Side Channel Attacks

Victor Lomné, Amine Dehbaoui, Philippe Maurine, Michel Robert, Lionel Torres

Chapter 3
Side Channel Attacks

V. Lomné, A. Dehaboui, P. Maurine, L. Torres, and M. Robert

Abstract This chapter presents the main Side-Channel Attacks, a kind of hardware cryptanalytic techniques which exploits the physical behavior of an IC to extract secrets implied in cryptographic operations. We show in this chapter the main modern concepts about Side Channel Attacks (Simple and Differential Power Analysis) and how they can be deployed on FPGA architecture. We give also a set of details on platform and equipment needed to conduct such type of experiments. Then we propose a discussion about the leakage model of digital IC, comprising FPGA, and we illustrate these attacks on a set of real case study. We conclude this chapter by giving the latest information and link toward new efficient Side Channel Attacks.

3.1 Introduction

In the past 100 years, we have seen the emergence of modern cryptography, along with many cryptographic primitives and protocols. The development of new theoretical cryptanalytic techniques to try to defeat the main cryptographic algorithms has increased knowledge of how to design cryptographic primitives and schemes.

In theoretical cryptanalysis today, a cryptographic algorithm is considered as a black box. Even when attackers know the cryptographic algorithm, they only have access to pairs of plaintexts/ciphertexts, and their goal is to guess the cryptographic key.

The robustness of modern cryptographic algorithms is based on these assumptions, and a cryptanalytic attack is a method that allows an assailant to guess the key with a complexity (in time and/or in memory) lower than a brute force attack.

But if a cryptographic algorithm is modeled as a gray box, i.e. in such a way that an attacker can obtain intermediate information during the cryptographic operation requiring the key, these assumptions no longer hold true.

In this context, and with the increasing use of embedded cryptographic devices with embedded cryptographic secrets, several tamper attacks have appeared since

L. Torres (✉)
LIRMM—UMR CNRS 5506, University of Montpellier 2, Montpellier, France
e-mail: lionel.torres@lirmm.fr

the beginning of the 1990s. These enable an attacker to obtain intermediate information during the cryptographic operation, and then to deduce the secret key with a complexity much lower than classical theoretic cryptanalytic attacks.

Figure 3.1 shows different models from theoretical and hardware cryptanalytic points of view. In the gray box model, physical leakages from the device can be identified or the computation running on the device can be disturbed. As we explain in this chapter, these different approaches enable cryptographic secrets to be extracted. Cryptanalytic techniques for hardware are classified as invasive attacks, semi-invasive attacks, or non-invasive attacks. We describe these attacks in detail in the following sections.

3.1.1 Invasive Attacks

Invasive attacks are a tamper attack in which the device is completely destroyed in order to extract secret information. The best known invasive attack is hardware reverse engineering, as described in [69]. The goal is to retrieve the layout of the circuit using chemistry techniques and/or high resolution microscopy. This technique is often used for cloning or anti-cloning purposes.

The first step consists in decapsulating the chip. A high resolution picture is then taken of each metal layer, and chemistry techniques are used to remove each succeeding metal layer to penetrate deeper into the chip right down to the transistor level. When attackers have a picture of each metal layer, they can use dedicated tools to go back to the netlist of the circuit. Finally, they obtain the behavioral description of the chip.

This attack has been conducted in the case of the MyFare stream encryption algorithm, which was kept secret by NXP. An invasive attack revealed the algorithm's functionality [46]. The method is shown in Fig. 3.2.

Although this technique is very powerful, but it has two drawbacks. Firstly, it requires very expensive equipment and highly qualified engineers. Secondly, thanks to technology shrinking techniques, it requires more and more sophisticated microscopy. In the case of SRAM and Flash based FPGAs, the configuration file, also called the bitstream, is stored in a dedicated off or on chip Flash memory. Thus, considering that the layout of an FPGA is roughly an array of reconfigurable cells, invasive attacks that allow attackers to obtain the hardware description of the attacked chip still provide no information about the configuration file, because the reconfigurable cells lose their configuration data as soon as the FPGA is switched off.

For instance, Fig. 3.3 shows the layout of an FPGA; as can be seen, optical observation of the floorplan provides no clues about the FPGA's configuration.

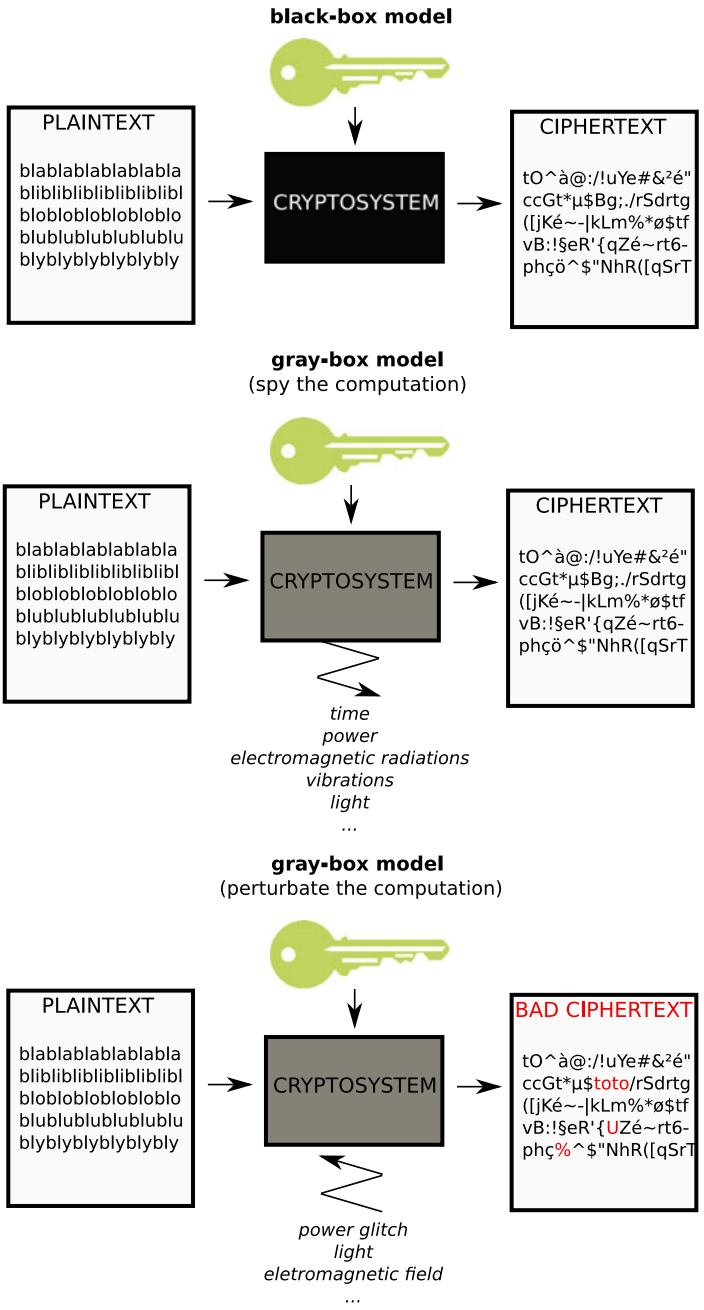

Fig. 3.1 Black-box and gray-box models from a cryptanalytic point of view

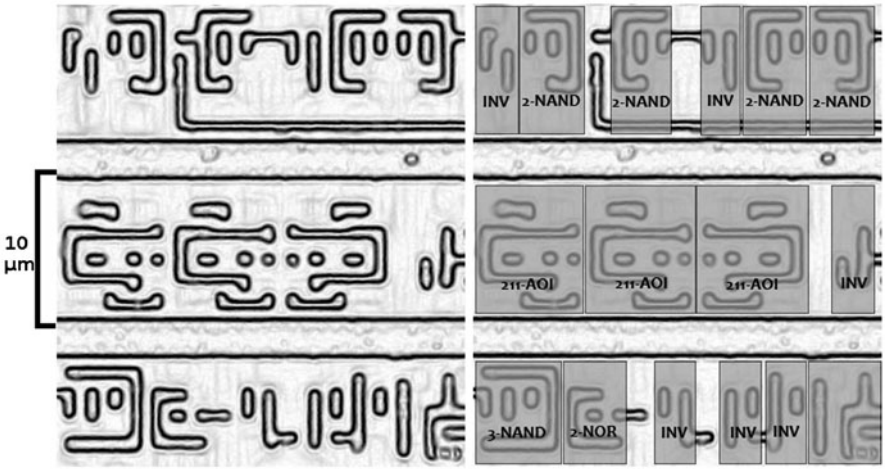

Fig. 3.2 Reverse-engineering of the MyFare CRYPTO1 algorithm. *On the left*, pictures of the circuit. *On the right*, netlist reconstruction

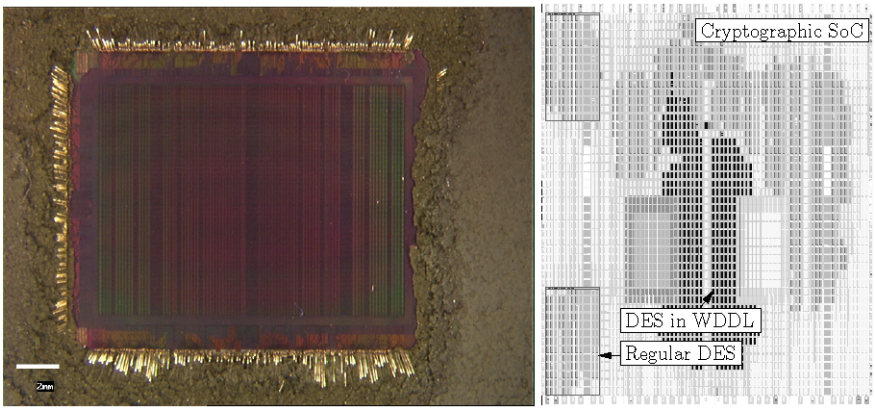

Fig. 3.3 Layout of a Stratix FPGA (ALTERA). *On the left*, a magnified photograph of the silicon die. *On the right*, the application actually programmed in the matrix, viewed under QUARTUS but invisible under the microscope

3.1.2 Semi-invasive Attacks

Semi-invasive attacks consist in retrieving physical information or disturbing the cryptographic operation computed by the device after decapsulation. These kinds of attacks are called semi-invasive because they require physical modification of the device, but do not destroy it. Indeed, attackers usually uses chemistry techniques to remove the package of the chip. In some cases, other sophisticated techniques may

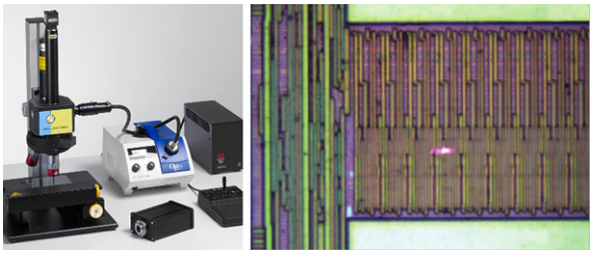

Fig. 3.4 Commercial laser attack setup from Riscure (*on the left*)—zoomed view of memory with the laser pulse in *pink* (*on the right*) (color online)

be required for example thinning down the substrate. For this purpose, different approaches can be used.

The first approach belongs to the *fault attacks* category. One or several faults are generated during the cryptographic operation using, for instance, light pulses. A concrete light based fault attack was first reported in [63]. Since these first concrete results, more sophisticated techniques have been elaborated using a laser (for instance red or infra-red diode laser), allowing the fault to be generated on either the front or back of the chip, and/or the fault to be focused on one or a few bits [8]. Figure 3.4 shows a commercial laser attack setup from Riscure (on the left), and zoomed view of memory with the tiny laser pulse in pink (on the right).

Another technique, called *probing attack* [32], consists in placing a small needle on a bus line of the microcontroller to retrieve information. Different authors have shown that even if attackers can only spy on one random bus line, they can nevertheless obtain enough information to reveal a cryptographic key. As an example, the authors of [62] describe a probing attack applied on AES. Recently, a general side-channel method based on single bit probing [24] was introduced as a way to optimize cube attacks [25]. Such attacks use a learning step, and can consequently even work on a proprietary algorithm.

A recent kind of semi-invasive attack consists in measuring, on the back side, photons emitted by transistors when they switch. This attack [13] requires thinning down the substrate and a high performance camera. Then, after the acquisition of photons emitted by the chip during the cryptographic computation of different data and the same unknown key, a statistical treatment, similar to that used in SCA reveals the cryptographic key. Moreover, this powerful technique follow the displacement of the current to be monitored by computing a movie disclosing the switching of transistors. However, this attack requires very expensive equipment that is available only in a few advanced laboratories.

3.1.3 Non-invasive Attacks

Non-invasive attacks are powerful techniques to extract secret information from cryptographic devices without physical modification of the device. As explained above, we distinguish two approaches, one consists in identifying physical leakages during the cryptographic operation (called *Side Channel Attacks* (SCA)), the other in disturbing computation (called *Fault Attacks* (FA)).

Side Channel Attacks consist in measuring physical leakages from the device during a cryptographic operation. The main idea behind this approach is that CMOS technology has an interesting property from a cryptanalytic point of view: it leaks physical information correlated with processed data. This class of attacks is very powerful, and requires only affordable measuring and testing equipment. Different physical leakages can be measured. Typical leaks are the computational time of the cryptographic operation [35], power consumption by the device [36], or the electromagnetic radiations emitted by the chip [26].

In some cases secret information can be extracted via a physical leakage from only one cryptographic operation, in others the attacker needs to record physical leakages from several cryptographic operations and to apply a statistical treatment on these records. On a classical smart card without dedicated countermeasures, several thousand records of cryptographic operations enable both symmetric ciphers (like DES or AES) and asymmetric ciphers (like RSA or ECC) to be broken in a very short computational time (a few hours using a desktop computer).

The first type of *Side-Channel Attacks* is the *Timing Attack*, which exploits differences in the computational time of a cryptographic operation. But from a practical point of view, methods have been proposed to implement a cryptographic algorithm in both hardware or software, ensuring that its computational time is constant.

However, power and electromagnetic side channels are physical leakages that are tightly correlated with the processed data due to CMOS properties, and these kinds of attacks are considered to be very powerful by manufacturers and government agencies. Moreover, using a tiny electric or magnetic sensor that is smaller than the chip, a 3D electromagnetic radiation map can be computed. This is achieved by running the same set of instructions several times, while placing the sensor in different positions.

The second approach belongs to the *fault attacks* category. The initial idea, from [17], is to run the cryptographic operation twice, one safe and another faulty. Dedicated cryptanalytic techniques enable one pair of safe/faulty cryptographic computations to be exploited to extract the key, with a low complexity in comparison to classical cryptanalytic techniques. In other cases a differential approach is required to retrieve the key [16], in which case the attacker needs several pairs of safe/faulty cryptographic computations to retrieve the cryptographic key.

The first concrete way proposed in the literature to generate faults during a computation is the *glitch attack* [10]. The idea is to modify the supplying signal of the device during a sharp time shorter than a clock period. The device can be overpowered or underpowered. As a result, a setup time or hold time violation will occur, and the attacker generates a logical error in the result of the cryptographic operation. A variant consists in suddenly modifying the clock period.

There is another way to generate faults in cryptographic devices without physically modifying the circuit. The idea is to place an electromagnetic sensor above the device and to generate an electromagnetic field through the chip. Although few works have reported concrete results of this kind of attack, concrete experimental results can be found in [61].

Some circuits can even be disturbed by a steady stress. For instance, paper [34] shows how to obtain incorrect computations from an AES implemented in an FPGA.

The method consists in underfeeding the device, so as to generate setup time violations. A comprehensive study of this way of injecting such faults in an FPGA is available in [15] for various FPGA families. The outcome of these experiments is that the description of the algorithm can greatly improve the resistance of the implementations against these "global faults".

In the rest of this chapter, we only deal with Power and Electromagnetic based Attacks will be detailed [67]. For more information on Timing Attacks, the reader can consult [35] or [19]. For information on Fault Attacks, [16, 17, 30] and [12] are good starting points.

The following explanations are applicable to for all kind of ICs used in embedded devices, and especially for FPGAs that use hardware or software cryptographic primitives.

3.2 Power and Electromagnetic Measurement Platform

In this section, we describe the platform used to perform power and/or electromagnetic measurements of an FPGA running cryptographic operations.

3.2.1 Equipment

One example of an overall platform is given in Fig. 3.5, it is made up of several parts described in the following sections. Other boards are described in [67]. Also of particular interest are the so-called "Side Channel Analysis Standard Evaluation BOards" (SASEBO) [54].

3.2.1.1 Attacked Device

For the experiments described in this chapter, the attacked device is an Xilinx FPGA Spartan3-1000 embedded in a Digilent board. The Spartan die is encapsulated in a Cavity Up Ball Grid Array (BGA) package, and is roughly 7 mm by 7 mm in size, whereas the whole package is roughly 17 mm by 17 mm.

3.2.1.2 Oscilloscope

The main component of the equipment required for a Side Channel Attack, (and the most expensive device in a Side Channel Attack platform) is an oscilloscope. A Lecroy 735Zi scope was used for the experiments described in this chapter. It belongs to the latest generation of Lecroy scopes, and has impressive properties:

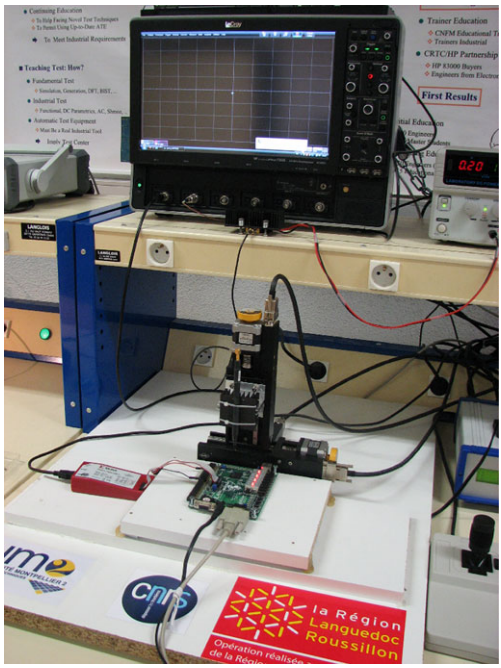

Fig. 3.5 Power and Electromagnetic Measurement Platform

- 4 channels;
- a maximum sampling rate of 40 GSa/s;
- a frequency bandwidth of 3.5 GHz;
- a resolution of 8 bits per point (without averaging);
- a vertical sensitivity of 2 mV;
- a maximum memory depth of 32 MSa.

Moreover, like other high-performance scopes, it allows remote control via Ethernet connections. Thus, once the scope is on, it can be fully controlled remotely. Furthermore, specific advanced modes for the trigger can be very useful for SCA purposes, but these functionalities were not exploited in our experiments.

3.2.1.3 Current Probes

To accurately measure the current consumed by the attacked device, current probes have interesting properties and are affordable. Tektronix current probes CT1 and CT6 [1] were used in the platform described here. First, the voltage regulator on the Digilent board supplying the power to the FPGA core was removed. Power was supplied by a battery, and the current probe was placed on the VDD or GND wire (between the battery and the pins of the core).

The current probe CT1 has a frequency bandwidth of 1 GHz and the CT6 of 2 GHz, and an accuracy of 5 mV/mA. Low-noise amplifier (described below) was

plugged at the output of the current probe to increase the amplitude of the signal enabling accurate measurement of power consumption.

3.2.1.4 Magnetic Sensors

The size and the shape of a magnetic sensor are the most important characteristics that determine the accuracy of the sensor. Hand-made magnetic sensors can be built. Attempts have often been made to build the smallest coiled wire, made of copper, and with several turns. Although these sensors produce good results (for instance some DEMA attacks succeeded with several hundred measurements), commercial magnetic sensors are often preferred [55].

Consequently, we used the Rhode and Schwarz HZ-15 probe set which contains different electric and magnetic sensors. Among them, the H field probe $RSH2.5\text{-}2$ gives accurate measurements, with good precision (its diameter is around 0.5 mm, thus when placed close to the IC, one is about 1 mm when working on encapsulated chips). Moreover, it allows precise measurements with a huge frequency bandwidth (spanning from 1 MHz to 3 GHz) even through the package of the FPGA. It consequently appears to be a good compromise to measure the magnetic activity of packaged chips.

3.2.1.5 Low-Noise Amplifier

To ensure sufficient power and accurate electromagnetic measurements, a low noise amplifier can be used. We used one with a frequency bandwidth of 1 GHz, and a gain of 63 dB.

3.2.1.6 Motorized X-Y-Z Stage

Using a tiny magnetic sensor to measure a particular area of the attacked chip requires positioning the sensor at the exact location required. For this task, we used a motorized X-Y-Z stage. The three axes have a precision of 50 µm, and the stage can be fully controlled remotely by serial connection.

3.2.1.7 Computer

A personal computer was used to control the whole platform. It was connected to the board embedding the attacked device via one serial port, to the scope via Ethernet, and to the X-Y-Z motorized stage via another serial port to drive it.

3.2.2 Acquisition Software

To automate the acquisition of measurements, we developed a software in Matlab. It runs on the computer controlling the whole platform. Typically, to perform an EM cartography, i.e. to obtain EM measurements of the same calculation(s) with the magnetic sensor placed at different positions above the chip to draw a map, the following steps are recommended:

- the initial position (X_{init}, Y_{init}, Z_{init}), the number of positions (nb_X, nb_Y), the displacement step (dis), the key (K) and the number of plaintexts (nb_{PTIs}) are chosen by the user;
- once the script is launched, *PTIs* are chosen randomly;
- the motorized stage is positioned at the first position (X_{init}, Y_{init}, Z_{init});
- the computer sends to the scope and adjusts it so that the entire computation is correctly recorded; it also sends the key K (optional) and the PTI_1 to the chip; the chip sends to the scope a trigger signal on the channel 1 to launch the recording of the magnetic signal on channel 2 (the magnetic sensor is connected to the low noise amplifier, which is connected to channel 2 of the scope);
- once the computation is finished, the scope sends the acquired EM trace to computer;
- the two previous steps are repeated nb_{PTIs} times;
- the motorized stage is positioned at the second position ($X_{init} + dis$, Y_{init}, Z_{init}), and the three previous steps are repeated;
- the same steps are repeated until all the positions have been covered.

The localization of the cryptographic modules is an advanced topic that is beyond the scope of this introduction concerned with side channel attacks. Interested readers should refer to [23, 52, 56, 57].

3.3 Leakage Models

Power and Electromagnetic (EM) based Side Channel Attacks is a hardware cryptanalytic technique that exploits physical leakages emitted by a cryptographic device. In this section, we explain how physical measurements can be realized and how leakage model can be used.

3.3.1 Power Consumption Leakage

Complementary Metal-Oxide-Semiconductor (CMOS) is the most used technology in microprocessors, microcontrollers, static RAM, and other digital logic circuits. Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Significant power is only drawn while the transistors in

the CMOS device are switching between on and off states. Consequently, CMOS devices do not produce as much waste heat as other forms of logic, like NMOS. CMOS also enables a high density of logic functions on a chip. It was primarily for these reasons that CMOS won the race in the 1980s and became the most widely used technology for VLSI chips.

The power consumption of a CMOS circuit is the sum of the power consumption of the logic cells that make up the circuit [31]. Hence, the total power consumption mainly depends on the number of logic cells in a circuit, the connections between them, and how the cells are built. The power consumption of a CMOS gate is generally considered in terms of two components [22]:

- *The dynamic power* component: this is mainly related to the charging and discharging of the load capacitance at the gate output, but also to the short circuit current that flows during the transition of the input from one voltage level to another. Indeed there is a short period during which both PMOS and NMOS transistors are on simultaneously, thus creating an electrical path between the VSS and VDD rails.
- *The static power* component: this is due to leakage in the substrate that flows even when the gate is not switching. In turn, this leakage is made up of several components including gate to source leakage, which flows directly through the gate insulator, mostly by tunneling, and source-drain leakage attributed to both tunneling and sub-threshold conduction. However, so far no research paper has reported an attack using this static leakage.

While the transistors that comprise the CMOS gate are switching between on and off states, a significant amount of power is consumed, due to the dynamic power consumption. This characteristic of the CMOS technology is interesting from a cryptanalytic point of view. Indeed, an attacker can use the fact that the dynamic power consumption is tightly correlated with the number of switching bits to guess a secret value in a cryptographic operation.

Moreover, electrical simulations in recent works [39] showed that, in deep-submicron CMOS technologies, the percentage of static power consumption among the global power consumption of a circuit increases with a decrease in the technology shrinking techniques used. As the static power consumption of a gate mainly depends on the biasing of its inputs and its internal nodes, static power consumption is correlated with the last value computed by this gate. An attacker can thus exploit the relation between static power consumption and a computed value to guess a secret value in a cryptographic computation [51].

Different practical techniques can be used to measure the power consumption of a chip, depending mainly on how the chip is encapsulated, and how it is supplied:

- if a small resistor is placed between the VDD (GND) of the circuit and the VDD (GND) of the power supply, the consumption of the circuit can be measured at the pins of the resistor;
- another way is to use a dedicated current probe [1].

3.3.2 Electromagnetic Leakage

EM radiations emitted by an integrated circuit are mainly due to the displacement of current through the rails of the metal layers. This phenomenon has been formalized by Maxwell's equations.

Practical experiments in [47] showed that the power and ground (P/G) network represent the majority of EM radiations of a CMOS device. Thus, when transistors are switching between on and off states, current crosses the P/G network to supply the pins of the switching transistors, and this displacement of current creates variations in the EM field induced by the electrical behavior of the circuit.

Practically speaking, either the electric or the magnetic field can be measured. But for EM based SCA, measuring the magnetic field generally gives better results. The magnetic field emitted by a chip is measured using a near field magnetic sensor [48].

3.3.3 Hamming Weight vs. Hamming Distance Models

In Kocher's original paper [36], the leakage model used is based on the Hamming Weight of the bit that the attacker tries to guess. This model, called the *Hamming Weight leakage model*, considers that a 0 does not lead to excess of power consumption (or EM radiations), whereas a 1 involves a significant amount of power consumption. Thus:

- transitions $0 \rightarrow 0$ and $1 \rightarrow 0$ are considered as not leading to excess power consumption;
- transitions $0 \rightarrow 1$ and $1 \rightarrow 1$ are assumed involve an excess of power consumption.

Actually this model does not exactly match with the reality, except in precharged logics (where each register is precharged at 0 before being updated, in this case the Hamming Distance of a such word is equal to its Hamming Weight).

So the *Hamming Weight (HW) leakage model* could be improved by considering the switching state rather than the output state of the word concerned [18] (the Hamming Distance between the previous and the new state):

- transitions $0 \rightarrow 0$ and $1 \rightarrow 1$ do not lead to excess power consumption;
- transitions $0 \rightarrow 1$ and $1 \rightarrow 0$ involve excess power consumption.

This leakage model is called the *Hamming Distance (HD) leakage model*. Figure 3.6 summarizes how these two leakage models rank transitions of one bit, considering its four possible transitions.

Other leakage models are possible, for instance a leakage model that distinguishes rising transitions and falling transitions has been studied in [45, 48], but the results obtained were similar to those obtained with the HD model.

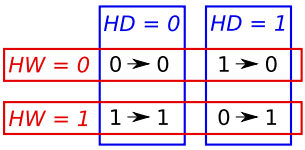

Fig. 3.6 Ranking of a transition of one bit following the Hamming Weight and Hamming Distance leakage models

3.4 Side-Channel Attacks

As shown previously, both the power consumption and the EM radiations of a circuit are tightly correlated with the data it processes. In this context, different cryptanalytic methods have been proposed to exploit this behavior, and to guess secrets involved in a cryptographic operation. The four main methods that exploit power or EM leakage presented here are respectively SPA/SEMA, DPA/DEMA, Template attacks, and Information theoretic attacks.

Note that, in the rest of the chapter, we do not distinguish between power consumption and EM radiations of cryptographic operations. Our explanations focus on the algorithmic treatment of physical leakages, mainly considering power based attacks. Indeed, up to now most published works are related to the power side channel but the algorithmic treatment is the same as for the EM side channel.

3.4.1 SPA/SEMA

The Simple Power Analysis (SPA), originally described in [36], exploits a single power consumption trace of a cryptographic operation to guess the secret used in the computation (SEMA stands for Simple ElectroMagnetic Analysis, in the case of an EM trace of a cryptographic operation).

Actually, this method works directly when applied on some asymmetric cryptography operations. For instance, in the case of an RSA decryption, the computation of the modular exponentiation:

$$m = c^d \pmod{n} \tag{3.1}$$

with m being the plaintext, c the ciphertext, d the private key exponent and n the modulus, can be naively calculated using the right-to-left binary method (d_i is the ith bit of d), as described in Algorithm 1.

Depending the value of each bit of the private key exponent d, one computes only a squaring operation or both a multiplication and a squaring operation. Hence the power consumption (or EM radiations) of each elements of the *for* loop will be different depending on the value of the bit of the private key exponent d, and by simply analyzing one trace, an attacker can easily guess bit values of the private key d. Figure 3.7 shows a typical trace (after demodulation at the FPGA clock frequency [44]).

A similar attack is possible on ECC, but rather than attacking the square and multiply algorithm, one can attack the double-and-add algorithm involved in the point multiplication.

Algorithm 1 Right-to-left binary method for modular exponentiation

Inputs: c, $d = (d_{l-1}d_{l-2}\dots d_0)_2$, n
Output: m
$m = 1$
for $i = 0$ to $l - 1$ **do**
  if $d_i = 1$ **then**
    $m = m * c \pmod{n}$
  end if
  $c = (c * c) \pmod{n}$
end for
Result: m

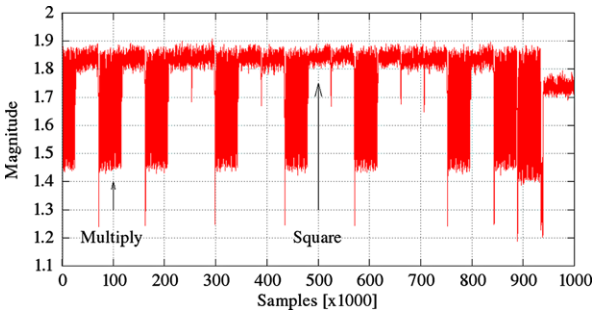

Fig. 3.7 Example of an SEMA on RSA

Concerning symmetric key cryptography, SPA attacks focusing on specific implementations of the key schedule have been published. [42] reports an SPA against the AES Key Expansion, and [5] describes an SPA against the key schedule of the Camellia block cipher.

3.4.2 DPA/DEMA

The Differential Power Analysis (DPA), originally described in the Kocher's pioneering paper [36], is a powerful technique that allows the attacker to guess secret keys used in a lot of cryptographic primitives. A lot of authors have proposed improvements of this attack along with countermeasures. When the attacker uses the EM radiations of the chip rather than its power consumption, DPA becomes DEMA, for Differential Electromagnetic Analysis.

3.4.2.1 Original Algorithm

DPA is a known plaintext or known ciphertext attack. The adversary enciphers (resp. deciphers) N PlainText Inputs (*PTI*) (resp. N CipherText Outputs, *CTO*) with an unknown key stored in the device, and monitors the power consumption of the device

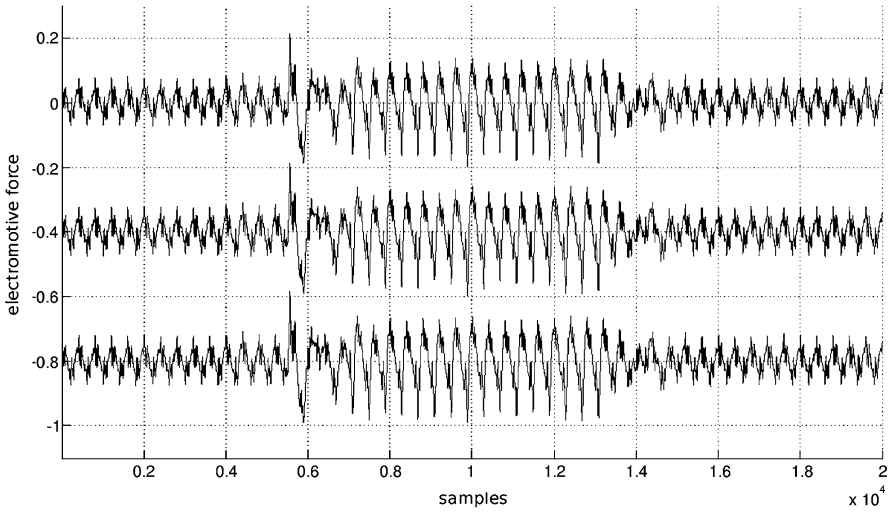

Fig. 3.8 Electromagnetic radiation traces of three different DES encryptions monitored on an FPGA

during each ciphering (resp. deciphering) operation. At the end of the first stage, he obtains N *PTIs* (resp. N *CTOs*) and N power consumption traces. Each trace is the change versus time in the power consumption of the chip.

Note that these traces have to be well aligned; this means that the time index of the beginning of ciphering has to be the same for all measurements.

If the measurements are not well aligned (due for instance to countermeasures like random clock frequency or dummy instructions), different preprocessing techniques enables traces to be re-synchronized [4, 6, 7, 21, 33, 53].

Figure 3.8 shows several EM radiation traces corresponding to the DES encryptions of different *PTIs* with the same key monitored on an FPGA.

The second stage is statistical processing of the N *PTIs* (resp. N *CTOs*) with the N traces.

In the rest of the chapter, the DES [2] is used as example, because it is the best most known block cipher and the principles that apply to the DPA also apply to other cryptographic ciphers, such as AES [3]. For the sake of convenience, we consider that the adversary is using a known plaintext attack and is trying to guess the round-key 1 of the DES (the remaining 8 bits could then be discovered through a brute force attack). A similar algorithm allows the round-key 16 in a known ciphertext attack to be discovered.

Since the set of all possible values for the round-key 1 is too large to test all of them, the adversary usually divides the round-key 1 into 8 parts of 6 bits each (here called sub-keys) and attacks each sub-key independently and sequentially. Thus, for each sub-key, there are 64 possible values.

Moreover, attacking each sub-key independently allows all kind of implementations to be targeted. For instance, in software implementations, depending on the size of the data bus (generally 8, 16 or 32), sbox processing can be computed se-

quentially, and so sbox 1 is not processed at the same time index as sbox 8. Unlike in some hardware implementations, all the sboxes are processed at the same time, meaning other approaches are possible, as explained below.

The adversary makes hypothesizes on the 6 bits of the attacked sub-key, and for each *PTI*, he computes the output (4 bits) of the corresponding sbox. This value is called the Intermediate Value (*IV*). In the state-of-the-art mono-bit DPA [36], the adversary targets out of the four, for instance the Less Significant Bit (LSB).

If the LSB of IV_1 (corresponding to the first plaintext, PTI_1) is equal to 0, the associated trace (T_1) is ranked in set A. If the LSB of IV_1 is equal to 1, T_1 is ranked in set B. As explained above, the adversary ranks all the traces, in sets A or B, and then computes the difference in the means of sets A and B. The resulting curve is called a differential curve, and corresponds to a sub-key hypothesis. The adversary computes the 64 differential curves corresponding to the 64 possible values for a given sub-key hypothesis.

As explained in [36], the differential curve, denoted Δ, for a sub-key hypothesis K_s, is calculated as follows:

$$\Delta_{K_s}[j] = \frac{\sum_{i=1}^{N} D(PTI_i, K_s) T_i[j]}{\sum_{i=1}^{N} D(PTI_i, K_s)} - \frac{\sum_{i=1}^{N} (1 - D(PTI_i, K_s)) T_i[j]}{\sum_{i=1}^{N} (1 - D(PTI_i, K_s))}, \tag{3.2}$$

where $\Delta_{K_s}[j]$ is the jth sample of the differential curve, N is the number of traces used, PTI_i is the ith plaintext, $T_i[j]$ is the jth sample of the trace and D the decision function than ranks traces in set A or B, also called *selection function*.

If the sub-key hypothesis is wrong, all the computed intermediate values will be wrong with respect to the data really processed on the chip. In this case, the traces will be randomly classified in sets A and B. The mean curves of sets A and B will be similar, and the differential curve will look like a thick horizontal line (mainly composed of noise).

On the other hand, if the sub-key hypothesis is correct, all the computed intermediate values will match the data really processed on the chip, and traces in set A will have the same characteristic: at the time index where the intermediate value is computed, the LSB of *IV* equal to 0 will not lead to excess power consumption. Conversely, when the LSB of *IV* is equal to 1, a bit more energy will be consumed at the same time index and spikes corresponding to the clock cycle where *IV* is computed will be greater on traces in set B than on traces in set A. When the difference in the means of the 2 sets is being computed, a spike will appear at the time index of the differential curve concerned indicating that the sub-key hypothesis is correct.

Figure 3.9 shows the 64 differential curves computed following guesses of sub-key 1 of round-key 1 of the DES, using 500 EM traces (each one averaged 20 times). Differential curves corresponding to wrong guesses of the sub-key are in cyan, whereas the curve corresponding to the correct guess of the sub-key is in black, and has the greatest peak. To guess the eight parts of round-key 1, processing is applied sequentially on each sub-key.

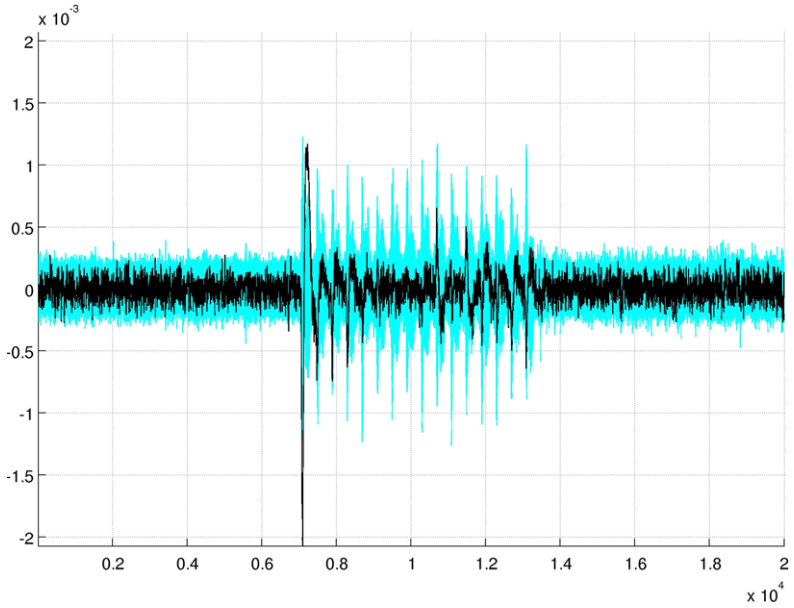

Fig. 3.9 Example of a successful DEMA: 64 differential curves computed following hypotheses of sub-key 1 of round-key 1 of the DES, using 500 EM traces (each one averaged 20 times). Differential curves corresponding to wrong hypotheses of the sub-key are in *cyan*, the curve corresponding to the correct hypothesis of the sub-key is in *black* (color online)

3.4.2.2 Improvements of DPA

Hamming Weight vs. Hamming Distance Leakage Models In the original DPA algorithm, and as explained in the previous section, the selection function follows the Hamming Weight (HW) power consumption model. But previously, we explained that the Hamming Distance (HD) power consumption model matches reality better.

Moreover, an iterative implementation is used in most hardware implementations of block ciphers. Thus, only one generic round is implemented, and at each clock cycle the intermediate result of the previous round is used as input with the current round-key generated by the key schedule.

Hence, rather than considering an output bit of an sbox, we can consider the bit linked to the previous one in R1 [2]. Thus, as the adversary knows the value of this bit in R0 (because he/she knows the PTI), and assuming that R0 and R1 are stored in the same register which is updated at each round, we can compute the Hamming Distance between the value of the targeted bit in R0 and R1. In this case, it is generally not the power consumption of an inverter that is estimated, but the power consumption of a flip-flop.

Multi-bit DPA Another way to improve DPA attacks consists in considering the four bits linked to the output of the sbox rather than only one. As proposed in [14],

we can compute the differential curve for each bit out of the four from the output of the sbox, and sum the four differential curves.

We can also rank traces using another criterion [43]: by summing the number of switching bits out of the four considered, and if the sum is smaller than 2, we rank the associated trace in set A, whereas if the sum is greater than 2, we rank the trace in set B.

Distinguishers In the original DPA algorithm, the *distinguisher* used to correlate predictions with measurements is called *Difference of Mean* (DoM). Different works have proposed other methods, based on different mathematical tools. Here, we describe the best known.

Partition Distinguishers As explained previously, the *Difference of Means* (DoM) distinguisher consists in ranking, according to a key hypothesis, all the traces in two sets, following a criterion. Then the difference in the means of the two sets of traces is computed. Applying this method to the different key hypotheses reveals the correct key hypothesis.

In the previous paragraph, we described different improvements of Kocher's attack, like the multi-bit DPA proposed by Messerges [43], or the method suggested by Bevan [14].

Another proposal, presented in [37], consists in ranking traces in more than two sets. In the case of attacking a DES, if the attacker focuses on the four output bits of an sbox, he/she can rank traces in five sets, one per Hamming Weight. Considering four bits b_0, b_1, b_2 and b_3, the Hamming Weight of the word $b_3b_2b_1b_0$ is between 0 and 4, and leads to five possible values. Then the attacker has to choose coefficients a_i, $i = 0, \ldots, 3$, assigned to each set. This method has been called Partitioning Power Analysis (PPA).

Unfortunately, no efficient method has been proposed to determine these coefficients in the general context. Indeed, due to the process variations, foundries cannot ensure that all the bus lines have *exactly* the same geometrical dimensions. For example, bit b_0 will not consume exactly the same amount of power (or radiate exactly the same field) as bit b_1. This is all the more true in the presence of countermeasures. For instance, in a circuit where the nets are balanced (for instance thanks to a dual-rail logic [68]), the coefficients a_i are given in Fig. 3.10 [58].

Blind techniques enable identification of the coefficients. The optimal coefficients in the context of masking are derived in [50]. In a general context, the stochastic approach, discussed in Sect. 3.4.3.2, always applies.

Comparison Distinguishers In a differential SCA, the attacker wants to estimate the relation between predictions, depending on a key hypothesis, and on measurements. In [18], the authors suggest using a well-known statistical tool, *Pearson's correlation coefficient*. Let X and Y be two random variables, $\mathrm{cov}(X, Y)$ be the covariance between X and Y, and σ_X and σ_Y be the standard deviations of X and Y, *Pearson's correlation coefficient* is computed as follows:

$$\rho(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \times \sigma_Y}. \tag{3.3}$$

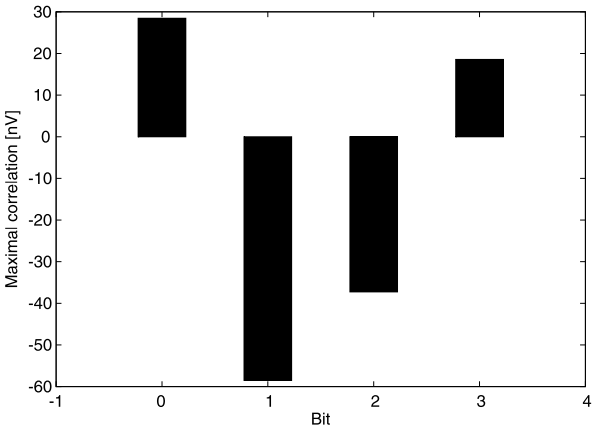

Fig. 3.10 Optimal weights found for the sbox output of DES in dual rail with precharge logic

This mathematical tool measures the dependence between two quantities, and for SCA, its application is straightforward. Let N be the number of traces, K_s the key hypothesis, PTI_i the ith PlainText Input, $H(K_s, PTI_i)$ the number of switching bits according to the key hypothesis K_s and the plaintext PTI_i, $T_i(j)$ the jth sample of the ith trace, Eq. (3.3) becomes:

$$\rho_{K_s}[j] = \frac{N.\sum_{i=1}^{N} H(K_s, PTI_i).T_i(j) -}{\sqrt{N.\sum_{i=1}^{N} H(K_s, PTI_i)^2 - (\sum_{i=1}^{N} H(K_s, PTI_i))^2}} \times \frac{\sum_{i=1}^{N} H(K_s, PTI_i).\sum_{i=1}^{N} T_i(j)}{\sqrt{N.\sum_{i=1}^{N} T_i(j)^2 - (\sum_{i=1}^{N} T_i(j))^2}}. \quad (3.4)$$

The variant of the DPA using the *Pearson's correlation* has been called Correlation Power Analysis (CPA), and was first introduced in [18].

However, *Pearson's correlation* measures only linear dependencies between two quantities. Other tools, like Spearman's rank correlation, or Kendall's tau rank correlation, can measure linear dependencies between two random variables, but practical experiments have shown that these tools give very similar results [9].

Another author [28] introduced a different statistical tool, mutual information. Mutual information allows both linear and non-linear dependencies between two random variables to be measured. This is called Mutual Information Analysis (MIA). This technique is discussed in Sect. 3.4.4.1.

Key Search Generally, when the attacker has computed the 64 differential curves corresponding to guesses of the sub-key S_k using N traces, he/she searches for the maximum (in absolute) of each differential curve. Among the 64 maxima, the biggest one reveals the guessed during the attack using N traces. A classical optimization for the key search step consists in reducing the proportion of the differential curve inspected to detect the maximum. The key search may concentrate on a part of the curve corresponding to the attacked round.

3.4.3 Template Attack

The Template Attack (TA) is considered to be the most powerful type of SCA. The scenario of such an attack is slightly different than that required for other kinds of SCA. First, to build *templates* for the different possible values of the secret, attackers have to be in possession of a circuit identical to the circuit under attack. Then, they measure one trace on the attacked device, and using appropriate methods, compare this trace with the different templates they built to guess the value of the secret.

3.4.3.1 Original Template Attack

This attack, firstly described in [20], happens in two stages, the template building stage, and the template matching stage.

Template Building Phase In the first stage, attackers use the circuit under their full control, which is identical to the circuit under attack, to build templates for each pair of data and key.

Power traces can be characterized by a multivariate normal distribution, which is fully defined by a mean vector M and a covariance matrix C. In the rest of the paragraph, the pair (M, C) is referred to as a *template*. The attacker builds a *template* $(M, C)^{d_i,k_j}$ for each pair of data and key (d_i, k_j) following these steps:

- for each pair of data and key (d_i, k_j), acquire p traces $T_1^{d_i,k_j}, \ldots, T_p^{d_i,k_j}$;
- compute a mean vector M^{d_i,k_j} for each pair (d_i, k_j) from the p traces as follows:

$$M^{d_i,k_j} = \frac{1}{p} \sum_{l=1}^{p} T_l^{d_i,k_j}; \tag{3.5}$$

- (optional) compute pairwise differences between the mean vectors M^{d_i,k_j} in order to identify and select only points $P_1, \ldots, P_n$ at which large differences show up. This optional step significantly reduces the processing overhead with only a small loss of accuracy;
- for each pair d_i, k_j and for each acquired trace $T_l^{d_i,k_j}$, $l = 1, \ldots, p$, compute the noise vector $N_l^{d_i,k_j}$, $l = 1, \ldots, p$ as follows:

$$N_l^{d_i,k_j} = (T_l^{d_i,k_j}(P_1) - M^{d_i,k_j}(P_1) \ldots T_l^{d_i,k_j}(P_n) - M^{d_i,k_j}(P_n)); \tag{3.6}$$

- for each pair d_i, k_j, compute the noise covariance matrix C^{d_i,k_j} between all pairs of components of the noise vectors $N_l^{d_i,k_j} s$, $l = 1, \ldots, p$, for all the p traces, as follows:

$$C^{d_i,k_j}[u, v] = \mathrm{cov}(N_l(P_u), N_l(P_v)). \tag{3.7}$$

Thus, for each data and key pair (d_i, k_j), the attacker has built a *template*, that characterizes the deterministic part of the power trace, but also its noise part, modeling it using a multivariate Gaussian model.

Template Matching Phase In the second stage, the attackers use a power trace from the device under attack to determine the key. To this end, they evaluate the probability density function of the multivariate normal distribution with $(M, C)^{d_i,k_j}$ and the trace of the device under attack.

In other words, given a power trace t of the chip under attack, and a *template* $(M, C)^{d_i,k_j}$, the attacker computes the probability:

$$p(t; (M, C)^{d_i,k_j}) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(C)}} e^{-\frac{1}{2}(t-M)^T \cdot C^{-1} \cdot (t-M)}. \tag{3.8}$$

The attacker does this for every template. As a result, he/she obtains the probabilities $p(t; (M, C)^{d_1,k_1}), \ldots, p(t; (M, C)^{d_D,k_K})$. The probabilities measure how well the templates fit to a given trace. Intuitively, the highest probability should indicate the correct template. Because each template is associated with a key, the attacker also gets a clue to the correct key. This intuition is also supported by the maximum-likelihood decision rule.

The points of interest can be either selected heuristically [20], by *ad hoc* techniques [27], or by dimensionality reduction tools, such as principal components analysis [11].

3.4.3.2 Stochastic Method

An improvement of TA is suggested in [59, 60]. It consists in using a stochastic method to approximate the real leakage function within a suitable vector subspace. The attack requires some engineering skills to introduce a relevant parametric model. Then, the on-line attack basically consists in simultaneously:

- estimating the best parameters, using a linear regression, and
- deciding which model is the closest to the side channel measurements, using the minimal Euclidean distance as a distinguisher.

An empirical comparison of template and stochastic attacks is provided in [65].

3.4.4 Information Theoretic Attacks

3.4.4.1 MIA—Mutual Information Analysis

At CHES 2008, a generic side channel distinguisher, MIA, was proposed in [29]. It is an attractive alternative to the above-mentioned attacks since some assumptions about the adversary can be disregarded. In particular, it does not require a linear dependency between the leakage and the predicted data, as is the case for DPA and CPA, and so it is not only able to exploit any kind of dependency but does not need to profile the leakage as it is the case for template attacks [20].

The rationale of MIA is to compare distributions of observations with random distributions rather observations with a model. This is why the MIA measures the Kullback-Leibler divergence between observations knowing the correct sub-key and observations not knowing anything about the internal values [70].

3.4.4.2 EPA—Entropy Power Analysis

Mutual information analysis has been tested in noisy real world designs. It indeed appears to be a powerful approach to break unprotected implementations. However, the MIA fails when applied on a DES cryptoprocessor with masked substitution boxes (sboxes) in ROM [66]. However, this masking implementation remains sensitive to Higher Order Differential Power Analysis (HO-DPA). For instance, an attack based on variance analysis clearly reveals the vulnerabilities of a first-order masking countermeasure. A novel approach to information theoretic HO attacks, called Entropy-based Power Analysis (EPA), has therefore been proposed. This new attack gives greater importance to highly informative partitions and distinguishes the key hypotheses better [41].

3.4.4.3 VPA—Variance Power Analysis

Information theoretic attacks are extremely powerful, as they exploit any deviation from a random probability density function (PDF). However, estimating PDFs is a hard task [49, 70], since for the PDFs to be accurate, it ideally requires a lot of measurements. To simplify the attack, articles [38, 40, 64] suggest limiting the analysis of the PDFs to their second cumulant. The estimation is much faster and the distinguisher thus gains strength. This is referred to as variance power analysis (VPA). This metric to compute distances between PDFs is extremely useful against implementations protected by first-order masking.

3.5 Conclusions

More than ten years after the first publication of Kocher's attack, a lot of improvements have been proposed for power and EM based Side Channel Attacks. Firstly, it has been shown that both power consumption and EM radiations of a circuit can leak sensitive information. Secondly, different methods have been proposed to extract a secret key used in a cryptographic operation from physical leakages emitted by the device. In this chapter we described SPA/SEMA, DPA/DEMA and Template Attacks. We also described different improvements of differential SCA that give better results than the original attack, especially when the attacker has some knowledge of the implementation. To thwart these attacks, a lot of methods have been proposed. Due to the specificities of FPGAs, some of these countermeasures are not applicable, while others are relatively easy to integrate. The next chapter describes the main methods used to protect cryptographic algorithms implemented in FPGAs against Side Channel Attacks.

References

1. Tektronix Current Probes Ct1, Ct2, Ct6. http://www.tek.com

2. Data Encryption Standard: FIPS PUB 46-3 (1999)
3. Advanced Encryption Standard: FIPS PUB 197 (2001)
4. A method for resynchronizing a random clock on smartcards. In: Eurosmart (2001)
5. A simple power analysis attack against the key schedule of the camellia block cipher. Inf. Process. Lett. **95**(3), 409–412 (2005)
6. Improving the DPA attack using wavelet transform. In: NIST Physical Security Testing Workshop (2005)
7. High-resolution side-channel attack using phase-based waveform matching. In: CHES, pp. 187–200 (2006)
8. Diode Laser Station. Riscure (2009)
9. DPA contest 2008/2009. http://www.dpacontest.org (2009)
10. Anderson, R., Kuhn, M.: Low cost attacks on tamper resistant devices. In: Proceedings of the 5th International Workshop on Security Protocols, pp. 125–136 (1998)
11. Archambeau, C., Peeters, É., Standaert, F.-X., Quisquater, J.-J.: Template attacks in principal subspaces. In: CHES, Yokohama, Japan, October 10–13. LNCS, vol. 4249, pp. 1–14. Springer, Berlin (2006)
12. Bar-el, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The sorcerer's apprentice guide to fault attacks (2004)
13. Di-Battista, J., Courrège, J.-C., Rouzeyre, B., Torres, L., Perdu, P.: When failure analysis meets side-channel attacks. In: CHES, pp. 188–202 (2010). doi:10.1007/978-3-642-15031-9_13
14. Bevan, R., Knudsen, E.: Ways to enhance differential power analysis. In: ICISC, pp. 327–342 (2002)
15. Bhasin, S., Selmane, N., Guilley, S., Danger, J.-L.: Security evaluation of different AES implementations against practical setup time violation attacks in FPGAs. In: HOST (Hardware Oriented Security and Trust), July 27th, pp. 15–21. IEEE Comput. Soc., Los Alamitos (2009). doi:10.1109/HST.2009.5225057. In conjunction with DAC-2009, Moscone Center, San Francisco, CA, USA
16. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: CRYPTO, pp. 513–525 (1997)
17. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: EUROCRYPT, pp. 37–51 (1997)
18. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: CHES, pp. 16–29 (2004)
19. Brumley, D., Boneh, D.: Remote timing attacks are practical. In: SSYM'03: Proceedings of the 12th Conference on USENIX Security Symposium, pp. 1–1. USENIX Association, Berkeley (2003)
20. Chari, S., Rao, J., Rohatgi, P.: Template attacks. In: CHES, pp. 13–28 (2002)
21. Clavier, C., Coron, J.-S., Dabbous, N.: Differential power analysis in the presence of hardware countermeasures. In: CHES, pp. 252–263 (2000)
22. Coron, J.-S., Naccache, D., Kocher, P.: Statistics and secret leakage. ACM Trans. Embed. Comput. Syst. **3**(3), 492–508 (2004)
23. Dehbaoui, A., Lomne, V., Maurine, P., Torres, L.: Magnitude squared incoherence EM analysis for integrated cryptographic module localisation. Electron. Lett. **45**(15), 778–780 (2009). doi:10.1049/el.2009.0342
24. Dinur, I., Shamir, A.: Generic analysis of small cryptographic leaks. In: FDTC, Santa Barbara, CA, USA, August 21, pp. 51–65. IEEE Comput. Soc., Los Alamitos (2010). doi:10.1109/FDTC.2010.11
25. Dinur, I., Shamir, A.: Side channel cube attacks on block ciphers. Cryptology ePrint Archive, Report 2009/127. http://eprint.iacr.org/ (March 2009)
26. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: concrete results. In: CHES, pp. 251–261 (2001)
27. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. stochastic methods. In: CHES, Yokohama, Japan, October 10–13. LNCS, vol. 4249, pp. 15–29. Springer, Berlin (2006)

28. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: CHES, pp. 426–442 (2008)
29. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F., Veyrat-Charvillon, N.: Mutual information analysis: a comprehesive study. J. Cryptol. **24**(2), pp. 269–291 (2010)
30. Giraud, C., Thiebeauld, H.: A survey on fault attacks. In: Smart Card Research and Advanced Applications VI, IFIP 18th, World Computer Congress, TC8/WG8.8 & TC11/WG11.2 Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS), Toulouse, France, 22–27 August, pp. 159–176. Kluwer, Dordrecht (2004)
31. Guilley, S., Hoogvorst, P., Pacalet, R.: Differential power analysis model and some results. In: Proceedings of WCC/CARDIS, Toulouse, France, August, pp. 127–142. Kluwer, Dordrecht (2004). doi:10.1007/1-4020-8147-2_9
32. Handschuh, H., Paillier, P., Stern, J.: Probing attacks on tamper-resistant devices. In: CHES, pp. 303–315 (1999)
33. Kafi, M., Guilley, S., Marcello, S., Naccache, D.: Deconvolving protected signals. In: ARES/CISIS, Fukuoka, Kyūshū, Japan, March 16th–19th, pp. 687–694. IEEE Comput. Soc., Los Alamitos (2009). doi:10.1109/ARES.2009.197
34. Khelil, F., Hamdi, M., Guilley, S., Danger, J.-L., Selmane, N.: Fault analysis attack on an FPGA AES implementation. In: NTMS, Tangier, Morocco, November, pp. 1–5. IEEE (2008). doi:10.1109/NTMS.2008.ECP.45
35. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, pp. 104–113. Springer, London (1996)
36. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: CRYPTO, pp. 388–397 (1999)
37. Le, T.-H., Clédiere, J., Canovas, C., Robisson, B., Serviere, C., Lacoume, J.-L.: A proposition for correlation power analysis enhancement. In: CHES, pp. 174–186 (2006)
38. Li, Y., Sakiyama, K., Batina, L., Nakatsu, D., Ohta, K.: Power variance analysis breaks a masked ASIC implementation of AES. In: DATE, Dresden, Germany, March 8–12, pp. 1059–1064. IEEE (2010)
39. Lin, L., Burleson, W.: Analysis and mitigation of process variation impacts on power-attack tolerance. In: DAC, pp. 238–243 (2009)
40. Maghrebi, H., Danger, J.-L., Flament, F., Guilley, S.: Evaluation of countermeasures implementation based on Boolean masking to thwart first and second order side-channel attacks. In: SCS, Jerba, Tunisia, pp. 1–6. IEEE (2009). Complete version online: http://hal.archives-ouvertes.fr/hal-00425523/en/. doi:10.1109/ICSCS.2009.5412597
41. Maghrebi, H., Guilley, S., Danger, J.-L., Flament, F.: Entropy-based power attack. In: HOST, Anaheim Convention Center, Anaheim, CA, USA, June 13–14, pp. 1–6. IEEE Comput. Soc., Los Alamitos (2010). doi:10.1109/HST.2010.5513124
42. Mangard, S.: A simple power-analysis (SPA) attack on implementations of the AES key expansion. In: ICISC, pp. 343–358 (2002)
43. Messerges, T., Dabbish, E., Sloan, R.: Investigations of power analysis attacks on smartcards. In: WOST, pp. 17–17 (1999)
44. Meynard, O., Rçal, D., Guilley, S., Danger, J.-L., Homma, N.: Enhancement of simple electromagnetic attacks by pre-characterization in frequency domain and demodulation techniques. In: DATE, Grenoble, France, March 14–18. IEEE Comput. Soc., Los Alamitos (2011)
45. Natale, G.D., Flottes, M.-L., Rouzeyre, B.: An integrated validation environment for differential power analysis. In: DELTA, pp. 527–532 (2008)
46. Nohl, K., Evans, D., Starbug, S., Plötz, H.: Reverse-engineering a cryptographic RFID tag. In: Proceedings of the 17th Conference on Security Symposium, pp. 185–193. USENIX Association, Berkeley (2008). http://portal.acm.org/citation.cfm?id=1496711.1496724
47. Ordas, T., Lisart, M., Sicard, E., Maurine, P., Torres, L.: Near-field mapping system to scan in time domain the magnetic emissions of integrated circuits. In: PATMOS, pp. 229–236 (2008)
48. Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Power and electromagnetic analysis: improved model, consequences and comparisons. Integration VLSI J. **40**, 52–60 (2007). doi:10.1016/j.vlsi.2005.12.013

49. Prouff, E., Rivain, M.: Theoretical and practical aspects of mutual information based side channel analysis. In: ACNS, Paris-Rocquencourt, France, June 2–5. LNCS, vol. 5536, pp. 499–518. Springer, Berlin (2009)
50. Prouff, E., Rivain, M., Bevan, R.: Statistical analysis of second order differential power analysis. IEEE Trans. Comput. **58**(6), 799–811 (2009)
51. Quisquater, J.-J., Standaert, F.-X.: Physically secure cryptographic computations: from micro to nano electronic devices. In: DSN, Workshop on Dependable and Secure Nanocomputing (WDSN), June 28. IEEE Comput. Soc., Edinburgh (2007). Invited Talk, 2 pages
52. Réal, D., Valette, F., Drissi, M.: Enhancing correlation electromagnetic attack using planar near-field cartography. In: DATE, Nice, France, April 20–24, pp. 628–633. IEEE (2009)
53. Réal, D., Canovas, C., Clédiere, J., Drissi, M., Valette, F.: Defeating classical hardware countermeasures: a new processing for side channel analysis. In: DATE, pp. 1274–1279 (2008)
54. Satoh, A.: Side-channel Attack Standard Evaluation Board, SASEBO. Project of the AIST—RCIS (Research Center for Information Security). http://www.rcis.aist.go.jp/special/SASEBO/
55. Sauvage, L.: Cartographie électromagnétique pour la cryptanalyse physique. PhD thesis, TELECOM-ParisTech, Paris, France (September 2009)
56. Sauvage, L., Guilley, S., Mathieu, Y.: Electromagnetic radiations of FPGAs: high spatial resolution cartography and attack of a cryptographic module. ACM Trans. Reconfigurable Technol. Syst. **2**(1), 1–24 (2009). Full text in http://hal.archives-ouvertes.fr/hal-00319164/en/. doi:10.1145/1502781.1502785
57. Sauvage, L., Guilley, S., Flament, F., Danger, J.-L., Mathieu, Y.: Cross-correlation cartography. In: ReConFig, Cancún, Quintana Roo, México, December 13–15, pp. 268–273. IEEE Comput. Soc., Los Alamitos (2010). doi:10.1109/ReConFig.2010.75
58. Sauvage, L., Nassar, M., Guilley, S., Flament, F., Danger, J.-L., Mathieu, Y.: Exploiting dual-output programmable blocks to balance secure dual-rail logics. Int. J. Reconfigurable Comput. **2010**, 375245 (2010). 12 pages. doi:10.1155/2010/375245
59. Schindler, W.: Advanced stochastic methods in side channel analysis on block ciphers in the presence of masking. J. Math. Cryptol. **2**(3), 291–310 (2008). ISSN (Online) 1862-2984, ISSN (Print) 1862-2976. doi:10.1515/JMC.2008.013
60. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: CHES, pp. 30–46 (2005)
61. Schmidt, J.-M., Hutter, M.: Optical and EM fault-attacks on CRT-based RSA: concrete results. In: Austrochip (2007)
62. Schmidt, J.-M., Kim, C.H.: A probing attack on AES, pp. 256–265 (2009)
63. Skorobogatov, S., Anderson, R.: Optical fault induction attacks. In: CHES, pp. 2–12 (2002)
64. Standaert, F.-X., Gierlichs, B., Verbauwhede, I.: Partition vs. comparison side-channel distinguishers: an empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. In: ICISC, Seoul, Korea, December 3–5. LNCS, vol. 5461, pp. 253–267. Springer, Berlin (2008)
65. Standaert, F.-X., Koeune, F., Schindler, W.: How to compare profiled side-channel attacks? In: ACNS, Paris-Rocquencourt, France, June 2–5. LNCS, vol. 5536, pp. 485–498. Springer, Berlin (2009)
66. Standaert, F.-X., Rouvroy, G., Quisquater, J.-J.: FPGA implementations of the DES and triple-DES masked against power analysis attacks. In: Proceedings of FPL 2006, Madrid, Spain, August. IEEE (2006)
67. Standaert, F.-X., Batina, L., Mulder, E.D., Lemke, K., Mentens, N., Oswald, E., Peeters, E.: Report on DPA and EMA Attacks on FPGAs. July 31 ECRYPT IST-2002-507932, "European Network of Excellence in Cryptography". Deliverable D.VAM.5. http://www.ecrypt.eu.org/ecrypt1/documents/D.VAM.5-1.pdf
68. Tiri, K., Verbauwhede, I.: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: DATE'04, Paris, France, February, pp. 246–251. IEEE Comput. Soc., Los Alamitos (2004). doi:10.1109/DATE.2004.1268856

69. Torrance, R., James, D.: The state-of-the-art in IC reverse engineering. In: CHES, pp. 363–381 (2009)
70. Veyrat-Charvillon, N., Standaert, F.-X.: Mutual information analysis: how, when and why? In: CHES, Lausanne, Switzerland, September 6–9. LNCS, vol. 5747, pp. 429–443. Springer, Berlin (2009)