

Chapter 2

Side-Channel Cryptanalysis: A Brief Survey

Traditionally, the mathematical cryptanalysis considers that the cryptographic device is an abstract machine and target primarily the weaknesses of the cryptographic algorithm by taking advantage of the input and output data. In Shannon's 1949 paper on "Communication Theory of Secrecy Systems", he defined what should be *perfect secrecy*:

The cryptanalyst intercepts a particular ciphertext C and can then calculate, in principle at least, the a posteriori probabilities for the various messages, $\mathbf{P}[M|C]$. It is natural to define perfect secrecy by the condition that, for all C the a posteriori probabilities are equal to the a priori probabilities independently of the values of these. In this case, intercepting the message has given the cryptanalyst no information.

We will dwell deeper into this definition in the last chapter in which we will try to mount a theoretical framework for side-channel analysis. However, the idea is important since it shows us that if the condition of perfect secrecy is not fulfilled, information on the secret key is then available.

In the real world, an adversary has also access to the cryptographic device and can tamper with it or monitor some physical leakages that emanate from the chip. These are classified as implementation attacks which target the cryptographic device itself. These attacks can be active attacks which range from changing the environmental conditions to the physical opening of the cryptographic device. Another class of attacks acts in a passive way, just by observing the inherent physical leakage of the cryptographic device. These passive attacks are even more dangerous as they do not leave damage to the cryptographic device that can be detected or recognized later on. These attacks exploit the fact that the cryptographic device itself leaks physical information during the processing of a cryptographic algorithm which can be measured externally. These measurements (e.g. power consumption, electromagnetic emanation ...) can then be used to compromise secret keys of cryptographic algorithms by some statistical methods which are discussed in Chap. 6. It is not surprising that the first targets of these attacks are mobile devices such as smart cards as they have external controllable pins for power supply, clock, and I/O. For concreteness,

the discussion here will be put in that context, although most of it applies to other (cryptographic) devices as well (e.g. ATM systems ...).

As already mentioned in the previous chapter, there are different categories of attacks on embedded devices and we provide the main lines about them hereafter.

2.1 Invasive Attacks

Invasive attacks involve getting access to the silicon to observe, manipulate, and interfere with the system internals. Since invasive attacks typically require relatively expensive infrastructure, they are much harder to deploy.

2.1.1 Micro-Probing

This technique uses a micro-probing workstation to remove part of the passivation layer (protecting the silicon) of an integrated circuit. Subsequently, an attacker can establish a direct contact with the system (usually the data bus). An attacker can then eavesdrop the data during the execution of cryptographic algorithms [HPS99]. These attacks are obviously invasive and passive attacks.

2.1.2 Reverse Engineering

Several attack techniques target particular parts of the smart card namely the buses, memories, CPU, coprocessor, and sensors Deploying such attacks (fault attacks, microprobing...) requires access to the layout of the chip, in order to locate and distinguish internals of the chip. One can make use of image processing and form recognition systems to retrieve the hardware structure from simple microscope pictures (e.g. optical microscope with a CCD camera). Recent techniques [SSAQ02] illuminate the unplugged chip thanks to a focused laser spot and probe the variation of current between power and ground. Shining light on a transistor makes it generating a micro-current depending on its state. This technique can thus reveal the mapping of the integrated circuit as well as the data stored.

2.1.3 Fault Attacks

Fault induction techniques intend to manipulate the environmental conditions of the system (voltage, clock, temperature, radiation, light, eddy current, etc.) to generate faults and observe the related behavior. They are not necessarily invasive as they

often only require a simple depackaging of the device. Consequently, they are often classified as semi-invasive. Most of these attacks target data being computed or manipulated by a cryptographic algorithm. Nevertheless, some of them attempt to corrupt the data directly in the memory. While there are many ways of producing a fault in a mobile device, these attacks can be termed semi-invasive as knowledge of the architecture is often required.

Such attacks can be engineered by simply illuminating a transistor with a laser beam, which causes it to conduct (the photovoltaic effect) [SA02, AK97]. Glitch attacks generate a malfunction by changing the working frequency during computation, thereby, causing some bits to assume a wrong value. The notion of using an error induced during a computation to guess the secret key has been practically observed in implementations of the RSA that use the Chinese Remainder Theorem (CRT) [JLQ99, BDL01, AK97]. Fault induction techniques can lead to both transient and nonreversible faults [QS02, SA02].

2.2 Noninvasive Attacks

Conversely, noninvasive attacks do not require the device to be opened. While they require knowledge of the system, they tend to be cheap and scalable (compared to invasive attacks). There are many forms of noninvasive attacks.

2.2.1 *Timing Attack*

Introduced in 1996 by Kocher [Koc96, DKL+98], timing attack exploits the observation that the computations performed in some of the cryptographic algorithms often take different amounts of time on different inputs. For example, a last reduction step is required in most of modular multiplication technique. Depending on the result at the end of the multiplication, this last reduction may (or not) be necessary. This simple fact makes these implementations of the RSA public-key cryptosystem vulnerable to timing attacks [DKL+98]. Nevertheless, solutions were proposed to overcome this issue [Waltersub99, HachezQ00].

2.2.2 *Simple and Differential Power Analysis*

These attacks [KJJ99] record and analyze the power traces leaked by a device. Two kinds of them are typically used: simple power analysis (SPA) and the differential power analysis (DPA). SPA techniques perform a simple (visual) inspection of the power consumption traces and rely on the identification of the Hamming weight of the data during encryption/decryption. DPA methods allow sensitive information to

be uncovered by performing a statistical analysis. Originally, a difference of mean test was used although more efficient statistical tools can be used as well. DPA technique is very powerful since it is architecture independent and relies on the fact that noise effects tend to cancel out.

2.2.3 Electromagnetic Analysis

In 2001, two entities, sharing information during European project, UCL [QS01] and Gemplus [GMO01] suggested to recover the sensitive information lying in a secure device by exploiting the electromagnetic emanations due to the current flowing through the device. The equivalents of SPA and DPA were introduced and they were termed simple and differential electromagnetic analysis (SEMA and DEMA, respectively). Using such techniques, it becomes possible to monitor only the signal from one particular location of the device (e.g. cryptoprocessor, buses, oscillators, etc.), without being significantly affected by the noise produced from the rest of the chip (usually refereed to as algorithmic noise). This will be showed in Chaps. 3 and 5.

2.3 Attacker's Taxonomy

Finally, the level of tamper resistance offered by any particular product can be measured by the time and cost penalty that the protective mechanisms impose on the attacker. Estimating these penalties is clearly an important problem, and as already said we try to answer to this question in Chap. 8. For this reason, a taxonomy was introduced by IBM to guide designers of security system [AK97]:

- Class I (clever outsiders): They are often very intelligent but may have insufficient knowledge of the system. They may have access to only moderately sophisticated equipment. They often try to take advantage of an existing weakness in the system, rather than try to create one.
- Class II (knowledgeable insiders): They have substantial specialized technical education and expertise. They have varying degrees of understanding of parts of the system but potential access to most of it. They often have highly sophisticated tools and instruments for analysis.
- Class III (funded organizations): They are able to assemble teams of specialists with related and complementary skills backed by great funding resources. They are capable of in-depth analysis of the system, designing sophisticated attacks, and using the most advanced analysis tools. They may use Class II adversaries as part of the attack team.

2.4 Countermeasures

To protect a device against the power and electromagnetic side-channel analyses, cryptographers developed a bunch of defensive measures. Sometimes the effect of a particular countermeasure can be larger than expected (e.g. counteracting power analysis will very often help to prevent electromagnetic analysis), while sometimes a combination of countermeasures can be worse for the security configuration. In all cases, when designing a countermeasure, or a combination of them, designers must be well aware of the respective importance of the different parameters involved. For instance, it is pointless to design a noise generator whose standard deviation is an order of magnitude lower than the signal correlated to the key.

Without being exhaustive, we sift through the different common countermeasures and we distinguish three different levels where they could be introduced. The software level is certainly the cheapest as it only involves to redesign the algorithm code and not to redesign the whole chip. The subsequent price to pay is very often an increased computation time. In the hardware level, the idea is to add some components that make the monitored traces harder to exploit. The main components are for instance a noise generator, a random process interrupt module (random bubble introduced in the pipeline structure), etc. In a third level, the purpose is to tackle the side-channel issue at its root cause. Indeed, the issue of the power analysis lies into the fact that the power consumption of the device is correlated to the data handled. A straightforward solution is to make the power consumption independent of the data handled by making good use of an alternative logic style.

2.4.1 Software Level

This concerns all the possible solutions that a programmer can use to decorrelate the sensitive information manipulated within the execution of a secure algorithm with an unintentional leakage. Mainly, we provide here two general techniques while there exists a lot of dedicated solutions depending on the application aimed.

Transforming and masking data: A *Differential* (Power or Electromagnetic) Analysis cannot be successful if data, computed (loaded or stored) during the algorithm, are not correlated to a few key bits and the recorded trace. Two main ideas were proposed: the *Duplication Method* by Goubin and Patarin [GP99] and the *Boolean and Arithmetic Masking Method* by Coron and Goubin [CG00], Akkar and Giraud [AG01] and more recently in [AG03].

Register renaming and nondeterministic processor (NDISC): The basic idea is that the processor should select an instruction and a memory access randomly, thereby, the author expects to randomize the access pattern to the memory caused by both data and instruction streams [MMS01]. Our opinion is even such a solution could help, one must really care how it is implemented (e.g. the instruction bus or data bus should not be set to all “zeros” after each instruction or data fetch).

2.4.2 Hardware Level

We refer here to all the techniques that may be added to the hardware implementation of the secure device. That often involves the addition of some components that we survey hereafter:

Desynchronizing: The force of DPA relies on the correlation between traces average, the key bits and the inputs/outputs. This event arises at a certain point of the computation. For a dummy implementation, it always occurs after the same time on the trace. So the desynchronization was one of the first step to limit the DPA [ABDM00]. Infineon specialized in such technique by introducing some fake clock cycles during the computation, also mentioned in [KK99]. More recently, STMicroelectronics proposed to use a desynchronization based on a weak jitter but completely controlled. A way to defeat such countermeasures lies in classic resynchronization techniques [CCD00, BT06] based on jitter killer, phase-locked loop (PLL), or Costas loop (used for satellite signal tracking). Another simple way to desynchronize the different traces is to insert some random interrupts while performing a cryptographic algorithm. This simple trick, also known as *rand slip*, is often implemented in secure cores such as those of the MIPS32®4K STM Family.

Noise Generator: A way to defeat the DPA is to increase the noise to drown the signal in. The added noise must be uncorrelated with the underlying signal. This is usually obtained by injecting white noise on the channel [KJJ99] (AWGN Additive White Gaussian Noise). This also can be accomplished by trying to take advantage of some components card to mix up the consumption. The idea is to connect analog or even digital systems to some blocks enabling them to randomly modify the current on the power line.

Suppression Circuit: One of the simplest ways to avoid a correlation between the power supply and the data handled inside the device is certainly to insert a voltage regulator. It works basically thanks to a feedback loop and is thereby ideally able to consume always the same power from an external point of view [RWB04]. It is mainly constituted of a shunt transistor controlled by a opamp. The authors explain that their circuit degrades the SNR of the signal by around -20 dB.

Parallel Computation One way to protect a microchip from EM-analysis is to break the principle of locality [QS01]. The idea is to spread out the main blocks (cryptoprocessor, ROM, RAM, busses,...) of the RTL design over the whole surface of the chip. We investigated this solution on the well-known RSA cryptographic algorithm and came with a new architecture based on the Residue Number System and parallel processing units [CNPQ03]. The main drawback of the system is that it slows down by a factor of 50 the implementation of the RSA compared to common implementation. The same idea was investigated by Bajard et al. [BILT04].

2.4.3 Logic Style

Finally, as already mentioned, another trend is to tackle the problem at its origin and to make good use of an alternative logic style which consumes a constant amount of power. **Decreasing the power consumption: SOI:** One of the current important directions of the research in microelectronics relates to the consumption reduction through multiple techniques. One of them for example uses the silicon installation on insulator (SOI) technology [NFQ99]. SOI allows decreasing the current consumed by the processor, and consequently also reduces the radiated electromagnetic field. The use of such a technique will increase the computing power and might be useful for certain devices (smart cards without contacts, memories,...). The release of heat per Joule effect will also be reduced.

Dual Rail: Another idea proposed in the G3card project and the University of Cambridge was to use a balanced logic (dual rail logic), or even self-timed circuits [MMC+02, MRB+03]. In the asynchronous field, a few implementations of processor architecture were carried out, among these we principally mention the Amulet processors (Manchester University) and XAP processors (Cambridge University).

Dynamic and Differential Logic: As we will explain in the next part, the consumption of widespread CMOS devices is dependent on a bit flip and on the capacitance at a functional blocks output. The bit transition determines whether there is a consumption or not and the amplitude of this consumption is a function of the capacitance value (parasitics and interconnects). A solution would be to have a logic that consumes the same amount of power for every kind of bit transition (i.e. $1 \rightarrow 0$, $1 \rightarrow 1$, $0 \rightarrow 1$, $0 \rightarrow 0$). This issue can be solved by making the logic dynamic, that is, precharging the output in the first half of every clock cycle and evaluating the correct output value in the second half. Two kinds of dynamic and differential logic were investigated independently the SABL [TAV02] and DyCML [MSH+04]. The latter achieves better SNR and is thus considered more resistant than the former.

References

- [ABDM00] Akkar, M.-L., Bevan, R., Dischamp, P., & Moyart, D. (2000). Power Analysis, What Is Now Possible... In T. Okamoto, (Ed.), *ASIACRYPT, Lecture Notes in Computer Science* (Vol. 1976, pp. 489–502). New York: Springer.
- [AG01] Akkar, M.-L., & Giraud, C. (2001). An implementation of DES and AES secure againsts some attacks. In Ç. K. Koç et al. [cKKNPO1], (pp. 309–318).
- [AG03] Akkar, M.-L., & Goubin, L. (2003). A generic protection against high-order differential power analysis. In T. Johansson (Ed.), *FSE, Lecture Notes in Computer Science* (Vol. 2887, pp. 192–205). Berlin: Springer.
- [AK97] Anderson, R. J., & Kuhn, M. G. (1997). Low cost attacks on tamper resistant devices. In B. Christianson, B. Crispo, T. M. A. Lomas & M. Roe (Ed.), *Security Protocols Workshop, Lecture Notes in Computer Science* (Vol. 1361, pp. 125–136). Berlin: Springer.

- [BDL01] Boneh, D., DeMillo, R. A., & Lipton, R. J. (2001). On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, 14(2), 101–119.
- [BILT04] Bajard, J.-C., Imbert, L., Liardet, P.-Y., & Teglia, Y. (2004). Leak resistant arithmetic. In M. Joye & J.-J. Quisquater (Eds.), *CHES, Lecture Notes in Computer Science* (Vol. 3156, pp. 62–75). Berlin: Springer.
- [BT06] Benoit, O., & Tunstall, M. (2006). Efficient use of random delays. Technical report.
- [CCD00] Clavier, C., Coron, J.-S., & Dabbous N. (2000). Differential power analysis in the presence of hardware countermeasures. In Ç. K. Koç & C. Paar [cKKP00], (pp. 252–263).
- [CG00] Coron, J.-S., & Goubin, L. (2000). On boolean and arithmetic masking against differential power analysis. In Ç. K. Koç and C. Paar [cKKP00], (pp. 231–237).
- [cKKNP01] Koç, Ç. K., Naccache, D., & Paar, C. (Eds.). (2001). *Cryptographic hardware and embedded systems - CHES 2001, Third International Workshop, Paris, France, May 14–16, Proceedings, of Lecture Notes in Computer Science* (Vol. 2162). Berlin: Springer.
- [cKKP99] Koç, Ç. K., & Paar, C. (Eds.). (1999). *Cryptographic hardware and embedded systems. First International Workshop, CHES'99, Worcester, MA, USA, August 12–13, Proceedings, of Lecture Notes in Computer Science* (Vol. 1717). Berlin: Springer
- [Waltersub99] Koç, Ç. K., & Paar, C. (Eds.). (2000). *Cryptographic hardware and embedded systems—CHES 2000, Second International Workshop, Worcester, MA, USA, August 17–18, Proceedings, of Lecture Notes in Computer Science* (Vol. 1965). Berlin: Springer.
- [CNPQ03] Ciet, M., Neve, M., Peeters, E., & Quisquater, J.-J. (2003). Parallel FPGA implementation of RSA with residue number systems—can side-channel threats be avoided? In *MWSCAS '03. Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems* (Vol. 2, pp 806–810). Dec 2003.
- [DKL+98] Dhem, J.-F., Koeune, F., Leroux, P.-A., Mestré, P., Quisquater, J.-J. & Willems, J.-J. (1998). A practical implementation of the timing attack. In J.-J. Quisquater & B. Schneier (Eds.). *CARDIS, Lecture Notes in Computer Science* (Vol. 1820, pp. 167–182). Berlin: Springer.
- [GMO01] Gandolfi, K., Mourtel, C., & Olivier, F. (2001). *Electromagnetic analysis: concrete results*. In Ç. K. Koç et al. [cKKNP01], (pp. 251–261).
- [GP99] Goubin, L., & Patarin, J. (1999). DES and differential power analysis (The “Duplication” Method). In Ç. K. Koç & C. Paar [cKKP99], (pp. 158–172).
- [HPS99] Handschuh, H., Paillier, P., & Stern, J. (1999). *Probing attacks on tamper-resistant devices*. In Ç. K. Koç & C. Paar [cKKP99], (pp. 303–315).
- [JLQ99] Joye, M., Lenstra, A. K., & Quisquater, J.-J. (1999). Chinese remaindering based cryptosystems in the presence of faults. *Journal of Cryptology*, 12(4), 241–245.
- [KJJ99] Kocher, P. C., Jaffe, J., & Jun, B. (1999). Differential power analysis. In M. J. Wiener (Ed.), *CRYPTO, Lecture Notes in Computer Science* (Vol. 1666, pp. 388–397). Berlin: Springer.
- [KK99] Kömmerling, O., & Kuhn, M. G. (1999). Design principles for tamper-resistant smart-card processors. In *Proceedings of the USENIX Workshop on Smartcard Technology, Chicago, 10–11 May 1999* (pp. 9–20).
- [Koc96] Kocher, P. C. (1996). Timing attacks on implementations of diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz (Ed.), *CRYPTO, Lecture Notes in Computer Science* (Vol. 1109, pp. 104–113). Berlin: Springer.
- [MMC+02] Moore, S. W., Mullins, R. D., Cunningham, P. A., Anderson, R. J., & Taylor, G. S. (2002). Improving smart card security using self-timed circuits. In *ASYNC, IEEE Computer Society* (p 211–218).
- [MMS01] May, D., Muller, H. L., & Smart, N. P. (2001). Random register renaming to foil DPA. In Ç. K. Koç et al. [cKKNP01] (pp. 28–38).
- [MRB+03] Maurine, P., Rigaud, J.-B., Bouesse, G. F., Sicard, G., & Renaudin, M. (2003). Statistic implementation of QDI asynchronous primitives. In J. J.-Chico & E. Macii (Eds.), *PAT-MOS, Lecture Notes in Computer Science* (Vol. 2799, pp. 181–191). Berlin: Springer.

- [MSH+04] Mace, F., Standaert, F.-X., Hassoune, I., Legat, J.-D., & Quisquater, J.-J. (2004). A dynamic current mode logic to counteract power analysis attacks. In *DCIS 2004*. (pp. 186-191).
- [NFQ99] Neve, A., Flandre, D., & Quisquater, J.-J. (1999). Feasibility of smart cards in Silicon-on-insulator (SOI) technology. (pp. 1-7).
- [QS01] Quisquater, J.-J., & Samyde, D. (2001). ElectroMagnetic analysis (EMA): Measures and counter-measures for smart cards. In I. Attali & T. P. Jensen (Eds.), *E-smart, Lecture Notes in Computer Science* (Vol. 2140, pp. 200-210). Berlin: Springer.
- [QS02] Quisquater, J.-J., & Samyde, D. (2002). Eddy current for magnetic analysis with active sensor. In *Proceedings of Esmart 2002* (3rd ed.), Sept 2002. (pp. 183-194).
- [RWB04] Ratanpal, G. B., Williams, R. D., & Blalock, T. N. (2004). An on-chip signal suppression countermeasure to power analysis attacks. *IEEE Transactions on Dependable and Secure Computing*, 1(3), 179-189.
- [SA02] Skorobogatov, S. P., & Anderson, R. J. (2002). Optical fault induction attacks. In B. S. Kaliski Jr., Ç. K. Koç & C. Paar (Eds.), *CHES, Lecture Notes in Computer Science* (Vol. 2523, pp. 2-12). Berlin: Springer.
- [SSAQ02] Samyde, D., Skorobogatov, S. P., Anderson, R. J., & Quisquater, J.-J. (2002). On a new way to read data from memory. In *IEEE Security in Storage Workshop* (pp. 65-69).
- [TAV02] Kocher, P.C., Jaffe, J., Jun, B. (2002). A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of ESSCIRC 2002*.
- [Waltersub99] Walter, C.D. (1999). Montgomery exponentiation needs no final subtractions. *Electronics Letters*. 35(21), 1831-1832
- [HachezQ00] Hachez, G., & Quisquater, J.-J. (2000). Montgomery exponentiation with no final subtractions: Improved results, *CHES* (pp. 293-301). http://dx.doi.org/10.1007/3-540-44499-8_23

Advanced DPA Theory and Practice
Towards the Security Limits of Secure Embedded
Circuits

Peeters, E.

2013, XVI, 139 p., Hardcover

ISBN: 978-1-4614-6782-3