Introduction to Side-Channel Attacks

François-Xavier Standaert[*]

UCL Crypto Group, Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
e-mail: fstandae@uclouvain.be

Abstract. Side-channel cryptanalysis is a new research area in applied cryptography that has gained more and more interest since the mid-nineties. It considers adversaries trying to take advantage of the physical specificities of actual cryptographic devices. These implementation-specific attacks frequently turn out to be much more efficient than the best known cryptanalytic attacks against the underlying primitive seen as an idealized object. This paper aims to introduce such attacks with illustrative examples and to put forward a number of practical concerns related to their implementation and countermeasures.

1 Introduction

A cryptographic primitive can be considered from two points of view: on the one hand, it can be viewed as an abstract mathematical object or black box (*i.e.* a transformation, possibly parameterized by a key, turning some input into some output); on the other hand, this primitive will *in fine* have to be implemented in a program that will run on a given processor, in a given environment, and will therefore present specific characteristics. The first point of view is the one of classical cryptanalysis; the second one is the one of physical security. Physical attacks on cryptographic devices take advantage of implementation-specific characteristics to recover the secret parameters involved in the computation. They are therefore much less general - since specific to a given implementation - but often much more powerful than classical cryptanalysis, and are considered very seriously by cryptographic devices manufacturers.

Such physical attacks are numerous and can be classified in many ways. The literature usually sorts them among two orthogonal axes:

1. Invasive *vs.* non-invasive: invasive attacks require depackaging the chip to get direct access to its inside components; a typical example of this is the connection of a wire on a data bus to see the data transfers. A non-invasive attack only exploits externally available information (the emission of which is however often unintentional) such as running time, power consumption, ...
2. Active *vs.* passive: active attacks try to tamper with the devices proper functioning; for example, fault-induction attacks will try to induce errors in the computation. As opposed, passive attacks will simply observe the devices behavior during their processing, without disturbing it.

[*] Postdoctoral researcher of the Belgian Fund for Scientific Research (FNRS).

The side-channel attacks we consider in this paper are a class of physical attacks in which an adversary tries to exploit physical information leakages such as timing information [10], power consumption [11] or electromagnetic radiation [1]. Since they are non-invasive, passive and they can generally be performed using relatively cheap equipment, they pose a serious threat to the security of most cryptographic hardware devices. Such devices range from personal computers to small embedded devices such as smart cards and RFIDs (Radio Frequency Identification Devices). Their proliferation in a continuously larger spectrum of applications has turned the physical security and side-channel issue into a real, practical concern that we aim to introduce in this paper.

For this purpose, we start by covering the basics of side-channel attacks. We discuss the origin of unintended leakages in recent microelectronic technologies and describe how simple measurement setups can be used to recover and exploit these physical features. Then, we introduce some classical attacks: Simple Power Analysis (SPA) and Differential Power Analysis (DPA). In the second part of the paper, we put forward the different steps of an actual side-channel attack through two illustrative examples. We take advantage of these examples to stress a number of practical concerns regarding the implementation of side-channel attacks and discuss their possible improvements. Finally, we list a number of countermeasures to reduce the impact of physical information leakages.

2 Basics of side-channel attacks

2.1 Origin of the leakages

Side-channel attacks are closely related to the existence of physically observable phenomenons caused by the execution of computing tasks in present microelectronic devices. For example, microprocessors consume time and power to perform their assigned tasks. They also radiate an electromagnetic field, dissipate heat and even make some noise [24]. As a matter of fact, there are plenty of information sources leaking from actual computers that can consequently be exploited by malicious adversaries. In this paper, we focus on power consumption and electromagnetic radiation that are two frequently considered side-channels in practical attacks. Since a large part of present digital circuits is based on CMOS gates, this introduction also only focuses on this technology. As will be mentioned in Section 4, other types of logic circuits could be considered for side-channel attacks, sometimes providing improved resistance compared with standard CMOS.

Power consumption in CMOS devices. Static CMOS gates have three distinct dissipation sources [21]. The first one is due to the leakage currents in transistors. The second one is due to the so-called “short-circuit currents”: there exists a short period during the switching of a gate while NMOS and PMOS are conducting simultaneously. Finally, the dynamic power consumption is due to the charge and discharge of the load capacitance C_L represented by the dotted paths in Figure 1. The respective importance of these dissipation sources

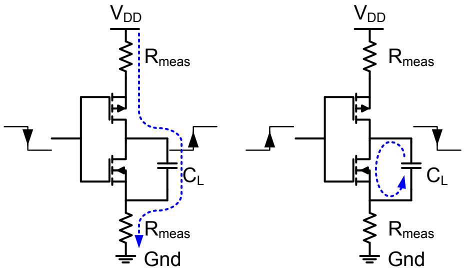


Fig. 1: Charge vs. discharge of a CMOS inverter.

typically depends on technology scalings. But the dynamic power consumption is particularly relevant from a side-channel point of view since it determines a simple relationship between a device's internal data and its externally observable power consumption. It can be written as:

$$P_{dyn} = C_L V_{DD}^2 P_{0\rightarrow 1} f, \tag{1}$$

where $P_{0\rightarrow 1} f$ is called the *switching activity*, $P_{0\rightarrow 1}$ is the probability of a $0 \rightarrow 1$ transition, f is the work frequency of the device and V_{DD} is the voltage of the power supply. In CMOS devices, when measuring the power consumption (either at the ground pin or at the power pin), the highest peak will appear during the charge of the capacitance (*i.e.* $0 \rightarrow 1$ event). During the discharge, the only current we can measure is the short-circuit path current. This data-dependent power consumption is the origin of side-channel information leakages.

EM radiation in CMOS devices. Just as the power consumption of CMOS devices is data-dependent, it can be showed that its electromagnetic radiation also is. From a theoretical point of view, electromagnetic leakages are usually explained from the Biot-Savart law:

$$d\mathbf{B} = \frac{\mu I d\mathbf{l} \times \widehat{r}}{4\pi r^2}, \tag{2}$$

where μ is the magnetic permeability, I is the current carried on a conductor of infinitesimal length $d\mathbf{l}$, $\widehat{r}$ is the unit vector specifying the distance between the current element and the field point and r is the distance from the current element to the field point. Although such a simple equation does not describe the exact (complex) radiation of an integrated circuit, it already emphasizes two important facts: (1) the field is data-dependent, due to the dependence on the current intensity and (2) the field orientation depends on the current direction. This data-dependent radiation is again the origin of side-channel information leakages. In general, any physically observable phenomenon that can be related to the internal configuration or activity of a cryptographic device can be a source of useful information to a malicious adversary.

Leakage models. From the previous physical facts, side-channel adversaries have derived a number of (more or less sophisticated) leakage models. They can be used both to simulate the attacks or to improve an attack's efficiency. For example, the *Hamming distance model* assumes that, when a value x_0 contained in a CMOS device switches into a value x_1, the actual side-channel leakages are correlated with the Hamming distance of these values, namely $H_D(x_0, x_1) = H_W(x_0 \oplus x_1)$. The *Hamming weigh model* is even simpler and assumes that, when a value x_0 is computed in a device, the actual side-channel leakages are correlated with the Hamming weight of this value, namely $H_W(x_0)$. As will be emphasized in Section 4, good leakage models have a strong impact on the efficiency of a side-channel attack. Hamming weight and distance models assume both that there are no differences between $0 \rightarrow 1$ and $1 \rightarrow 0$ events and that every bit in an implementation contributes identically to the overall power consumption. Improved models relax these assumptions, *e.g.* by considering different leakages for the $0 \rightarrow 1$ and $1 \rightarrow 0$ events [20], assigning different weights to the leakage contributions of an implementation's different parts [25] of by considering advanced statistical tools to characterize a device's leakage [6].

2.2 Measurement setups

As far as the practical implementation of a side-channel attack is concerned, the building of a good measurement setup is of primary importance. They aim to convert the physical features of an observable device into digitally exploitable data. Such setups are generally made of the following elements [13]:

- A target cryptographic device, *e.g.* a smart card, FPGA of integrated circuit running some cryptographic primitive, *e.g.* a block cipher.
- If not embedded on-chip, an external power supply, clock generator and any additional circuitry required for the device to run properly.
- A leakage probe. For example, power consumption can be monitored by inserting a small resistor within a the supply chain of the target device. Electromagnetic radiation can be captured with simple hand made coils.
- An acquisition device, *e.g.* digital oscilloscope with sufficient features (typically, 1 GS/s, 8 bits of resolution, ...), connected to a computer for the statistical analysis of the side-channel traces.

Just as leakage models, measurement setups have a strong influence on the efficiency of side-channel attacks. The quality of a measurement setup is mainly quantified by the amount of noise in its traces. Noise is a central issue in side-channel attacks and more generally in any signal processing application. In our specific context, various types of noise are usually considered, including physical noise (*i.e.* produced by the transistors and their environment), measurement noise (*i.e.* caused by the sampling process and tools), model matching noise (*i.e.* meaning that the leakage model used to attack does possibly not perfectly fit to real observations) or algorithmic noise (*i.e.* produced by parasitic computations in an implementation). All these disturbances similarly affect the efficiency of a side-channel attack and reduce the amount of information in the leakages.

2.3 Classical attacks: SPA and DPA

Beyond the previous classification of physical attacks (*i.e.* invasive *vs.* non-invasive, active *vs.* passive), the literature also classifies the attacks according to the statistical treatment applied to the leakage traces. For example, "simple" and "differential" attacks were introduced in the context of power analysis [11].

Simple Power Analysis (SPA) attempts to interpret the power consumption of a device and deduce information about its performed *operations*. This is nicely illustrated with the example in Figure 2. It shows the power consumption trace of a device performing an AES (Advanced Encryption Standard) encryption [19]. The figure clearly shows a pattern that is repeated 10 times and corresponds to the 10 rounds of the AES when implemented in its 128-bit version.

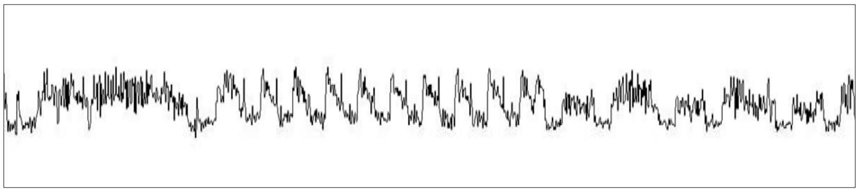


Fig. 2: SPA monitoring from a single AES encryption performed by a smart card.

Of course, this information is not an attack in itself. Everybody knows that AES-128 has 10 rounds, and knowing that a device is performing an AES encryption does not expose its secrets at all. However, such a visual inspection of the leakage traces may the preliminary step in a more powerful attack, *e.g.* by determining the parts of the traces that are relevant to the adversary. In addition, there are cases in which this sequence of operations can provide useful information, mainly when the instruction flow depends on the data. Modular exponentiation performed with a square and multiply algorithm is a good example. If the square operation is implemented differently than the multiply - a tempting choice, as this will allow specific optimizations for the square operation, resulting in faster code - and provided this difference results in different consumption patterns, then the power trace of an exponentiation directly yields the (secret) exponent's value. Generally speaking, all programs involving conditional branch operations depending on secret parameters are at risk.

By contrast, **Differential Power Analysis (DPA)** intends to take advantage of *data*-dependencies in the power consumption patterns. It is again better illustrated with an example. Figure 3 shows power consumption curves that typically correspond to the simple Hamming weight or distance leakage models introduced in Section 2.1. These data dependencies exploited by powerful statistics lead to a more general class of (so-called differential) attacks that are detailed through an example in the next section.

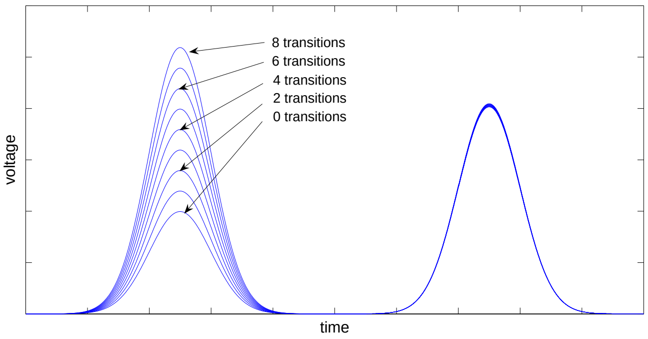


Fig. 3: Illustration of Hamming weight or distance data-dependencies in the power consumption traces of a smart card using an 8-bit data bus.

3 An exemplary differential attack against the DES

A side-channel attack against any cryptographic device typically involves a number of active steps for the adversary. In this section, we aim to illustrate these different steps with an exemplary attack against the DES (Data Encryption Standard) that is briefly described in Appendix A. For simplicity, we follow the practice oriented definition of a side-channel attack introduced in [26].

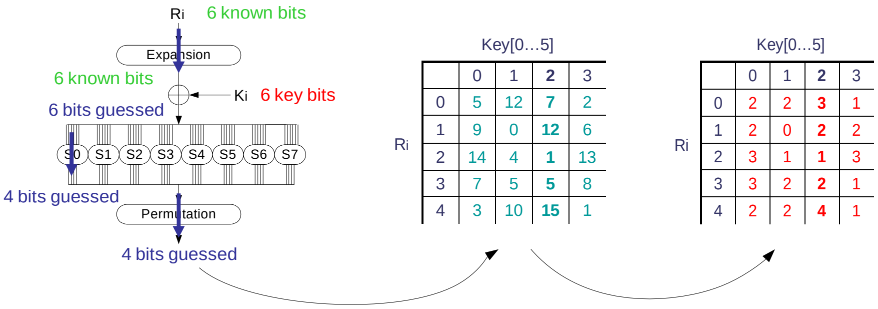
1. Selection of the target algorithm and implementation. The adversary determines the algorithm (*e.g.* the DES) and a target platform (*e.g.* an ASIC, FPGA or smart card) from which he aims to recover secret information.

2. Selection of the leakage source and measurement setup. The adversary determines the type of leakage he wants to exploit, *e.g.* power consumption, electromagnetic radiation or a combination of both. This step includes the preparation of the measurement setup described in Section 2.2.

3. Selection of the target signal. Side-channel attacks are generally based on a divide-and-conquer strategy in which different parts of a secret key are recovered separately. Consequently, the adversary selects which part of the key is the target of his attack, *e.g.* the six key bits entering the first DES S-box S0. We denote this target part of the block cipher key as a key class s.

4. Selection of the device inputs. If allowed, the adversary selects the inputs that are to be feeded to the target device, *e.g.* randomly. If not allowed, it is generally assumed that a side-channel adversary can monitor the plaintexts.

5. Derivation of internal values within the algorithm. This is the core of the divide-and-conquer strategy. For a number of (known) input plaintexts, the adversary predicts (key-dependent) internal values within the target device that are to be computed during the execution of the algorithm. For computational reasons, only values depending on a small part of the key are useful. For example, one could predict the 4 bits after the permutation in the first DES round, for each of the 64 possible key values entering S0, as illustrated in the central table of Figure 4. As a result of this values derivation phase, the adversary has predicted internal values of the block cipher implementation for q plaintexts and each key class candidate s^* (out of 64 possible ones), stored in vectors $\mathbf{v}_{s^*}^q$'s.



Key[0…5]

Ri	0	1	**2**	3
0	5	12	**7**	2
1	9	0	**12**	6
2	14	4	**1**	13
3	7	5	**5**	8
4	3	10	**15**	1

Key[0…5]

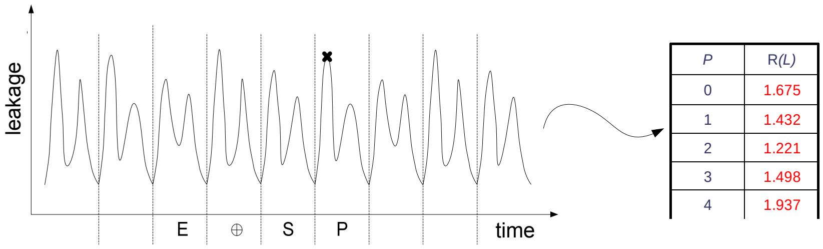
Ri	0	1	**2**	3
0	2	2	**3**	1
1	2	0	**2**	2
2	3	1	**1**	3
3	3	2	**2**	1
4	2	2	**4**	1

Fig. 4: Derivation of the internal values and leakage modeling within the DES.

6. Modeling of the leakage. For the same set of key class candidates as during the derivation of the internal values, the adversary models a part or function of the actual target device's leakage. For example, assuming that the power consumption in CMOS devices depends on the switching activity occurring during a computation, the Hamming weigh or distance models can be used to predict the leakage, as illustrated in the right table of Figure 4. In this context, the models are directly derived from the internal values, *e.g.* $\mathsf{M}(s^*, \mathbf{v}_{s^*}^q) = \mathrm{H_W}(\mathbf{v}_{s^*}^q)$.

7. Measurement of the leakage. Thanks to his measurement setup, the adversary monitors the leakage (*e.g.* the power consumption) of the target device. As a consequence, he obtains a leakage vector $\mathbf{l}_q = [l_1, l_2, \ldots, l_q]$ that contains q leakage traces l_i's corresponding to the encryption of q different plaintexts.

8. Selection of the relevant leakage samples. Since the leakage traces obtained from an acquisition device may contain hundreds of thousands samples, actual side-channel adversaries usually reduce the data-dimensions to lower values. This may be done using simple techniques such as SPA or by using advanced statistical processing. In the example of Figure 5, only the maximum value of the clock cycle corresponding to the DES permutation is extracted from the traces. As a result of this phase, the adversary obtains a reduced vector: $\mathsf{R}(\mathbf{l}_q)$.



P	R(L)
0	1.675
1	1.432
2	1.221
3	1.498
4	1.937

Fig. 5: Selection of the relevant leakage samples thanks to a transform T.

9. Statistical comparison. For each of the key class candidates, the adversary finally applies a statistic to compare the predicted leakages with the transformed measurements. If the attack is successful, it is expected that the model corresponding to the correct key candidate gives rise to the best comparison result. For example, in our previous illustrations, the values derivation vectors $\mathbf{v}_{s*}^{q}$ and reduced traces $\mathsf{R}(l_i)'s$ both have q elements. Therefore, if we store the hypothetical Hamming weight models in a vector $\mathbf{m}_{s*}^{q} = \mathrm{H_W}(\mathbf{v}_{s*}^{q})$ the empirical correlation coefficient can be used for comparison [5] :

$$\mathrm{corr}(s^*) = \frac{\sum_{i=1}^{q}(l_i - \hat{\mathbf{E}}(\mathsf{R}(\mathbf{l}_q))) \cdot (m_{s^*}^{i} - \hat{\mathbf{E}}(\mathbf{m}_{s^*}^{q}))}{\sqrt{\sum_{i=1}^{q}(l_i - \hat{\mathbf{E}}(\mathsf{R}(\mathbf{l}_q)))^2 \cdot \sum_{i=1}^{q}(m_{s^*}^{i} - \hat{\mathbf{E}}(\mathbf{m}_{s^*}^{q}))^2}}, \tag{3}$$

where $\hat{\mathbf{E}}(.)$ denotes the empirical mean. In Figure 6, such a correlation attack is applied to our leaking DES implementation and the coefficient is computed for an increasing number of observations. It clearly illustrates that the attack is successful after approximately 100 measured encryptions.

4 Improved side-channel attacks

The previous section described a typical side-channel attack against an unprotected implementation of the DES, based on simple statistical tools and leakage models. This section aims to put forward how such a simple attack can be improved. As a matter of fact, such improvements basically correspond to the improvement of any of the individual steps in the previous section. Specifically, the following ideas are generally considered in the literature:

1. Improving the measurement setup, by reducing any possible source of noise, better designing the side-channel probes, . . . This is a preliminary step to the development of any powerful side-channel attack.
2. Selecting the inputs adaptively as suggested and analyzed in [12].
3. Post-processing the side-channel leakage traces, *e.g.* by averaging or filtering.
4. Improving the leakage models, *e.g.* by profiling and characterizing the target device or by gaining information about critical implementation details.

Key[0...5]	0	1	**2**	3
corr	-0.09	0.05	**0.32**	-0.11

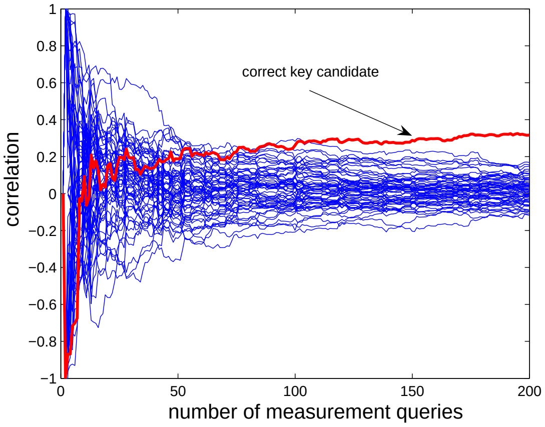


Fig. 6: Statistical comparison with the correlation coefficient.

5. Taking advantage of multivariate statistics, either by using the so-called "higher-order" attacks [15] or by considering optimal strategies such as template attacks [6] or stochastic models [22] (which generally require to characterize the device leakage prior to the actual application of the attack).
6. Using various statistical tests: difference of mean tests, correlation analysis or Bayesian classification are the most frequently considered ones.
7. Combining various types of side-channel leakages, *e.g.* power and EM [2]. With this respect, it is interesting to see that different side-channels generally give rise to different types of information. As an illustration, we provide two exemplary leakage traces of the same leaking device in appendix B, respectively corresponding to the power and EM channels. They clearly illustrate that, *e.g.* the field orientation and therefore the current direction within the device can be obtained from actual EM measurements while the power leakages only provide information about the amplitude of this current.

In practice, the Bayesian classification of key classes based on side-channel leakages exploiting the statistical profiling of a target device is usually denoted as a template attack [6]. It is particularly important both for theoretical and practical reasons. First from a theoretical point of view, it is usually assumed that such a side-channel attack is the most powerful from an information theoretic point of view. Consequently, it has important consequences in the security evaluation of a cryptographic device and when provable security issues are discussed [26]. But in practice, it also corresponds to a significantly different implementation context than the previously described differential attack. Indeed the construction of a statistical model for the side-channel leakages (*i.e.* templates) requires the profiling of a target device. In the worst case, this may involve the ability to change the keys within a device that is identical to the target. For these reasons, we now provide a second illustrative example of a side-channel attack, exploiting templates. We use the same steps as in the previous sections in order to put forward the specificities of such an adversarial context.

4.1 A exemplary profiled attack against the DES

The main objective of a profiled attack is to take advantage of a better leakage model than, *e.g.* assuming Hamming weight dependencies. For this purpose, one generally starts by profiling or characterizing the device leakages with a statistical model. In practice, this involves an additional step in the attack.

0. Preparation of the leakage model. Different approaches can be used for this purpose. The most investigated solution is to assume that the leakage samples $\mathsf{R}(l_i)$'s were drawn from a Gaussian distribution[1]:

$$\mathcal{N}(\mathsf{R}(l_i)|\mu_s^i,\sigma_s^i) = \frac{1}{\sigma_s^i\sqrt{2\pi}}\ \exp\frac{-(\mathsf{R}(l_i)-\mu_s^i)^2}{2{\sigma_s^i}^2}, \tag{4}$$

in which the means μ_s^i and standards deviation σ_s^i specify completely the noise associated to each key class s. In practice, these parameters are estimated thanks to sets of typically a few hundreds to a few thousands traces. As a consequence, the adversary has an estimation of the probabilities $\Pr[s^*|l_i]$ with the Gaussian distribution $\hat{\Pr}[\mathsf{R}(l_i)|s^*] = \mathcal{N}(\mathsf{R}(l_i)|\hat{\mu}_{s^*}^i, \hat{\sigma}_{s^*}^i)$ where $\hat{\mu}_s^i$ and $\hat{\sigma}_s^i$ respectively denote the sample mean and variance for a given leakage sample.

Once the leakage model has been characterized, the adversary follows essentially the same steps as during a classical differential attack, with only a few differences in steps 6 and 9 that we re-detail as follows.

6. Modeling of the leakage. Rather than using the Hamming weights of some internal (key dependent) values within the device, the adversary uses the previously defined probabilistic model. That is, $\mathsf{M}(s^*, \mathsf{R}(l_i)) = \hat{\Pr}[\mathsf{R}(l_i)|s^*]$.

9. Statistical comparison. Finally, from the estimated conditional probabilities $\hat{\Pr}[\mathsf{R}(l_i)|s^*]$'s, the adversary applies Bayes theorem and selects the key classes according to their likelihood: $\mathrm{L}(s^*) = \hat{\Pr}[s^*|\mathsf{R}(\mathbf{l}_q)]$. In Figure 7, such a template attack is applied to our leaking DES implementation and the key likelihoods are computed for an increasing number of observations. It clearly illustrates that the attack is successful after approximately 50 measured encryptions.

[1] We just consider the univariate case in this example. But the extension towards the multivariate case where several leakage samples are considered is straightforward. Note also that in practice, one has to decide what to characterize. For example, one can build templates for different key candidates or for different Hamming weights at the output of an S-box. The selection of operations are data to characterize is important from a practical point of view since it determines the computational cost of the attack (*i.e.* building more templates is more expensive).

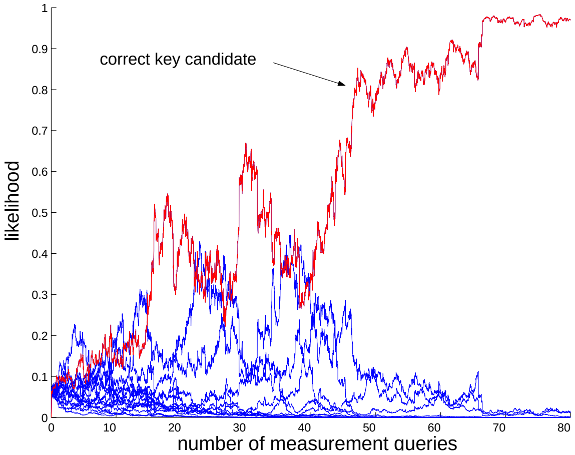


Fig. 7: Statistical comparison with the correlation coefficient.

5 Countermeasures

In this section, we finally describe possible countermeasures to prevent side-channel attacks and discuss the resulting security *vs.* efficiency tradeoff. Some of these techniques are extensively described in the following chapter of this book.

Countermeasures against side-channel attacks range among a large variety of solutions. However, in the present state-of-the-art, no single technique allows to provide perfect security. Protecting implementations against physical attacks consequently intends to make the attacks harder. In this context, the implementation cost of a countermeasure is of primary importance and must be evaluated with respect to the additional security obtained. The exhaustive list of all possible solutions to protect cryptographic devices from side-channel opponents would deserve a long survey in itself. In this section, we only suggest a few examples in order to illustrate that security can be added at different abstraction levels:

1. At the physical level, shields, conforming glues [3], physically unclonable functions [28], detectors, detachable power supplies [23], . . . can be used to improve the resistance of a device against physical attacks.
2. At the technological level, dynamic and differential logic styles (as an alternative to CMOS) have been proposed in various shapes (*e.g.* [27]) to decrease the data-dependencies of the power consumption.
3. At the algorithmic level, time randomization [14], encryption of the buses [4], hiding (*i.e.* making the leakage constant) or masking (*i.e.* making the leakage dependant of some random value, *e.g.* in [9]) are the usual countermeasures.
4. At all the previous levels, noise addition is the generic solution to decrease the amount of information in the side-channel leakages.
5. Countermeasures also exist at the protocol level, *e.g.* based on key updates.

6 Conclusions

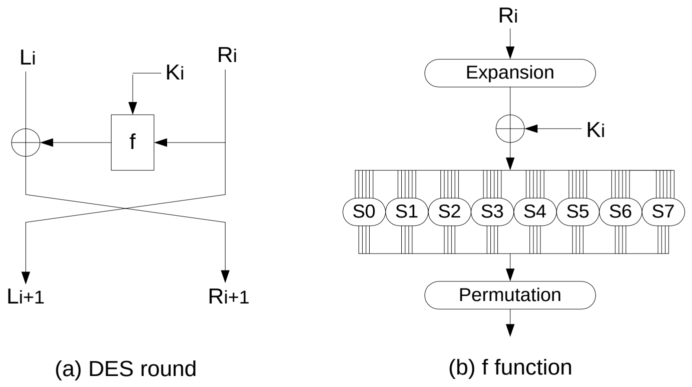
Side-channel attacks are an important class of cryptanalytic techniques. Although less generic than classical cryptanalysis, since they target a specific implementation rather than an abstract algorithm, they are generally much more powerful. Such attacks are applicable to most (if not all) present circuit technologies and have to be considered as a serious threat for the security of actual embedded devices. From an operational point of view, security against side-channel attacks can be obtained by the sound combination of various countermeasures. However, significant attention has to be paid to the fair evaluation of these countermeasures in order to properly assess the security of any cryptographic device and trade it with implementation efficiency [26]. Additionally, side-channel attacks are only a part of the physical reality and resisting them may induce weaknesses with respect to other issues. The development of a unified framework for the analysis of physical security concerns and possibly a theory of provable physical security is a long term goal in cryptographic research, initiated in [8, 17, 29].

References

1. D. Agrawal, B. Archambeault, J. Rao, P. Rohatgi, *The EM Side-Channel(s)*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 29-45, Redwood City, CA, USA, August 2002.
2. D. Agrawal, J. Rao, P. Rohatgi, *Multi-channel Attacks*, in the proceedings of CHES 2003, LNCS, vol 2779, pp 2-16, Cologne, Germany, Sept. 2003.
3. R. Anderson, M. Kuhn, *Tamper Resistance - a Cautionary Note*, in the proceedings of the USENIX Workshop on Electronic Commerce, pp 1-11, Oakland, California, USA, November 1996.
4. E. Brier, H. Handschuh, C. Tymen, *Fast Primitives for Internal Data Scrambling in Tamper Resistant Hardware*, in the proceedings of CHES 2001, LNCS, vol 2162, pp 16-27, Paris, France, May 2001, Springer-Verlag.
5. E. Brier, C. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 16-29, Boston, Massachusetts, USA, August 2004.
6. S. Chari, J. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 13-28, CA, USA, August 2002.
7. ECRYPT Network of Excellence in Cryptology, *The Side-Channel Cryptanalysis Lounge* , http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html.
8. R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, T. Rabin, *Algorithmic Tamper-Proof Security: Theoretical Foundations for Security Against Hardware Tampering*, in the proceedings of TCC 2004, Lecture Notes in Computer Science, vol 2951, pp 258-277, Cambridge, MA, USA, February 2004.
9. L. Goubin, J. Patarin, *DES and Differential Power Analysis*, in the proceedings of CHES 1999, Lecture Notes in Computer Science, vol 1717, pp 158-172, Worcester, Massachussets, USA, August 1999.
10. P. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems*, in the proceedings of Crypto 1996, Lecture Notes in Computer Science, vol 1109, pp 104-113, Santa Barbara, California, USA, August 1996.

11. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Crypto 1999, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa-Barbara, California, USA, August 1999.
12. B. Köpf, D. Basin, *an Information Theoretic Model for Adaptive Side-Channel Attacks*, CCS 2007, Alexandria, VA, USA, October 2007.
13. S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Chapter 3, Section 4, Springer, 2007.
14. D. May, H. Muller, N. Smart, *Randomized Register Renaming to Foil DPA*, in the proceedings of CHES 2001, Lecture Notes in Computer Sciences, vol 2162, pp 28-38, Paris, France, May 2001, Springer-Verlag.
15. T.S. Messerges, *Using Second-Order Power Analysis to Attack DPA Resistant Software.*, in the proceedings of CHES 2000, Lecture Notes in Computer Science, vol 2523, pp 238–251, Worcester, MA, USA, August 2000.
16. T.S. Messerges, E.A. Dabbish, R.H. Sloan, *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Transactions on Computers, vol 51, num 5, pp 541-552, May 2002.
17. S. Micali, L. Reyzin, *Physically Observable Cryptography*, in the proceedings of TCC 2004, Lecture Notes in Computer Science, vol 2951, pp 278-296, Cambridge, Massachusetts, USA, February 2004.
18. National Bureau of Standards, FIPS 46, *The Data Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, 1977.
19. National Bureau of Standards, FIPS 197, *"Advanced Encryption Standard,"* Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, 2001.
20. E. Peeters, F.-X. Standaert, J.-J. Quisquater, *Power and Electromagnetic Analysis: Improved Models, Consequences and Comparisons*, in Integration, the VLSI Journal, vol 40, pp 52-60, Spring 2007.
21. Jan M. Rabaey, Digital Integrated Circuits, *Prentice Hall International*, 1996.
22. W. Schindler, K. Lemke, C. Paar, *A Stochastic Model for Differential Side-Channel Cryptanalysis*, in the proceedings of CHES 2005, Lecture Notes in Computer Science, vol 3659, pp 30-46, Edinburgh, Scotland, September 2005.
23. A. Shamir, *Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies*, in the proceedings of CHES 2000, Lecture Notes in Computer Sciences, vol 1965, pp 238-251, Worcester, Massachusetts, USA, August 2000.
24. A. Shamir, E. Tromer, *Acoustic cryptanalysis On nosy people and noisy machines*, available from http://theory.csail.mit.edu/ tromer/acoustic/
25. F.-X. Standaert, E. Peeters, F. Macé, J.-J. Quisquater, *Updates on the Security of FPGAs Against Power Analysis Attacks*, in the proceedings of ARC 2006, LNCS, vol 3985, pp 335-346, Delft, The Netherlands, March 2006, Springer-Verlag.
26. F.-X. Standaert, T.G. Malkin, M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, International Association of Cryptographic Research, Cryptology ePrint Archive, Report 2006/139.
27. K. Tiri, M. Akmal, I. Verbauwhede, *A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards*, in the proceedings of ESSCIRC 2003.
28. P. Tuyls, G.J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, R. Wolters, *Read-Proof Hardware from Protective Coatings*, in the proceedings of CHES 2006, LNCS, vol 4249, pp 369-383, Yokohama, Japan, October 2006.
29. UCL Crypto Group, *Theoretical Models for Side-Channel Attacks*, home page and related publications: http://www.dice.ucl.ac.be/ fstandae/tsca.

A The Data Encryption Standard : a case study



(a) DES round

(b) f function

Fig. 8: Data Encryption Standard.

In 1977, the DES algorithm [18] was adopted as a Federal Information Processing Standard (FIPS) for unclassified government communication. Although a new Advanced Encryption Standard was selected in October 2000 [19], DES is still widely used, particularly in the financial sector. DES encrypts 64-bit blocks with a 56-bit key and processes data with permutations, substitutions and XOR operations. The plaintext is first permuted by a fixed permutation *IP*. Next the result is split into two 32-bit halves, denoted with L (left) and R (right) to which a round function is applied 16 times. The ciphertext is calculated by applying the inverse of the initial permutation *IP* to the result of the 16th round. The secret key is expanded by the key schedule algorithm to sixteen 48-bit round keys K_i and in each round, a 48-bit round key is XORed to the text. The key schedule consists of known bit permutations and shift operations. Therefore, finding any round key bit directly involves that the secret key is corrupted. The round function is represented in Figure 8 (a) and is easily described by:

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus f(R_i, K_i) \end{aligned}$$

where f is a nonlinear function detailed in Figure **??** (b): the R_i part is first expanded to 48 bits with the E box, by doubling some R_i bits. Then, it performs a bitwise modulo 2 sum of the expanded R_i part and the 48-bit round key K_i. The output of the XOR function is sent to eight non-linear S-boxes. Each of them has six input bits and four output bits. The resulting 32 bits are permuted by the bit permutation P. Finally, DES decryption consists of the encryption algorithm with the same round keys but in reversed order.

B Exemplary power and EM leakage traces

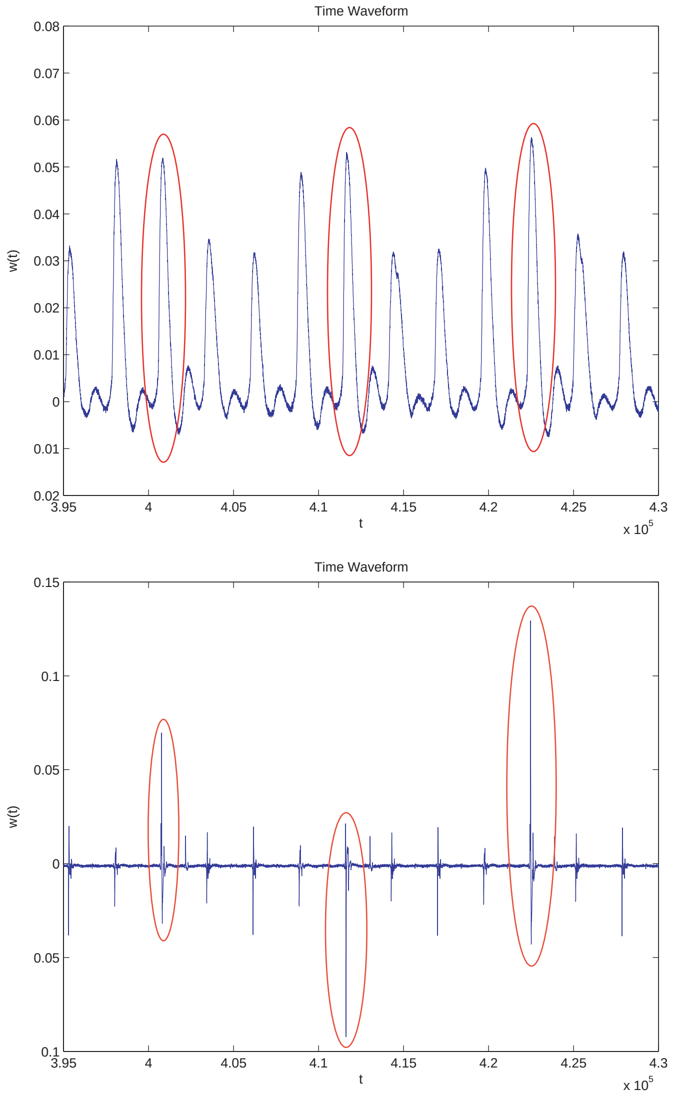


Fig. 9: Exemplary power and EM leakage traces.