mathcal{A}_{inv}$

The invasive adversary $\mathcal{A}_{inv}$ penetrates the cryptographic boundary. Matter can be inserted or removed from the cryptographic boundary. It is typically required that the passivation is removed at invasive attacks. Moreover, $\mathcal{A}_{inv}$ is allowed to probe within the overall internal construction. By doing so, $\mathcal{A}_{inv}$ is bounded by L different locations that can be mounted in parallel. Fault injections are caused by particles with non-zero mass (as ions) or directly at the probes. $\mathcal{A}_{inv}$ acts adaptively within the cryptographic implementation. Therefore, $\mathcal{A}_{inv}$ is able to target and possibly deactivate the most critical parts of the implementation. $\mathcal{A}_{inv}$ is able to gain privileged insights by physical reverse engineering. Of particular interest are fault injections at interconnections as well as at memory cells. The probabilities p_{Time}, p_{Area} and p_{Depth} are high resulting in a high overall probability to induce specific faults. Note, that also for $\mathcal{A}_{inv}$ p_{Depth} decreases with increasing depth z though the dependency is more complex.

The boundaries between an invasive and semi-invasive adversary can be a matter of interpretations, e.g., if the passivation is locally removed by microprobing needles [8]. Existing high-end chip defenses include a top-layer metal shielding [86, 63] which in turn yields a clear distinguishing between an invasive and semi-invasive adversary.

4.2.5 Defense Models

As defenses have to be part of the implementation under attack they are subject to fault induction, too. The question arises whether an implementation can ever be provable secure against multiple precise fault injections, especially by $\mathcal{A}_{inv}$ which has to be negated. An increase in the number of defenses requires a corresponding increase in the number of fault injections, but with decreasing overall success probability. By assumption it is feasible for $\mathcal{A}_{inv}$ to monitor L locations in parallel, where at each location M successive fault injections can be done at different points in time. From the theoretical point of view the redundancy of an implementation shall therefore exceed L in space to counteract an $\mathcal{A}_{inv}$. It was shown in [51] that the transformation of a n-gate circuit into a circuit of size $O(nL^2)$ is perfectly secure against all probing attacks leaking up to L bits at a time. Such a redundancy of the circuit might be feasible at specialized developments but surely not for high volume products. In summary, perfect protection against multiple fault injections of an invasive attacker does not hold for high volume products leaving a remaining non-zero success rate. Because of their bounds in precision for time and space the reduction of p for $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$ is more rigorous. The decision whether or not the device shall enter a permanent non-responsive mode in case of alarms depends on the concrete impact probability as well as the concrete security service and is a matter of risk analysis.

Note, that the probability p can be reduced by extending the randomness of the timing in the implementation of S, as $p \sim p_{Time}$. An alternative approach is to reduce $\eta(z)$ at the locations of security relevant and enforcing parts of the device, e.g. by shielding with metal layers.

Due to the shrinking process semiconductor devices become more and more compact. Shrinking decreases the resolution for the adversary, but it also enhances the sensitivity of the circuit towards fault inductions. Due to the underlying physics, there will be some boundaries in the future as a reliable testing of new products has to be still feasible. Devices used for product development are always possible tools for fault induction.

We discuss three aspects that are related to the hardware of cryptographic devices in more detail in Section 4.4.

4.3 Post-processing

4.3.1 Block Ciphers.

Mounting a fault attack against a block cipher is relatively easy; the challenge is to find one which uses as few faulty ciphertexts as possible, and without too strict constraints on the timing of the fault. The first fault attack against a block cipher is due to E. Biham and A. Shamir in 1997 [18]; their paper deals mainly with DES.

In this section, we first present a very generic attack from [18], targeting memory registers in which a key is stored; however this attack is only possible under a very restrictive fault model. We then briefly describe Biham-Shamir's attack on DES. Then we present a fault attack which works against any substitution-permutation network cipher [77], with the AES as a particular case. Finally, we briefly discuss the problem of exploiting faults occurring during the first few rounds of a block cipher [48].

A Generic Attack.

Assume an attacker is able to apply some kind of stimuli to a to a cryptographic device such as a smart card, whose effect is that each bit at 1 in a given register could flip to 0 with probability p_1, but a $0 \rightarrow 1$ transition is not possible. If p_1 is small enough, the probability that two bits flip simultaneously is assumed negligible. Consider the register contains an unknown key K of length n, and suppose the attacker is able to ask the smart card to encrypt under this key. The following attack applies:

1. In a first step, the attacker asks for the encryption of some plaintext P, then applies a "stick-at-0" stimuli, asks for the encryption of P again, etc. This way she obtains a sequence of ciphertexts $C_0, C_1, ..., C_r$ (if the same ciphertext has been obtained several times due to the stimuli having no effect, we keep only one copy of it). C_0 is the ciphertext corresponding to key K, and if the process described above has been conducted a sufficient number of times, C_r corresponds to the encryption of P under the all-0 key, and r is the Hamming weight of the initial key.

2. Let K_i be the key used to obtain C_i ($K_0 = K$, $K_r = 0$). Then K_i and K_{i+1} differ only by one bit. Knowing K_{i+1}, it is possible to retrieve K_i by doing trial encryptions under keys $K_{i+1} \vee \delta_j (j = 1, ..., n)$, where δ_j has all its bits equal to 0 except the j^{th}. As K_r is known, K_0 can be retrieved by iterating this process.

The overall complexity of the attack is $\Theta(r^2)$. Trial encryptions in the second stage require prior knowledge of the encryption algorithm. However this is not mandatory provided we can ask the smart card to load a chosen key and perform encryption using this key. Nor is it if the location of the bit stuck at 0 can be chosen.

In [76] P. Paillier examined the following modified model (probably more realistic): when the card is exposed to the physical constraint mentioned above, in addition to the $1 \rightarrow 0$ transitions, it is also possible for a 0-bit to flip at 1 with a small probability $p_0 < p_1$. Under this hypothesis he shows that the sequence $(K_i)_i$ does not converge to 0: as the Hamming weight decreases, the number of candidates for a $0 \rightarrow 1$ transition grows, while the number of candidates for $1 \rightarrow 0$ decreases. If p_0 is not small enough compared to p_1, an equilibrium is reached at some Hamming weight; this can make the complexity of the attack much bigger.

Differential Fault Attack against DES.

Most fault attacks against block ciphers are *differential*. It means that a right ciphertext (resulting from an encryption without induction of a fault) is considered together with a faulty ciphertext corresponding to the same plaintext. The comparison of both encryptions allows the retrieval of key material using techniques related to those of differential cryptanalysis [17] (hence the name).

E. Biham and A. Shamir presented a differential fault attack on DES. The faulty ciphertexts used in the attack are assumed to result from one bit being flipped in the register keeping the right half of the data in one of the 16 rounds. The index of the round affected and the precise location of the bit are unknown; there are thus $16 \cdot 32 = 512$ possibilities. However by observing the difference between the right and the faulty ciphertext, the attacker can deduce whether the round affected is the 16^{th}, the 15^{th}, the 14^{th}, or one of the rounds before. If the fault occurred before the 11^{th} round, it is not exploitable by Biham-Shamir's attack.

Assume the round affected is the last one. The output difference of the last round function is equal to the difference in the left part of the ciphertext[2]; and the data entering this round function is equal to the right part of the ciphertext. The attack is simple: we guess the key bits entering the S-box (or the two S-boxes) affected by the fault, and check whether the

[2] The final permutation FP is neglected.

guessed value agrees with the expected difference at the output of this (these) S-box(-es). On average, about 4 possible 6-bit values of the key remain for each of the S-boxes concerned.

If the round affected is the last but one, the attack is very similar. If it is the last but two (or earlier), a counting method [17] must be used, where for each S-box a counter is associated to each 6-bit candidate, and the right value is expected to be counted more than the others.

Assuming faults occur randomly in all rounds, E. Biham and A. Shamir found that between 50 and 200 ciphertexts were needed to retrieve the key. If the attacker can choose the exact position of the fault, this number can be reduced to about 3 ciphertexts.

If the induced faults affect several bits simultaneously, the attack still works; in fact it works even better because more S-boxes are affected (but on the other hand it could be more difficult to identify the round during which the fault occurred). In [40] C. Giraud and H. Thiebeauld claimed to have succeeded in retrieving the key using 2 ciphertexts only, in a real attack against a smart card.

An Attack Against SPN Structures and the AES.

In [77] an attack working against any substitution-permutation network is presented. Assume the cipher deals with blocks of n bytes, and a fault disturbs one (and only one) entire byte. The attack relies on the following observation: there are $255n$ possibilities for a fault occurring before the last diffusion layer (n possible positions, and 255 possible values). After this layer, the number of possible differences resulting from such fault is still $255n$ by linearity. However the difference is now spread out on several bytes, due to the diffusion effect. Therefore it is possible to apply a classical "key guess" approach: the attacker guesses the last round key, and uses it to compute the difference after the last diffusion layer from the right and the faulty ciphertext. She then checks whether the computed difference is amongst the $255n$ differences that could have been caused by a fault, and reject the key candidates for which it is not the case.

As such, this approach is not very practical, as it requires to try all the possibilities for the last key guess. However a clever implementation of it is possible: two bytes are initially guessed, and the possible candidates for these two bytes are progressively lengthened.

In the case of AES[3], the diffusion layer is not optimal, in the sense that a difference on one byte before the diffusion layer implies a difference on 4 bytes only at its output (corresponding to the output of a given `MixColumns`). Therefore one fault at the input of the last diffusion layer allows to retrieve information on 4 key bytes only. However in [77] G. Piret and J.-J. Quisquater remark that inducing a fault before the 8^{th} (rather than the 9^{th}) diffusion layer, one obtains a difference on one byte at the input of each `MixColumns` of the last diffusion layer. So we obtain information on all 16 bytes of the last round key instead of information on only 4 bytes. By using this attack the key could be retrieved with only 2 faulty ciphertexts.

Several people tried to do fault attacks against the AES. Table 4.1 summarizes their results and compares them to the attack of [77].

[3] We focus on AES-128 for simplicity, but the same principle can be applied to AES-192 and AES-256.

Table 4.1: Comparison of existing fault attacks against the AES. θ_i stands for the i^{th} diffusion layer of AES, and R for the number of rounds.

Ref.	Fault Model	Fault Location	# Faulty Enc.
[21]	Force 1 bit to 0	Chosen	128
[21]	Fct of impl.	Chosen	256
[39]	Switch 1 bit	Any bit of chosen bytes	~ 50
[39]	Disturb 1 byte	Anywhere among 4 bytes (including in the key schedule)	~ 250
[27]	Disturb 1 byte	Anywhere among 3 bytes in the key schedule	usually 32
[36]	Disturb 1 byte	Anywhere between θ_{R-3} and θ_{R-2}	10
Our result	**Disturb 1 byte**	**Anywhere between $\theta_{\mathbf{R-3}}$ and $\theta_{\mathbf{R-2}}$**	**2**

Exploiting Faults Occurring During the First Few Rounds.

The attacks presented until now exploit faults occurring during the last few rounds of encryption. Assessing the security of the cipher against faults occurring elsewhere is important as well, in order to know which rounds have to be protected. Remark that if we assume that the block cipher reduced to half of its rounds is secure, then it is not possible to exploit faults occurring at the middle of the computation, as it would be in contradiction with this security assumption. On the other hand, assume an attacker is allowed to obtain right and faulty encryptions of plaintexts, where the faults are induced during the first few rounds; moreover, she has access to a decryption oracle as well (but without being able to induce faults during it). Then the following attack can be applied: the attacker asks for the encryption of a plaintext P, such as to obtain a faulty ciphertext C^*. This ciphertext is then decrypted to obtain the corresponding plaintext P^*. The pair (P, P^*) can be used in an attack similar to those of the previous sections, as if they were ciphertexts.

The question is whether faults occurring during the first few rounds are exploitable when a decryption oracle is *not* available. This problem has been dealt with by L. Hemme in [48]. The scenario requires for the attacker to be able to choose the plaintexts P to encrypt. The principle is to try to get a collision with another plaintext $P_\bullet$, in the sense that the faulty ciphertext C^* corresponding to P equals the correct ciphertext $C_\bullet$ corresponding to $P_\bullet$. To this end, principles of differential cryptanalysis are used. Using this attack and exploiting faults occurring during the second round of DES, the number of correct and faulty encryptions required to find the DES key are respectively about 8000 and 500. The attack could theoretically be applied to other block ciphers. However, it is not possible for AES because the probability of the differentials through it is much smaller than for DES.

4.3.2 Stream Ciphers.

The first fault attack against stream cipher constructions only came up in 2004. It was published at the CHES conference by J.J. Hoch and A. Shamir [49]. In that paper, attacks against several constructions are presented:

- LFSR(s) of which the output is filtered through a non-linear function.
- *Data LFSR* of which the clocking is controlled by a *clock LFSR.*
- LFSR(s) of which the output is filtered through a *Finite State Machine* (FSM).

The linearity of LFSRs is at heart of these attacks. In [49] they are applied to stream ciphers LILI-128 and SOBER-t32. A dedicated attack against RC4 is also presented. However two better attacks on RC4 have been presented at FSE'05 [16]. One of them relies on the new concept of *impossible fault analysis.* The principle is that some internal states of RC4 are impossible to reach (this observation has been made in 1994 by H. Finney); they can only be obtained by fault induction. Moreover, these states are easy to identify and permit to retrieve the internal state and then the key-dependent initial state.

It is worth noting that attacking stream ciphers requires most of the time to have the algorithm in exactly the same state several times. Such reinitialization of the state is not always possible in real-life applications, which is an important limitation of these attacks.

4.3.3 Asymmetric Primitives.

Attack of CRT Implementation of RSA.

The first attack on RSA-CRT is due to D. Boneh, R. DeMillo and R. Lipton and was published in [22, 23]. It requires the attacker to obtain one faulty signature together with the correct one; knowledge of the signed message is not mandatory. This attack was improved by M. Joye, A. Lenstra and J.-J. Quisquater in [54]. We describe the improved attack below. It is one of the most powerful known fault attack, as it requires very few hypothesis on the fault induced.

Let $N = p \cdot q$ be the product of two large primes, e be the public exponent such that $\gcd(e, (p-1)(q-1)) = 1$, and $d = e^{-1} \mod (p-1)(q-1)$ the corresponding private exponent. An RSA signature of a message m is computed as $S = \mu(m)^d \mod N$, where μ is a given deterministic padding function[4]. However the trivial implementation of signature (direct exponentiation $\mod N$ using square-and-multiply) is not the fastest one: a speed factor of 4 can be gained by using the *Chinese Remainder Theorem.* The signer first computes $S_p = \mu(m)^d \mod p$ and $S_q = \mu(m)^d \mod q$. He then computes the signature S as

$$S = S_q + q \cdot (a_q \cdot (S_p - S_q) \bmod p), \tag{4.1}$$

where $a_q := p^{-1} \bmod q$ is a precomputed value. (4.1) is known as *Garner's formula.*

Therefore, instead of an exponentiation modulo N, the signer needs to do two exponentiations with moduli p and q. The fact that p and q have about twice less bits than N, makes the whole algorithm faster than a trivial implementation.

Assume a fault occurs during the computation of the exponentiation $S_q = \mu(m)^d \mod q$ (a fault on S_p would have the same effect). Then a wrong $\widetilde{S}_q$ is obtained, and with overwhelming

[4] The attack does not work against a probabilistic padding scheme such as PSS.

probability $\widetilde{S}_q \not\equiv S_q \pmod{q}$. Let $\widetilde{S} = \widetilde{S}_q + q \cdot (a_q \cdot (S_p - \widetilde{S}_q) \bmod p)$ be the wrong signature obtained. Then

$$\begin{aligned} & amp; \widetilde{S} \mod p = S_p \mod p = S \mod p \\ \text{and} \quad & amp; \widetilde{S} \mod q = \widetilde{S}_q \mod q \neq S \mod q. \end{aligned}$$

So we have $\widetilde{S}^e \equiv S^e \equiv \mu(m) \pmod{p}$ but $\widetilde{S}^e \not\equiv S^e \equiv \mu(m) \pmod{q}$. This implies that $\gcd(N, \mu(m) - \widetilde{S}^e) = p$. As all variables of the left-hand side are known to the attacker, she can compute p.

For this fault attack to succeed, it is only needed that a fault injected during the computation corrupted either S_p or S_q, but not both; alternatively, a fault induced in a_q would have the same effect. The number of bits that are flipped and their position is not important. The attack requires knowledge of the message m to be signed and of one faulty signature. The right signature is not even mandatory. It means that someone receiving a certificate with a false authority's signature, could retrieve the private key of the system.

Attacks on "Modular Exponentiation-Based" Cryptosystems.

Even when the Chinese Remainder Theorem is not used to implement them, public key cryptosystems are vulnerable to fault attacks. We describe here an attack on RSA decryption first published in [11]. Several attacks work along the same lines, and allow to retrieve one bit of the key per faulty computation. [11] describe attacks on the ElGamal, Schnorr and DSA signature schemes. The attack on ECDSA described in [35] basically relies on the same principle.

The attack against RSA decryption works as follows: an attacker chooses a plaintext m at random, and computes the ciphertext c. She then asks for the decryption of c and induces a fault during it, corresponding to the flip of one bit of the decryption exponent d. A faulty plaintext $\widetilde{m}$ is obtained. Assuming that bit $d[i]$ flips to $\overline{d[i]}$, dividing the faulty plaintext by the correct one yields

$$\frac{\widetilde{m}}{m} = \frac{c^{2^i \overline{d[i]}}}{c^{2^i d[i]}} \pmod{N}.$$

If

$$\frac{\widetilde{m}}{m} = \frac{1}{c^{2^i}} \pmod{N} \quad \text{then} \quad d[i] = 1,$$

and if

$$\frac{\widetilde{m}}{m} = c^{2^i} \pmod{N} \quad \text{then} \quad d[i] = 0.$$

So one bit flip allows the attacker to retrieve one bit of d. The attack is repeated until enough bits of d are known.

This attack has been extended and generalized by M. Joye et al. in [55]. In the case of RSA, the new attack requires the faulty decryption only.

Attacks on Elliptic Curves Cryptosystems.

A simple attack against elliptic curves multiplication, first published in [15], consists in shifting the computation from a given secure elliptic curve E (in Weierstrass parametrization) to an insecure one E'. Consider a smart card that has to compute $d \cdot P$, for d a scalar and P a point on the curve

$$E \equiv y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

If a fault is induced on P, it is changed into a point P' of a curve

$$E' \equiv y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6'.$$

The attack exploits the fact that the parameter a_6 is not used in the usual point addition formula on Weierstrass elliptic curves. As a consequence, the whole computation is performed on the curve E', and $d \cdot P'$ is obtained. If now E' is an insecure curve (typically because $\mathrm{ord}(E')$ has a small factor), the discrete logarithm problem can be solved on it, which gives information about d.

Faults occurring during the computation of $d \cdot P$ are also exploitable. We refer to [15] for more details. However, the attack assumes that only a few bits of errors are inserted in order to be successful. This hypothesis has been relaxed in [29], which uses random and unknown faults either in the base point P, in the base field $\mathbb{F}_p$ or $\mathbb{F}_{2^q}$ underlying to the curve, or in the curve's parameters. Yet another attack, which does not change the curve or the base point, but exploits changes in the sign of a point during the computation, is presented in [20].

4.4 Countermeasures

Countermeasure against fault attacks can be deployed in software or hardware and generally help circuits to detect and/or correct faults. Three categories of countermeasures are usually considered: passive protections, active protections and redundancies. Depending of the proposals, single or multiple-bit faults can be detected, corrected or prevented. The implementation cost of the countermeasure usually depends on quality of the fault coverage.

1. Passive protections such as randomization of the clock cycles or bus and memory encryption may be used to increase the difficulty of successfully attacking a device. Typical examples of such protections use data-scrambling functions as suggested in [24, 43].

2. Active protections use detectors to react to any abnormal circuit behaviors. Typical such sensors include [88]: (1) Light detectors that detect changes in the gradient of light. (2) Supply voltage detectors that react to abrupt variations in the applied potential. (3) Frequency detectors that impose an interval of operation outside which the electronic circuit will reset itself. (4) Active shields: metal mashes that cover the entire chip and has data passing continuously in them.

3. Redundancies. In practice, the most investigated countermeasures are based on classical error detection/correction codes, using space or time redundancies. For example, a number of naive solutions are proposed in [88], including simple duplication, multiple duplication, complementary redundancies, etc. A number of other countermeasures use similar principles, but aim to reduce the actual overhead to less than the cost of duplication/repetition codes. We detail here a number of these solutions that mainly apply to block ciphers. Remark that proposals related to asymmetric primitives also exist, e.g. in [10, 83]. In addition, redundancies can be added at the technological level, e.g. by the use of dual-rail logic including an alarm mechanism [86, 93].

- **Parity codes** are proposed in [14, 61, 96] as a possible solution to increase the security of block cipher implementations against fault attacks. Such proposal result in a tradeoff between the fault coverage and the hardware overhead. Basic solutions (*e.g.* one parity bit for the whole block cipher [14, 61]) are less expensive than duplication, but they don't allow to detect faults of all multiplicities. Fault coverage can then be improved by increasing the number of parity bits [96], at the cost of a higher hardware cost. In general, parity codes are anyway susceptible to the cancellation of the parity bits and are not a perfect countermeasure.
- **Non-linear robust codes** are another proposal proposed in [59, 60] in order to protect the AES Rijndael against fault attacks. Their efficiency actually depends on the block cipher operations and cannot be stated in general. For Rijndael, they allow to reach very high fault coverage and to deal with faults of all multiplicities. The major drawback is the implementation cost. Regarding the ratio throughput/area which gives a good estimation of hardware efficiency, the solution in [60] is equivalent to duplication.
- **Repetition/duplication in certain specific contexts**
    - **Block ciphers in feedback modes.** When block ciphers are used in feedback modes, pipelining cannot be used for throughput improvement. However, it can still be used to deal twice with the same plaintext. As a result, a repetition code is obtained, with a timing overhead less than 100%. Such idea was applied to involutional ciphers in [53], where the permanent faults can additionally be detected (which is not the case of repetition codes in general).
    - **Encryption/decryption designs** If an encryption/decryption block cipher design is used in a mode such that encryption and decryption don't have to be used concurrently, the inverse operation can be used to detect faults, by simply checking if $f^{-1}(f(x)) = x$. Again, this is theoretically equivalent to duplication or repetition, but in practice, the overhead may be transparent to the user.

Chapter 5

Conclusions and Future Research

This deliverable described state of the art physical attacks reported against cryptographic devices, with discussions about measurements, post-processing, algorithms and countermeasures. We focused on electromagnetic analysis and fault insertion attacks.

Over many aspects, attacks based on the electromagnetic leakage have proved their relative efficiency compared to power analysis. The unintentional emanations of integrated circuits possess a variety of sources that may independently reveal different information about the secret stored on a chip. Monitoring the electric and/or magnetic fields uncovers different parts of sensitive information. Moreover, researchers have shown that some information (such as one bit of the key) may modulate an inner carrier (e.g. multiples of the work frequency) and so be demodulated in the far-field (even a few meters away from the chip) with appropriate equipment.

Issues for further research include how to obtain sufficiently good measurement setups (oscilloscope, amplifier, filter, probes,...) to avoid many noisy effects. The success of an electromagnetic attack depends greatly on the ability of the attacker to use properly this setup (calibration process, characterization of spurious signal due to equipment imperfections are two classical instances).

Behind these practical investigations, future research may be oriented towards the theoretical modelling of the underlying physical processes (including a deeper understanding of electromagnetism theory related to integrated circuits) and finally towards the development of (possibly provably) secure countermeasures in these relevant models. Note that this requirement for secure and efficient countermeasures is generally an open question for side-channel attacks, not specifically for EMA.

A similar picture can be painted on the topic of fault attacks. As EMA, they proved to be a very efficient way to compromise a variety of devices and cryptographic algorithms. In practice, the question of the techniques for fault insertion and physical processes (shields, ...) to avoid them is still open, mainly in terms of accuracy: how precisely can faults be inserted, temporally and spacially?

Behind practice, a general framework for the modelling of fault attacks is a scope for future research as well as the development of good countermeasures. As mentioned in this report, a number of the present error correcting solutions have a cost similar to naive solutions (repetition, duplication). Finding more efficient ways to protect circuits at the algorithmic/coding level is consequently challenging.

Acknowledgements: Marc Joye (G+), Eric Peeters (UCL).

Bibliography

[1] http://en.wikipedia.org/wiki/fault.

[2] FIPS PUB 140-2, Security Requirements for Cryptographic Modules, 2001.

[3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM side-channel(s): Attacks and assessment methodologies. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[4] D. Agrawal, J. R. Rao, and P. Rohatgi. Multi-channel attacks. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 2–16, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[5] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM Side-channel(s). In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002*, volume 2535 of *Lecture Notes in Computer Science*, pages 29–45. Springer, 2003.

[6] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power analysis, what is now possible... In Tatsuaki Okamoto, editor, *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology - ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 489–502, Kyoto, Japan, December 3-7 2000. Springer-Verlag.

[7] M.-L. Akkar and C. Giraud. An implementation of DES and AES, secure against some attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318, Paris, France, May 13-16 2001. Springer-Verlag.

[8] Ross Anderson and Markus Kuhn. Tamper Resistance — A Cautionary Note. In *The Second USENIX Workshop on Electronic Commerce Proocedings*, pages 1–11, 1996.

[9] Ross J. Anderson and Markus G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In Christianson et al. [28], pages 125–136.

[10] Christian Aumüller, Peter Bier, Wieland Fischer, Peter Hofreiter, and Jean-Pierre Seifert. Fault attacks on rsa with crt: Concrete results and practical countermeasures. In Jr. et al. [57], pages 260–275.

[11] F. Bao, R.H. Deng, Y. Han, A.B. Jeng, A.D. Narasimhalu, and T.-H. Ngair. Breaking Public Key Cryptosystems on Tamper Resistant Devices in the Presence of Transient Faults. In Christianson et al. [28], pages 125–136.

[12] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The Sorcerer's Apprenctice's Guide to Fault Attacks. Technical report, 2004.

[13] L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Poncino, and F. Pro. Energy-aware design techniques for differential power analysis protection. In *Proceedings of the 40th Design Automation Conference (DAC)*, Anaheim, CA, USA, June 2-6 2003.

[14] G. Bertoni, L. Breveglieri, I. Koren, and P. Maistri. An efficient hardware-based fault diagnosis scheme for aes: Performance and cost. In *proceedings of DFT 2004*, 2004.

[15] I. Biehl, B. Meyer, and V. Muller. Differential Fault Attacks on Elliptic Curve Cryptosystems. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, Santa Barbara, USA, August 20-24, 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 131–146. Springer-Verlag, 2000.

[16] E. Biham, L. Granboulan, and P. Nguyen. Impossible Fault Analysis and Differential Fault Analysis of RC4. In *Preproceedings of FSE 2005*, pages 371–379, 2005.

[17] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[18] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In Burton S. Kaliski, editor, *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer-Verlag, 1997.

[19] I. Blake, G. Seroussi, and N. P. Smart. *Elliptic Curves in Cryptography.* London Mathematical Society Lecture Note Series. Cambridge University Press, 1999.

[20] J. Blomer, M. Otto, and Seifert J.-P. Sign Change Fault Attacks On Elliptic Curve Cryptosystems. Technical Report 2004/227, IACR eprint archive, 2004. Available at http://eprint.iacr.org/2004/227.

[21] J. Blömer and J.-P. Seifert. Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). In Rebecca N. Wright, editor, *Financial Cryptography, 7th International Conference, FC 2003, Guadeloupe, January 27-30, 2003*, Lecture Notes in Computer Science, pages 162–181. Springer-Verlag, 2003. Also available at http://eprint.iacr.org/, 2002/075.

[22] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[23] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Eliminating Errors in Cryptographic Computations. *Journal of Cryptology*, 14(2):101–119, 2001.

[24] E. Brier, H. Handschuh, and C. Tymen. Fast primitives for internal data scrambling in tamper resistant hardware. In *proceedings of CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 16–27. Springer-Verlag, 2001.

[25] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier. Electromagnetic side channels of an FPGA implementation of AES. Cryptology ePrint Archive-2004/145, 2004. `http://eprint.iacr.org/`.

[26] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In M. Wiener, editor, *Advances in Cryptology: Proceedings of CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412, Santa Barbara, CA, USA, August 15-19 1999. Springer-Verlag.

[27] C.-N. Chen and S.-M. Yen. Differential Fault Analysis on AES Key Schedule and Some Countermeasures. In R. Safavi-Naini and J. Seberry, editors, *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003*, volume 2727 of *Lecture Notes in Computer Science*, pages 118–129. Springer-Verlag, 2003.

[28] Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors. *Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings*, volume 1361 of *Lecture Notes in Computer Science*. Springer, 1998.

[29] M. Ciet and M. Joye. Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Faults. Design, Codes and Cryptography, To appear, 2004.

[30] M. Ciet, M. Neve, E. Peeters, and J.-J. Quisquater. Parallel FPGA implementation of RSA with residue number systems -can side-channel threats be avoided? In *IEEE Midwest 2003 Proceedings*.

[31] G. M. Clarke and D. Cooke. *A basic course in statistics*. Arnold London, 4th edition, 1998.

[32] J.-S. Coron and L. Goubin. On boolean and arithmetic masking against differential power analysis. In Ç. K. Koç and C. Paar, editors, *Proceedings of 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 231–237, Worcester, Massachusetts, USA, August 17-18 2000. Springer-Verlag.

[33] J. Daemen and V. Rijmen. Resistance against implementation attacks: A comparative study of the AES proposals. In *Proceedings of the Second Advanced Encryption Standard (AES) Candidate Conference*, March 1999. `http://csrc.nist.gov/encryption/aes/round1/Conf2/aes2conf.htm`.

[34] J. Daemen and V. Rijmen. Rijndael for AES. In *Proceedings of The Third Advanced Encryption Standard Candidate Conference*, pages 343–348, New York, NY, USA, April 13-14 2000. National Institute of Standards and Technology.

[35] E. Dottax. Fault Attacks on NESSIE Signature and Identification Schemes. NESSIE Report, October 2002. Available from http://www.cryptonessie.org/nessie/reports/phase2/SideChan_1.pdf.

[36] P. Dusart, G. Letourneux, and O. Vivolo. Differential Fault Analysis on A.E.S. In J. Zhou, M. Yung, and Y. Han, editors, *Applied Cryptography and Network Security, First International Conference, ACNS 2003. Kunming, China, October 16-19, 2003*, volume 2846 of *Lecture Notes in Computer Science*, pages 293–306. Springer-Verlag, 2003.

[37] J. J. A. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor. Security evaluation of asynchronous circuits. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 137–151, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[38] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 255–265, Paris, France, May 13-16 2001. Springer-Verlag.

[39] C. Giraud. DFA on AES. Technical Report 2003/008, IACR eprint archive, 2003. Available at http://eprint.iacr.org/2003/008.ps.

[40] C. Giraud and H. Thiebeauld. A Survey on Fault Attacks. In J.-J. Quisquater, P. Paradinas, Y. Deswarte, and A.A. El Kalam, editors, *Smart Card Research and Advanced Applications VI - 18th IFIP World Computer Congress*, pages 159–176. Kluwer Academic Publishers, August 2004.

[41] J. D. Golìc. DeKaRT: A new paradigm for key-dependent reversible circuits. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 98–112, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[42] J. D. Golić and C. Tymen. Multiplicative masking and power anaylsis of AES. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, number 2535 in Lecture Notes in Computer Science, pages 198–212, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[43] J.D. Golić. Dekart: A new paradigm for key-dependent reversible circuits. In *proceedings of CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 98–112. Springer-Verlag, 2003.

[44] L. Goubin. A sound method for switching between boolean and arithmetic masking. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of the 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 3–15, Paris, France, May 13-16 2001. Springer-Verlag.

[45] L. Goubin and J. Patarin. DES and differential power analysis (the "duplication" method). In Ç. K. Koç and C. Paar, editors, *Proceedings of the 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172, Worcester, Massachusetts, USA, August 12-13 1999. Springer-Verlag.

[46] J. C. Ha and S. J. Moon. Randomized signed-scalar multiplication of ECC to resist power attacks. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *Lecture Notes in Computer Science*, pages 129–143, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[47] M. A. Hasan. Power analysis attacks and algorithmic approaches to their countermeasures for Koblitz curve cryptosystems. *IEEE Transactions on Computers*, 50(10):1071–1083, October 2001.

[48] L. Hemme. A Differential Fault Attack Against Early Rounds of (Triple-)DES. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004, Cambridge, USA, August 11-13, 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 254–267. Springer-Verlag, 2004.

[49] J.J. Hoch and A. Shamir. Fault Analysis of Stream Ciphers. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004, Cambridge, USA, August 11-13, 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 240–253. Springer-Verlag, 2004.

[50] International Electrotechnical Commission (IEC). IEC 61967: Integrated Circuits - Measurement of Electromagnetic Emissions, 150 kHz to 1 GHz, 2003. Available online at `http://www.iec.ch`.

[51] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[52] T. Izu and T. Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. In D. Naccache and P. Paillier, editors, *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC)*, volume 2274 of *Lecture Notes in Computer Science*, pages 280–296, Paris, France, February 12-14 2002. Springer-Verlag.

[53] N. Joshi, K. Wu, and R. Karry. Concurrent error detection schemes for involution ciphers. In *proceedings of CHES 200*, volume 3156 of *Lecture Notes in Computer Science*, pages 400–412. Springer-Verlag, 2004.

[54] M. Joye, A. Lenstra, and J.-J. Quisquater. Chinese Remaindering Based Cryptosystems in the Presence of Faults. *Journal of Cryptology*, 12(4):241–245, 1999.

[55] M. Joye, J.-J. Quisquater, F. Bao, and R.H. Deng. RSA-type Signatures in the Presence of Transient Faults. In Michael Darnell, editor, *Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17-19, 1997*, volume 1355 of *Lecture Notes in Computer Science*, pages 155–160. Springer-Verlag, 1997.

[56] M. Joye and C. Tymen. Protections against differential analysis for elliptic curve cryptography. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 377–390, Paris, France, May 13-16 2001. Springer-Verlag.

[57] Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.

[58] M. Akmal K. Tiri and I. Verbauwhede. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of 28th European Solid-State Circuits Conference (ESSCIRC)*, pages 403–406, 2002.

[59] M. Karpovsky, K.J. Kulikowski, and A. Taubin. Differential fault analysis attack resistant architectures for the advanced encryption standard. In *proceedings of CARDIS 2004*.

[60] M. Karpovsky, K.J. Kulikowski, and A. Taubin. Robust protection against fault injection attacks on smart cards implementing the advanced encryption standard. In *proceedings of DSN 2004*.

[61] R. Karri, G. Kuznetsov, and M. Gössel. Parity-based concurrent error detection of substitution-permutation network block ciphers. In *proceedings of CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 113–124. Springer-Verlag, 2003.

[62] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48:203–209, 1987.

[63] Oliver Kömmerling and Markus G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Proceedings of the USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 9–20, 1999.

[64] Kerstin Lemke and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Fault Diagnosis and Tolerance in Cryptography — FDTC 2005, 2nd International Workshop Edinburgh (Scotland), UK, September 2nd, 2005*, 2005.

[65] Regís Leveugle. Early Analysis of Fault Attack Effects for Cryptographic Hardware. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.

[66] P. Y. Liardet and N. P. Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 391–401, Paris, France, May 13-16 2001. Springer-Verlag.

[67] P.-Y. Liardet and Y. Teglia. From Reliability to Safety. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.

[68] S. Mangard. Exploiting radiated emissions - EM attacks on cryptographic ICs. In *Proceedings of Austrochip*, Linz, Austria, October 3 2003.

[69] D. May, H. Muller, and N. Smart. Random register renaming to foil DPA. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of the 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 28–38, Paris, France, May 13-16 2001. Springer-Verlag.

[70] T. S. Messerges. *Power Analysis Attacks and Countermeasures on Cryptographic Algorithms.* PhD thesis, University of Illinois, 2002.

[71] V. Miller. Uses of elliptic curves in cryptography. In H. C. Williams, editor, *Advances in Cryptology: Proceedings of CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Santa Barbara, CA, USA, August 18-22 1985. Springer-Verlag.

[72] S. Moore, R. Anderson, R. Mullins, G. Taylor, and J. Fournier. Balanced self-checking asynchronous logic for smart card applications. *to appear in the Microprocessors and Microsystems Journal*, 2003.

[73] Michael Neve, Eric Peeters, David Samyde, and Jean-Jacques Quisquater. Memories: a Survey of their Secure Uses in Smart Cards. Technical report. http://www.dice.ucl.ac.be/ mneve/document/Publications/sisw03.pdf.

[74] NSA. NSA TEMPEST Documents, 2003. `http://www.cryptome.org/nsa-tempest.htm`.

[75] E. Oswald and M. Aigner. Randomized addition-subtraction chains as a countermeasure against power attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of the 3rd International Workshop on Cryptograpic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 39–50, Paris, France, May 14-16 2001. Springer-Verlag.

[76] P. Paillier. Evaluating Differential Fault Analysis of Unknown Cryptosystems. In H. Imai and Y. Zheng, editors, *Second International Workshop on Practice and Theory in Public Key Cryptography, PKC '99, Kamakura, Japan, March 1-3, 1999*, volume 1560 of *Lecture Notes in Computer Science*, pages 235–244. Springer-Verlag, 1999.

[77] Gilles Piret and Jean-Jacques Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2003.

[78] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smard cards. In *Proceedings of ESmart 2002*, pages 185–194, 2002.

[79] David Samyde and Jean-Jacques Quisquater. Eddy Current for Magnetic Analysis with Active Sensor. In *Proceedings of ESmart 2002*, pages 185–194, 2002.

[80] David Samyde, Sergei Skorobogatov, Ross Anderson, and Jean-Jacques Quisquater. On a New Way to Read Data from Memory. Technical report. http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/SISW02.pdf.

[81] P. Schaumont and I. Verbauwhede. Domain specific codesign for embedded security. *IEEE Computer Magazine*, 36(4):68–74, April 2003.

[82] R. A. Serway. *Physics for scientists and engineers.* Saunders Golden sunburst series. Saunders college publishing, 1996.

[83] A. Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks, 1999. U.S. Patent Number 5,991,415, November 1999; also presented at the rump session of EUROCRYPT97.

[84] A. Shamir. Protecting smart cards from passive power analysis with detached power supplies. In C. Paar and Çetin Koç, editors, *Proceedings of the 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 71–77, Worcester, Massachusetts, USA, Aug 17-18 2000. Springer-Verlag.

[85] Sergei Skorobogatov. Low temperature data remanence in static RAM. Technical Report UCAM-CL-TR-536, 2002. http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-536.pdf.

[86] Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. In Jr. et al. [57], pages 2–12.

[87] Sergei S. Skorobogatov. Semi-invasive attacks — A new approach to hardware security analysis. Technical report, 2005.

[88] H. Bar-El *et al.* The sorcerer's apprentice guide to fault attacks. Technical Report 2004/100, IACR eprint archive, 2004. Available at http://eprint.iacr.org.

[89] K. Tiri and I. Verbauwhede. Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 125–136, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[90] K. Tiri and I. Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Proceedings of Design, Automation and Test in Europe Conference (DATE)*, pages 246–251, February 2004.

[91] E. Trichina. Combinational logic design for AES subbyte transformation on masked data. Cryptology ePrint Archive: Report 2003/236, 2003.

[92] E. Trichina, D. De Seta, and L. Germani. Simplified adaptive multiplicative masking for AES. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2535 of *Lecture Notes in Computer Science*, pages 187–197, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[93] J.D. Waddle and D.A. Wagner. Fault attacks on dual-rail encoded systems. Tech report UCB//CSD-04-1347, UC Berkeley, August 23, 2004.

[94] C. D. Walter and S. Thompson. Distinguishing exponent digits by observing modular subtractions. In D. Naccache, editor, *Proceedings of Topics in Cryptology - CT-RSA*, volume 2020 of *Lecture Notes in Computer Science*, pages 192–207, San Francisco, 8-12 April 2001. Springer-Verlag.

[95] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses. In C. Paar and Çetin Koç, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, pages 302–317, 2000.

[96] K. Wu, R. Karri, G. Kuznetsov, and M. Goessel. Low cost error detection for the advanced encryption standard. In *proceedings of ITC 2004*, 2004.