Université Paris 8 — Vincennes-Saint-Denis
École doctorale Cognition, Langage, Interaction
U.F.R. Informatique

Numéro attribué par la bibliothèque

THÈSE

Pour obtenir le grade de
docteur en Informatique

Présentée et soutenue publiquement
par

JULIEN DOGET

Le 20 décembre 2012

SIDE CHANNEL ANALYSIS AND COUNTERMEASURES

Analyse des attaques par canaux auxiliaires et recherche de contremesures associées

Directeur de thèse:

PROFESSEUR CLAUDE CARLET
PROFESSEUR FRANÇOIS-XAVIER STANDAERT

Encadrant scientifique:

DOCTEUR EMMANUEL PROUFF

Composition du jury:

Dr. Elisabeth OSWALD	Université de Bristol (*Rapporteur)*
Pr. Werner SCHINDLER	Université de Darmstadt (*Rapporteur)*
Pr. Louis GOUBIN	Université de Versailles St-Quentin-en-Yvelines (*Président)*
Dr. François KOEUNE	Université Catholique de Louvain-la-Neuve (*Secrétaire)*
Pr. Ingrid VERBAUWHEDE	Katholieke Universiteit Leuven

« Y'a l'al-gèbre. C'est comme des additions avec des lettres. Pour... pour ceux qu'on pas assez de cervelle pour les chiffres, tu vois ? »

— Frédéric Côlon *Va-t-en-guerre* TERRY PRATCHETT, Les annales du Disque-Monde, livre 21

« Plus un secret a de gardiens, mieux il s'échappe. »

— JACQUES DEVAL

Contents

Acknowledgments

I am sorry but the acknowledgments are available only in French, at page 191 but do not worry, I gratefully thank YOU !

Abstract

This thesis focused on side channel attacks in the field of cryptology. Traditionally, security proofs in cryptology are placed in a model known as *black box*, which assumes that the adversary knows the algorithm used and has only access to an oracle parameterized by a secret and providing the results (encryption or decryption) of its requests. In this model, it is possible to show that for some algorithms, an adversary using an optimal strategy can not find the secret of the oracle faster than exhaustive search. However, the black box model does not enable to prove the security of a system in practice.

In practice, indeed, the adversary may have physical access to the oracle (as it is the case for smart cards, widely used in banking and mobile telephony as security tools). In this context, he can see (or even disrupt) the computations made by the oracle and measure the impact on its environment (*e.g.* power consumption, computing time, *etc.*). These observations are generally related to the values of intermediate results handled by the oracle and thus provide additional information to the adversary, enabling him to find more efficiently the secret. This model where an adversary has access to intermediate values is called *gray box* model.

The use of physical information in order to break a cryptosystem and the study of associated countermeasures are within the scope of the side channel analysis and of this thesis.

Initially, attention is paid to the existing side channel attacks. One goal of this thesis is to provide a formal framework to guide the attacker. In particular, a precise classification of the existing attacks is proposed (Chap. 4). In addition, a link between the different attacks is established (Chap. 5) and a new generic attack more efficient than existing ones is exhibited.

In the second part of this thesis, we analysed in details this new attack (Chap. 6). Many experiments were conducted to validate the link between the various known attacks and the relevance of this new attack (Sect. 8-1).

In the third part of this thesis, we were interested in existing countermeasures: the shuffling of data and the sharing of data (Chap. 9). In this thesis we have proposed new schemes mixing data shuffling and data sharing in order to benefit of both types of countermeasures while limiting their defects. We also proposed a new framework for quantifying the security provided by such techniques (Chap. 10).

Part I

Preliminaries

CHAPTER 1

General Cryptography

1-1 Terminology

To understand what is *cryptography*, it is interesting to take a look at its etymology: from the Greek κρυπτος (*kruptos*) which means "hidden" and γραφειν (*graphein*) which means "writing", *cryptography* refers to the art of *encryption* (also known as *ciphering*) which is the process of rewriting a given message (called *plaintext*) into a non-understandable form (called *ciphertext* or *cryptogram*) for everyone who is not aware of the process. The inverse operation (*i.e.* recovering the plaintext from a ciphertext) is called *decryption* or *deciphering*. At the opposite, trying to recover the plaintext from a ciphertext without any knowledge about the encryption / decryption process (*i.e.* breaking the cipher) is called *cryptanalysis*. Both cryptography and cryptanalysis build up the so-called *cryptology*, the "science of secrecy".

1-2 History

By definition, cryptography is mainly used to provide secure communication* and thus was priorly used for military purpose†. The earliest sign of cryptography is found in the Old Kingdom of Egypt, 4,000 years ago and since this era, the art of cryptography has never stopped to be improved. A well-known historical example is the so-called *Caesar's cipher* used by Julius Caesar (near 50 BC) to protect military significant messages. It consists in replacing each letter of the message by a letter some fixed number of positions down the alphabet (looping back at the end). Such a cipher is called *mono-alphabetic substitution cipher*. This kind of cryptography was widely used until the expansion of the *frequency analysis*, around the ninth century, lead by the Arab mathematician Al-Kindi. Frequency analysis considers the frequency of each letter in the ciphertext to determine the corresponding letter in the plaintext. It was the most fundamental cryptanalytic advance until World War II. Nevertheless, around the sixteenth century, *poly-alphabetic substitution ciphers* were designed to overcome frequency analysis. It is based on the alternation of different mono-alphabetic substitutions. The *Vigenère cipher* is a well-known cipher of this kind, broken many years ago by an improvement of frequency analysis. From the beginning of the nineteenth century, more elaborated poly-alphabetic substitution ciphers were designed, certain based on mechanical devices. Moreover some design recipes began to appear such as the famous *Kerckhoffs' law* which states that the security of a cryptosystem must only rely on a secret parameter called the *key*. In other terms, the knowledge of the whole design (except the key) must not permit to break a given cipher. This reaches to another well-known historical example: the *enigma machine* used by the German during World War II. After this war, the seminal paper of Claude Shannon based on *information theory* introduced the foundation of *modern cryptography* [92].

*As a matter of interest, Kama Sutra recommended cryptography for lovers communication

†For instance, until 1996 in France, cryptography was considered as a military weapon of 2nd category such as tank and military aircraft

1-3 Modern Cryptography

In 1949, Claude Shannon proved the Vernam's principle of *one time pad* in his famous article "Communication theory of secrecy systems". It claims that to reach a *perfect secrecy* encryption, a key must be used once by encryption with a domain definition as large as for the plaintext. This property was already known but Shannon proved that it is sufficient. Such a constraint is impractical in real-life*, nevertheless Shannon also introduced some design principles to reach a *practical secrecy*. The emergence of computers and Internet made it possible to design more complicated layout and brought effective cryptography into a common use. It was the birth of the so-called *modern cryptography* based on the concept of *computational impossibility*†. Its aim is to provide the following security features through a – unsecured – communication channel:

Confidentiality Only authorized people (*i.e.* holding the corresponding key) can read the data.

Authentication The sender identity can be verified.

Integrity Data can be checked against modification during the transfer.

The 1970's saw two major advances. The first Data Encryption Standard (DES) is designed by IBM's team (composed of Horst Feistel and Don Coppersmith among others) and is still in use nowadays. In 1976, Whitfield Diffie and Martin Hellman introduced the concept of *public key cryptography* which revolutionized the cryptography world: *asymmetric* cryptography was born. Since that time, a whole scientific community has emerged and expanded quickly.

For the interested reader, more details about cryptography can be found in the reference books [66, 89, 100] and for a more entertaining reading [94].

*In this configuration the key must have the same length as the plaintext and must be transfered in a secure way. If such a secure way is available, one can directly transfer the plaintext. As a matter of interest, this kind of encryption was used during the Cold War, where the key was transfered in a diplomatic bag.

†An *exhaustive search* (also known as *brute force attack*) can always be performed but will need an impracticable time to succeed (an order of 10^6 billions of years for a 128-bit key).

1-3.1 Symmetric Cryptography

Symmetric Cryptography refers to cryptographic algorithms for which encryption and decryption need the same secret key to be executed. Until the seminal paper of Diffie and Hellman in 1976, it was the only kind of cryptography publicly known. Despite the fact that symmetric cryptography is computationally very fast, it suffers from two main drawbacks. First, it needs a unique different key for each couple of sender/receiver. Secondly, the key exchange issue is not addressed. From a mathematical point of view, a symmetric cipher can be formalized as a bijection function sym parameterized by a key k such that

$$c = \mathsf{sym}(p,k) \ , \tag{1.1}$$

where p is the plaintext and c the ciphertext. Thus, the function sym has the role of encryption whereas decryption is the inverse function sym^{-1}. Resolving $c = \mathsf{sym}(p,\cdot)$ (or similarly $p = \mathsf{sym}^{-1}(c,\cdot)$) must be computationally infeasible without knowing k – even with a large amount of plaintext / ciphertext pairs available – for the cipher to be secure. Symmetric ciphers are commonly split in two categories, *stream ciphers* and *block ciphers*. A side category is *Hash function* which is generally based on symmetric cryptography.

1-3.1.1 Stream Cipher

A stream cipher is an algorithm which aims to behave like a one-time pad cipher. That is it produces a *key stream* as long as the plaintext and then combines this key stream – generally by an exclusive-or operation – with the plaintext to obtain the ciphertext. In practice, stream ciphers rely on an internal state which is fed by an *initial value* at the beginning of the process. Note that it is mandatory to change the initial value at each processing. As a consequence, this initial value is generally the output of a pseudo random generator parameterized by the secret key. For instance, the operation of a key stream generator used to encrypt mobile phone conversation (A5/1) can be seen in Fig. 1-3-1. As a stream cipher generally processes the plaintext bit per bit, it is used to be faster than block cipher.

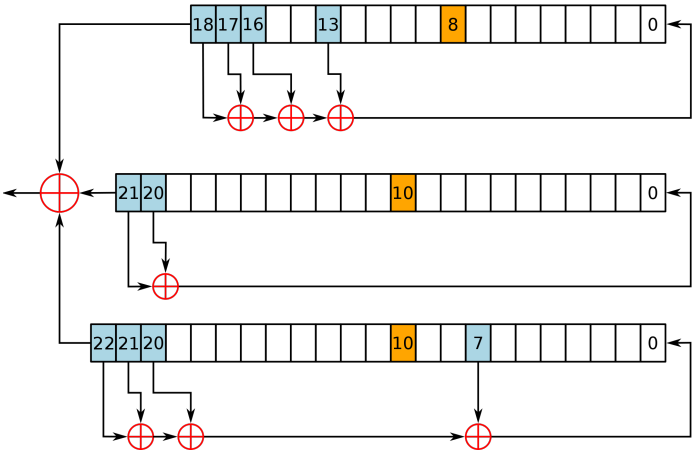


Fig. 1-3-1 – The operation of the key stream generator in A5/1, a LFSR-based stream cipher used to encrypt mobile phone conversations.

1-3.1.2 Block Cipher

A block cipher is a symmetric cipher which takes an n-bit *block* of plaintext as input and outputs an n-bit *block* of ciphertext. When the length of the plaintext is higher than the size of the block, the plaintext is split into blocks that are then encrypted. The encryption of more than one block is defined according to a *mode of operation*. The most used modes are:

- The *Electronic CodeBook* (ECB) mode which encrypts each block of plaintext independently (see Fig. 1-3-2).
- The *Cipher-Block Chaining* (CBC) mode where the previous ciphered block is exclusive-or'ed to the current plaintext block before encryption – an initial value is used for the first block (see Fig. 1-3-3).
- The *counter* mode which aims to simulate a stream cipher encryption producing a key stream from the ciphering of an initial value plus a counter (see Fig. 1-3-4).

These modes must be used with caution. For instance, ECB mode is not assumed secure because identical plaintext blocks yield identical ciphertext blocks (see Fig. 1-3-5 for an illustration). Moreover when an initial value is needed, it must be diversified as often as possible. For example, in CBC mode, the encryption of two plaintexts having the same pre-

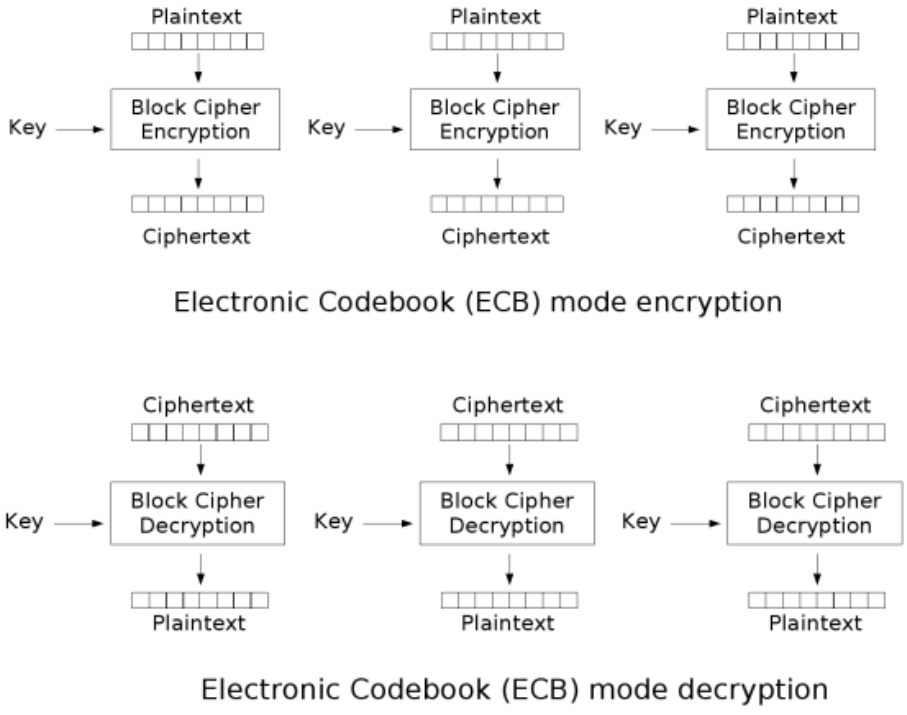


Fig. 1-3-2 – Electronic CodeBook mode ciphering

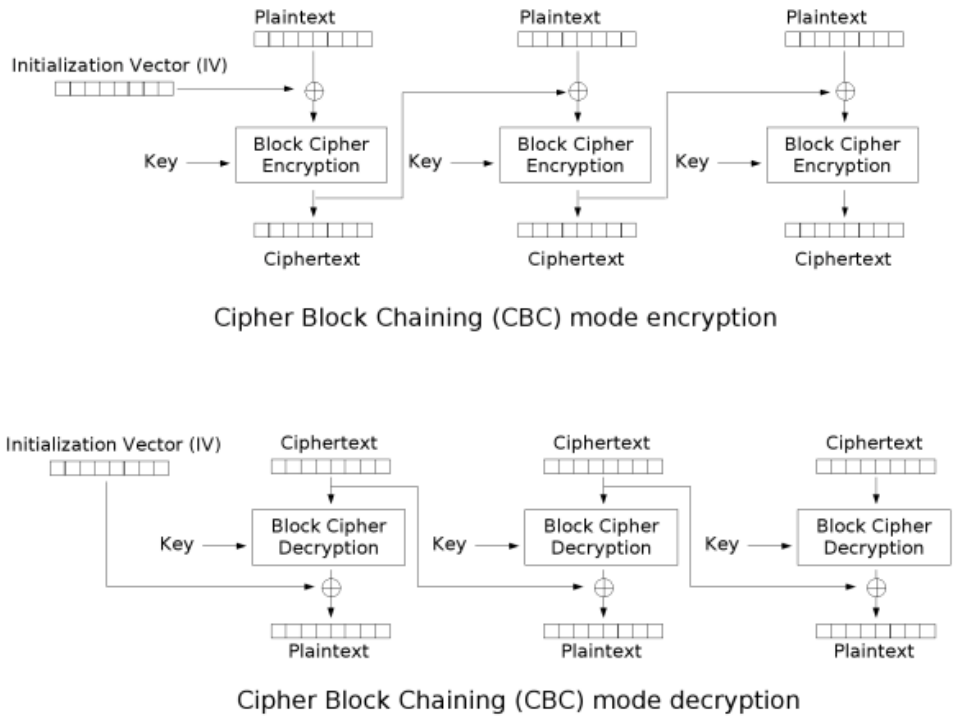


Fig. 1-3-3 – Cipher-Block Chaining mode ciphering

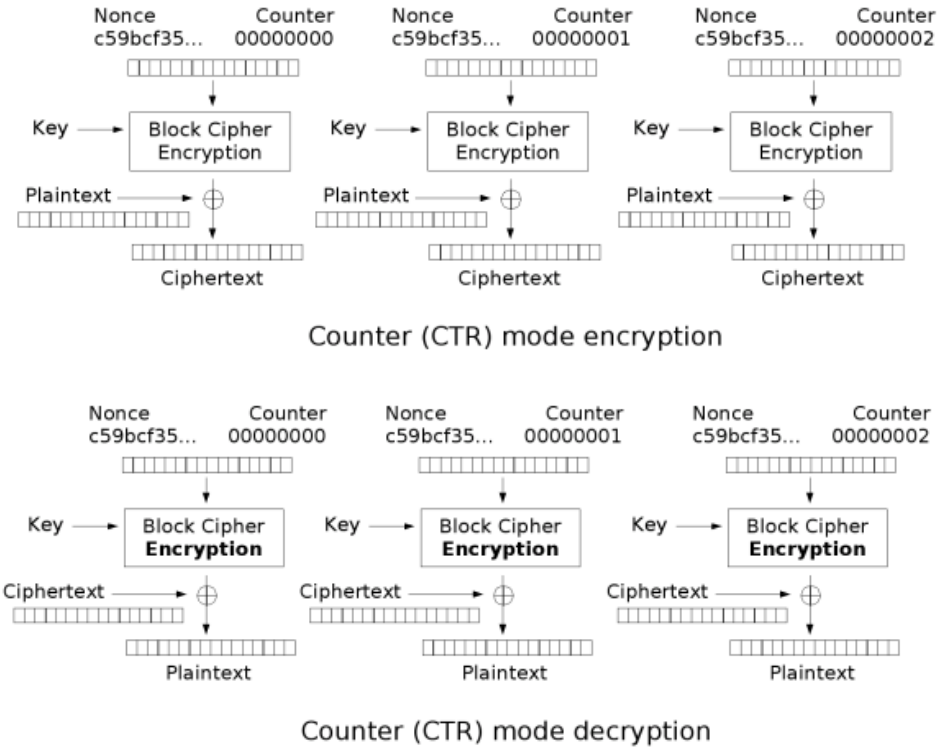
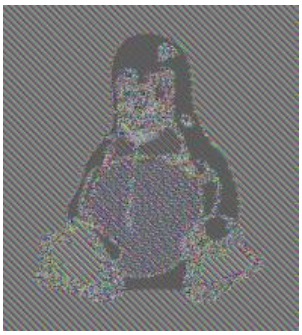


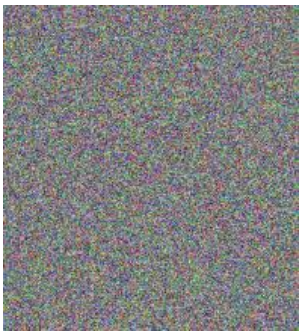
Fig. 1-3-4 – Counter mode ciphering

fix will yield two ciphertexts having the same prefix if the same initial value is used.

Original picture



ECB mode encryption



Another mode encryption

Fig. 1-3-5 – ECB drawback

We present hereafter the two main standard block ciphers, the Data Encryption Standard (DES) [2] and its successor the Advanced Encryption Standard (AES) [1].

DES standard. The DES was adopted by the US National Bureau of Standards in 1976 as a variation of the IBM Lucifer cipher. It processes a 64-bit block of plaintext into a 64-bit block of ciphertext using a 56-bit *master key* (usually expanded to 64-bit with 8 parity bits). It is based on a Feistel network *iterating* a function F called the *round transformation* over 16 rounds (see Fig. 1-3-6). Those 16 rounds come after an initial permutation IP and before a final permutation FP. The internal round transformation processes half a DES state (32-bit) and is parameterized by a 48-bit *round-key*. For more details on the DES structure and its key scheduling, the reader can refer to [3].

AES standard. Due to the large increase of computational power in common computers, the DES algorithm became too weak with regards to the exhaustive search (for instance, only a few hours are needed to find the correct key on a dedicated device [53]). Thus, the US National Institute of Standards and Technology (NIST) decided to launch a competition to find the DES successor. The Rijndael algorithm designed by the two Belgians Vincent Rijmen and Joan Daemen was chosen to be the

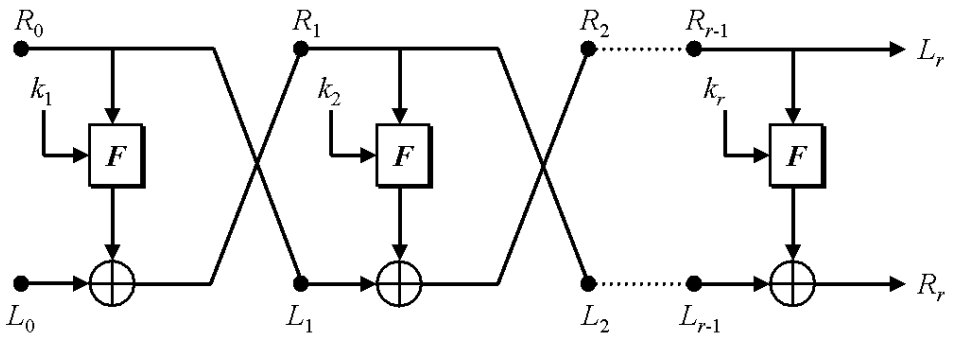


Fig. 1-3-6 – Feistel scheme

Advanced Encryption Standard AES. It is composed of a round transformation iterated 10 times for a 128-bit master key (12 for a 192-bit key and 14 for a 256-bit key). A round transformation is composed of four parts: *SubBytes*, a non linear substitution step; *ShiftRows*, a (row) transposition step; *MixColumns*, a (column) mixing step and *AddRoundKey*, which combines the round key to the state by exlusive or. Such a cipher is called a *Substitution Permutation Network (SPN)*. The AES scheme is recalled in Fig. 1-3-7. For more details on the AES structure and its key scheduling, the reader can refer to [36].

1-3.1.3 Few Words about Hashing

Symmetric ciphers permit to have secure communication. They moreover allow user identification by the way of a *challenge / response* protocol. That is one sends a challenge (*e.g.* a random message) to another (called the *challenger*) which sends back the ciphered challenge. The first sender can now decipher the challenge and thus verify if the challenger owns the same key. Nevertheless, this procedure does not permit to authenticate the message (*i.e.* check the sender identity) nor data integrity. The latter properties can be achieved by *hash functions* and *Message Authentication Code (MAC)*.

Cryptographic hash function. It is a function which computes a fixed length output from an arbitrary length input. A cryptographic hash function must achieve the following properties: *preimage resistance*, given a hash output, it is computationally infeasible to find a corresponding input; *second-preimage resistance*, given an input, it is

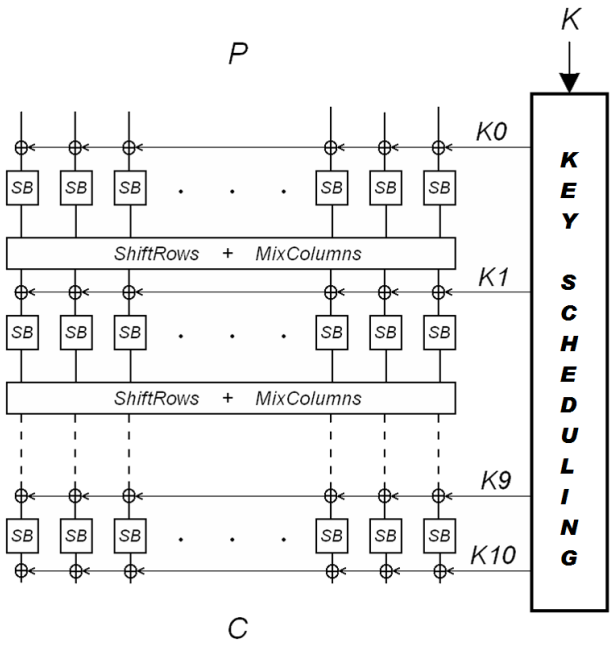


Fig. 1-3-7 – AES cipher description

computationally infeasible to find a second different input with the same hash output as the first one; *collision resistance*, it is computationally infeasible to find two inputs with the same hash output. An hash function permits to check data integrity as a hash value is characterized by the input message. A different message will lead to a different hash output. Nevertheless, the hash values have to be transmitted securely. To bypass the need for a secure transmission, one can imagine a *keyed hash function*. It is the purpose of a *Message Authentication Code*.

Message Authentication Code. It is a function parameterized by a key which computes a fixed length output from an arbitrary length input. It must be computationally infeasible to compute the MAC value of a given message without knowing the secret, even with a large sample of pairs message / MAC available (*existential forgery* property). MAC provides message authentication and data integrity as a receiver can check that the message has been sent by an owner of the key and not modified outside by a non-owner of the key. MAC algorithms are based on hash functions [23] or on block ciphers [25].

1-3.2 Asymmetric Cryptography

Asymmetric cryptography (also known as *public key cryptography*) was officially* invented in 1976 by Diffie and Hellman. The principle relies on a ciphering algorithm parameterized by a pair of key, one – *public* – for encryption and the other – *private* – for decryption such that it is computationally infeasible to deduce the *private key* from the *public key* (*i.e.* it must be computationally infeasible to decipher a ciphertext without the private key, while the public key is known). Thus, the public key is deployed through a *Public Key Infrastructure (PKI)* so that anyone can use it to encrypt a message. At the opposite, the private key is kept secret so that only its owner can decipher messages. The concept of PKI is important to "guarantee" the authentication of the public key owner. It is generally based on the concept of *certificate* and needs a *trusted third party* acting as a *certificate authority*. The well-known RSA (from the names of its authors Rivest-Shamir-Adleman) is the most widely used asymmetric ciphering algorithm. One can also notice other asymmetric cryptosystem such as *ElGamal* or *Elliptic Curve* based algorithms (*e.g. ECIES*).

1-3.2.1 Hybrid Mode

Asymmetric ciphers are based on – computationally – hard mathematical problems, generally from *number theory* such as the *discrete log problem*. Although they provide a *security proof*, it implies much slower algorithms than symmetric ciphers. More annoying, they operate on a fixed-length message! In fact, asymmetric ciphers are well-suited to resolve the key exchange issue of symmetric cryptography. That is in practice – for efficiency reasons – an hybrid ciphering is used: a symmetric random key is generated; then this key is used to cipher a message and finally, this key is ciphered using an asymmetric algorithm with the receiver public key. At the reception, the receiver just needs to decipher the key with its own private key and then decipher the message with the deciphered key.

*It was revealed in 1997 that James Ellis from the GCHQ (UK intelligence agency) had already established this concept in 1970.

1-3.2.2 Few Words about Signature

The concept of *public key cryptography* also renders possible the digital signature of electronic documents. That is, if someone uses his private key to sign a message, then every one can verify the message with the public key and thus can check the sender identity. In a more practical view, signature schemes are often based on the *hash* of the message. To ensure security, it should be computationally infeasible to compute a valid signature without the corresponding private key.

CHAPTER 2

Embedded Environment

2-1 Key Storage

In the previous sections, the main concepts of cryptology have been recalled. In particular, cryptology security is based on the secrecy of a key: the knowledge of the secret key annihilates every security properties ensured by the cryptosystem. Therefore the secure storage of this key became a crucial point. In modern cryptology, the key is usually represented as a long bit stream (generally more than fifty bit length) and thus cannot be memorized by an average human being. Secret keys must therefore be stored in an external device with an easy use (writing the key on a paper is thus not a convenient solution) and in a secure way (storing the key in a laptop under a password-based encrypted form seems to be a good solution in a security point of view, nevertheless the password is most of the time an easily guessable string such as a birthday date and moreover the user

must have its laptop every time a secure communication is required). To ensure practical external secure storage, one has imagined a small token that everyone could carry on in every place. This token would not only securely store the key but also perform the cryptographic computations by itself to avoid key exposure. Such a token already exists and is widely spread: it is known as *smart card*.

2-2 History of Smart Card

The concept of smart cards appeared in the early 70's more or less simultaneously in different countries, though several inventors coexist such as the German Gröttrup and Dethloff, the French Moreno and Ugon, the American Halpern, Castrucci and Ellingboe, the Japanese Arimura *etc.* First represented as a *memory card* (with a secure access), it was then developed in a *microprocessor card* (with an embedded chip). The first mass use of this kind of cards was telephone prepaid cards in France in the early 80's. Soon afterward the first debit cards were spread and smart cards began to expand all over the world. The 90's is another milestone for smart cards with the introduction of smart card based SIM (*Subscriber Identity Module*) in mobile phone equipment. With the exponential growth of chip capabilities, microprocessor cards became as powerful as personal computers and opened the ways to several new applications described in Sect. 2-4.

2-3 A Microprocessor Card

A microprocessor card is a plastic card of dimensions between $85.47 \times 53.92 \times 0.68$ and $85.72 \times 54.03 \times 0.84$ millimeters specified by the ISO/IEC 7810 and ISO/IEC 7816 series of standards [4–18]. An embedded chip is also present under a gold-platted area which acts as the interface between the chip and the card reader. The chip is the association of a microprocessor (a *Central Processing Unit (CPU)*), some memory units, some external communication channels and possibly some dedicated coprocessor. A typical microprocessor card is represented in Fig. 2-3-1

Fig. 2-3-1 – A French student smart card which also includes electronic wallet functionalities.

- The microprocessor is generally a low-cost processor with limited power (an 8-bit architecture with a frequency of 4 MHz is common).
- The memory can be classified in three categories:
  - The *Read-Only Memory (ROM)* which cannot be written nor erased. Thus, the ROM is written once during the manufacturing of the chip and contains the programs executed by the CPU.
  - The *Electrically-Erasable Programmable Read-Only Memory (EEPROM)* which can be written and erased and has the specificity to be non-volatile (*i.e.* the written data are preserved when power is removed). A peculiar low-cost EEPROM, the *flash* memory is also commonly used as it is faster but has a shorter lifetime. When flash memory is used in a chip, it can also take the role of ROM.
  - The *Random Access Memory (RAM)* which is a fast read-write access volatile memory and which is used to store working spaces of programs during their execution.
- Communication protocols can either be with contact through the contact plate or contactless through Radio-Frequency induction technology (*e.g.* using the *Near Field Communication (NFC)* protocol) or both.
- A chip can also include some co-processor such as a random gener-

ator, crypto-processors dedicated to particular algorithms (for instance DES or AES computation), or to particular operations (for instance large number modular computation for asymmetric purpose) or a checksum co-processor for integrity checking *etc.*

2-4 Nowadays

Smart cards are now widely used in cryptographic context for different applications. The most spread use of these applications are reviewed hereafter.

2-4.1 Banking

Smart cards are widely used as debit cards to provide an electronic access to a bank account. Thus it permits electronic payments and cash withdrawals. Currently the major part of debit card does not embed a microprocessor and thus user and bank account information are stored on a magnetic stripe. As magnetic stripe cards become easily copyable and do not contain any user authentication mechanisms (other than a handwriting payer signature), a secured debit card is needed. Nowadays secured debit cards are microprocessor cards with *Personal Identification Number (PIN)* check and strong authentication mechanisms are implemented according usually to the widely used banking standard *Europay Mastercard Visa (EMV)* specifications.

2-4.2 Mobile Telecommunications

In mobile telecommunications, the widely used standards *Global System for Mobile communication (GSM)* and *Universal Mobile Telecommunication System (UMTS)* are based on a SIM smart card. It contains some specific information about the subscriber, about the mobile network and also keys used for authentication on the network. This authentication is performed using a challenge-response protocol, then a secure communication channel can be established between the mobile phone and the provider's antennas. The user is authenticated to the card through the typing of a secret code, the PIN. Without the right PIN, the card will refuse to perform any operation.

2-4.3 Identity Context

Thanks to their authentication capabilities, smart cards are often used for identification purpose. It is generally based on the secure storage of personal data and on issuer certificates involved in a PKI scheme. That is in some countries (*e.g.* Belgium) the national identity card is a smart card which also contains an electronic identity (for instance to access some services by Internet). The driving licence can also be found in a smart card form for instance in Turkey. Another common use in identity cards is the electronic passport (also known as *e-passport*) in which a contactless chip is embedded. The chip contains some identity information such as a digitalized photograph of the holder as well as biometric information (fingerprints and/or iris data) and a certificate of the issuing country. Then an e-passport provides different level of security for authentication from challenge-response protocol to PKI based protocol. Eventually, *access badges* for instance in a company office are also a widely use of smart cards in an identity context. A good example of such a cards is electronic student cards (see Fig. 2-3-1).

2-4.4 Others

Beside the main uses of smart cards described above, we can notice some emerging utilization of cryptographic smart cards:

Health Care Smart cards are used for the secure storage of medical information (social security card).

Public Transit Smart cards take the role of an electronic ticket.

Strong Authentication and Signature Smart cards are viewed as a secure storage / secure generator of key for PKI (secure token).

Pay-TV Smart cards protect digital television stream.

Financial Smart cards act as an electronic wallet or a pre-payment card.

Smart card usages are close one from each other, therefore there is a convergence to a kind of smart card which can take more than one role for instance a mobile SIM card which permits banking transactions and/or electronic ticketing in transport. An example of a convergence card is represented in Fig. 2-3-1.

Consequently, embedded cryptography is actively used in everyday life to secure our communications. Thus the security efficiency is a crucial point. Do smart cards really protect our secrets? As explained in the next section and as developed in the rest of this thesis, this is not a yes/no question. The quantification of the security efficiency is a real issue and having a sound metric represents a real challenge.

2-5 Physical Analysis: Passive Vs Active

In order to evaluate the security efficiency of a cryptographic algorithm, a model is needed to define the scope of an attacker. In theory to evaluate a cryptographic algorithm $\mathscr{A}_k(\cdot)$ parameterized by a secret key k the attacker is usually supposed to know the algorithm and to have access to some – possibly chosen – pairs of (plaintext/ciphertext) to try retrieving the key k: this is the *black box* model. Another model called *white box* model assumes in addition that the attacker has access to every intermediate state of the algorithm. In practice cryptographic algorithms are implemented on a physical device (*e.g.* a smart card) such that an attacker can observe (or even modify) some interactions between the device and its environment (*e.g.* power consumption, timing, *etc.*). These interactions provide a quantity of information about some intermediate state resulting in an attacker model between black box and white box which is naturally called *gray box* model. In this context new kinds of cryptanalytic attacks become possible: *physical cryptanalysis* which can be split into two categories: *side channel analysis* also known as *passive attacks* which exploit the physical leakage during the cryptographic computation to deduce information about the secret key (see Chap. 4) and *fault analysis* also known as *active attacks* which consist in disrupting the cryptographic algorithm to produce faulty outputs analysed to recover information on the secret key.
Remark 1. The gray box model permits to break in practice cryptographic algorithms even when they are proved secure in the black box model

Depending on the targeted device preparation, physical attacks can be divided into three categories:

Invasive attacks The device is deeply unpackaged (and/or modified) to have access to some inner elements such as memory. It allows

very accurate probing attacks as well as very fine destructive attacks (Fig. 2-5-1). This kind of attack is complicated to mount and usually requires high-tech (and high cost) equipment (*e.g.* a chemistry laboratory, a Focused Ion Beam (FIB), *etc.*). Most of the time it results in a definitive alteration of the targeted device.

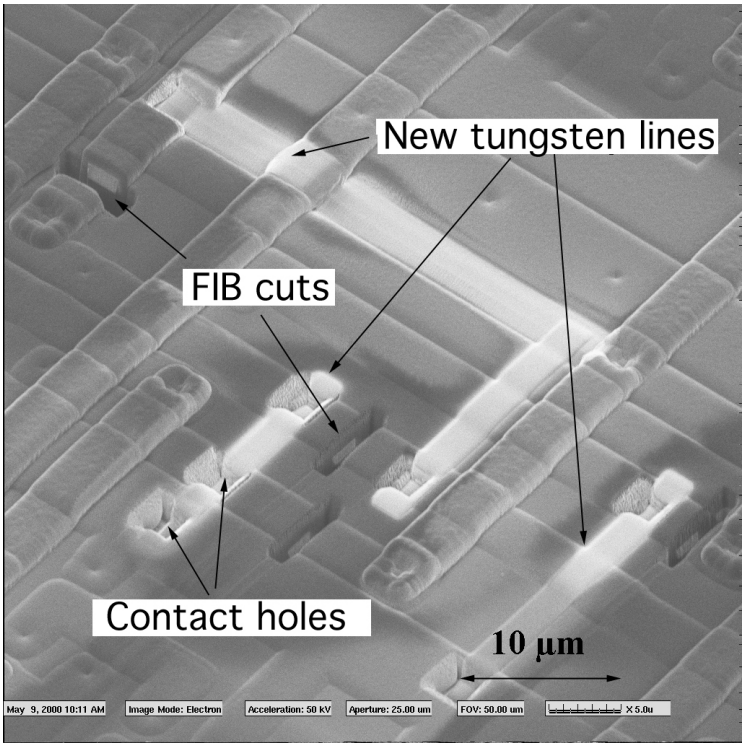


Fig. 2-5-1 – Example of an invasive attack using FIB alteration of an Integrated Circuit.

Semi-invasive attacks The device is partly unpackaged to ease physical cryptanalysis but the chip integrity is not altered (see Fig. 2-5-2).

Non-invasive attacks The targeted device does not need to be altered to perform the attack. In particular, power consumption and clock glitch based attacks are non-invasive ones.

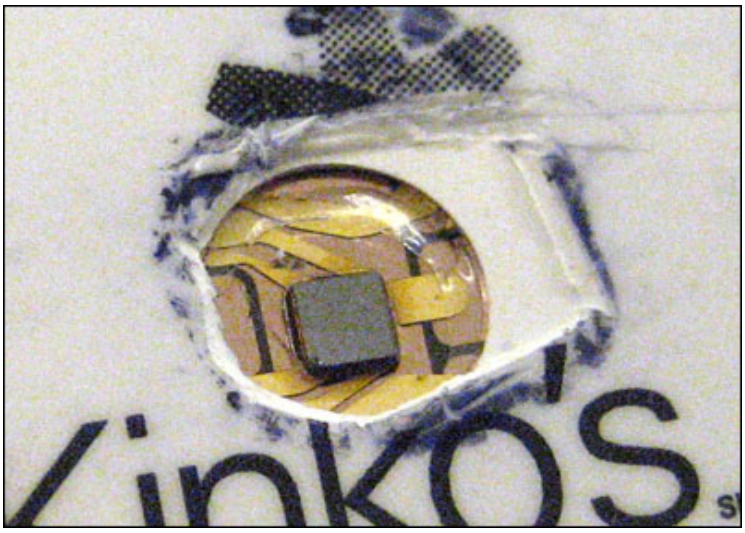


Fig. 2-5-2 – Example of a partial smart card unpacking in a semi-invasive attack.

CHAPTER 3

Technical Background

3-1 Statistics and Probabilities

In the sequel, random variables are denoted by large letters. A *realization* of a random variable X is denoted by the corresponding lowercase letter x. A *sample* of several observations of X is denoted by (x), or by (x_i) if an indexation is needed. It will sometimes be viewed as a vector defined over the definition set of X. The notation $(x) \hookleftarrow X$ denotes the instantiation of the set of observations (x) from X. We shall denote the probabilities associated to events $X \in \mathscr{X}$ and $X = x$ by $\mathrm{P}[X \in \mathscr{X}]$ and $\mathrm{P}[X = x]$ respectively. If X is *continuous*, it is associated with a *probability density function (pdf)* that satisfies for every $x \hookleftarrow X$:

$$\mathrm{P}[X \leqslant x] = \int_{-\infty}^{x} \mathrm{pdf}_X(t)\,\mathrm{d}t \quad .$$

In case of a *discrete* random variable X, it is associated with a *probability mass function (pmf)* defined as $\text{pmf}_X : x \mapsto \text{P}[X = x]$. In this – discrete – case the function $x \mapsto \text{P}[X \leqslant x]$ is called *cumulative distribution function (cdf)* of X. The notation $x \mapsto \text{P}[X = x]$ for a continuous variable X, may be used without ambiguity to denote the pdf of X. In our context, a particular pdf called *Gaussian* pdf plays an important role. It is defined w.r.t. a mean μ and a standard deviation σ by

$$\mathcal{N}_{\mu,\sigma}(u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(u-\mu)^2}{2\sigma^2}} \quad . \tag{3.1}$$

The Gaussian pdf (see Fig. 3-1-1 for some drawn examples) is also called

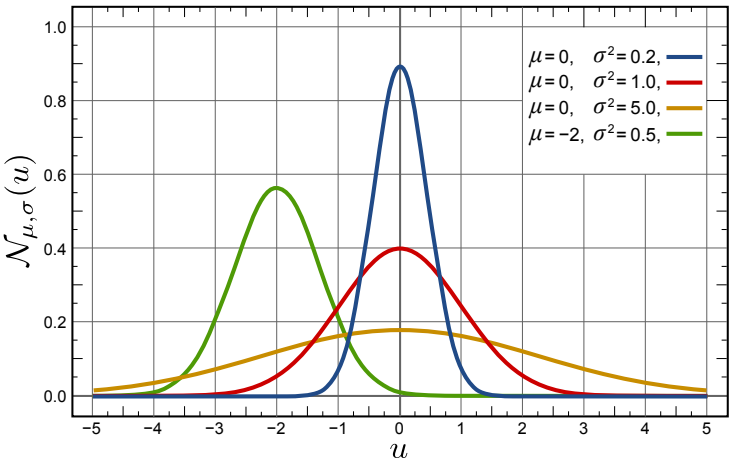


Fig. 3-1-1 – Example of Gaussian probability density functions for some (μ, σ).

the *normal distribution* and can be extended to a d-dimensional random variable $\boldsymbol{U} = (U_1, \ldots, U_d)$ by

$$\Phi_{\boldsymbol{m},\Sigma}(\boldsymbol{u}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{m})\Sigma^{-1}(\boldsymbol{u}-\boldsymbol{m})'} \quad , \tag{3.2}$$

where $\boldsymbol{m} \in \mathbb{R}^d$ stands for the *mean vector* $(\mathbb{E}[U_1], \ldots, \mathbb{E}[U_d])$, where $\Sigma = [m_{i,j}]_{d\times d} \in \mathscr{M}_{d,d}(\mathbb{R})$ stands for the matching *covariance matrix* $m_{i,j} = \text{cov}\left(X_i, X_j\right)$ and where $|\Sigma|$ is the determinant of Σ (see Fig. 3-1-2 for some illustrations). In a more general context, the joint pdf of a Gaussian variable can be represented as a mixture of Gaussian pdf called a *Gaussian Mixture Model (GMM)*. That is a GMM is defined as a weighted sum of Gaussian pdf. Such an example is represented in Fig. 3-1-3

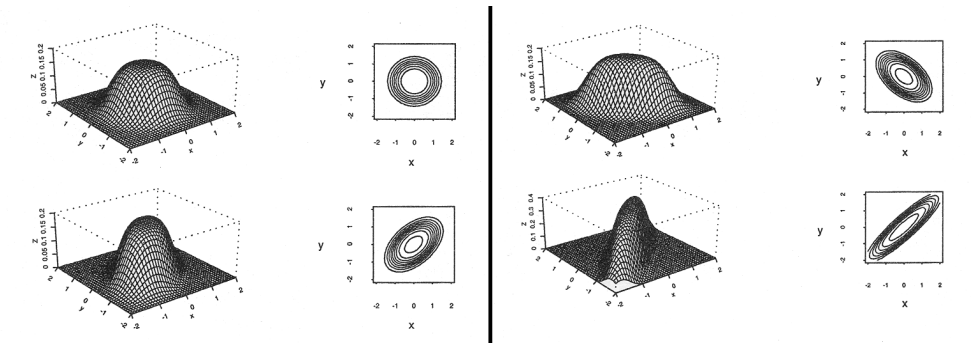


Fig. 3-1-2 – Example of some bivariate Gaussian probability density functions.

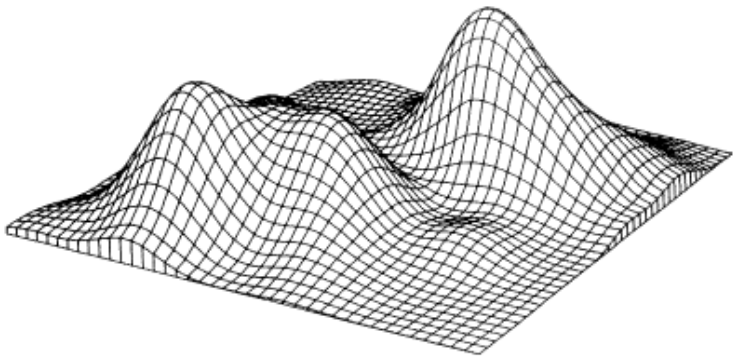


Fig. 3-1-3 – Example of a bivariate Gaussian mixture.

The average value of a random variable X called *expectation* or *mean* is denoted $\mathbb{E}[X]$. The deviation of a variable X from its mean is called *standard deviation* and the squared standard deviation is called the *variance*, denoted by $\mathrm{var}[X]$. The latter equals $\mathbb{E}\left[(X-\mathbb{E}[X])^2\right]$ which can also be written as $\mathbb{E}\left[X^2\right]-\mathbb{E}[X]^2$. The uncertainty associated with a discrete random variable X which measures the amount of information contained by one of its realizations is called *entropy* or *Shannon entropy* and is defined by

$$\mathrm{H}[X] = -\sum_{x\in\mathscr{X}} \mathrm{pmf}_X(x)\cdot\log_2\left(\mathrm{pmf}_X(x)\right) \ .$$

This notion can be extended to continuous random variables by

$$\mathrm{H}[X] = -\int_{\mathscr{X}} \mathrm{pdf}_X(x)\cdot\log_2\left(\mathrm{pdf}_X(x)\right)\mathrm{d}x \ ,$$

and is called *differential entropy*. Indeed it quantifies the expected value of information contained in a specific realization of the random variable and thus can also be expressed as

$$\mathrm{H}[X] = -\mathbb{E}\left[\log_2(\mathrm{P}[X=x])\right] \ .$$

When two random variables are observed, the quantification of some dependency can be done through dedicated statistical tools. That is the probability of an event X knowing an event Y is called the conditional probability of X given Y and is denoted by $\mathrm{P}[X \mid Y]$. In the same manner the conditional expectation of event X given event Y is denoted as $\mathbb{E}[X \mid Y]$. In this context, the law of total expectation is ensued as $\mathbb{E}[X]=\mathbb{E}[\mathbb{E}[X \mid Y]]$. By analogy, the remaining information contained by a realization of a random variable X knowing the realization of a random variable Y is the conditional entropy of X given Y and is denoted by $\mathrm{H}[X \mid Y]$ which satisfies $\mathrm{H}[X \mid Y]=\sum_{y\in\mathscr{Y}}\mathrm{pmf}_Y(y)\mathrm{H}[X \mid Y=y]$ in the discrete case or $\mathrm{H}[X \mid Y]=\int_{\mathscr{Y}}\mathrm{pdf}_Y(y)\mathrm{H}[X \mid Y=y]\mathrm{d}y$ in the continuous case.

To determine if a random variable X came from the same distribution as a random variable Y, one has to test some distribution properties. For instance, the *Difference-of-Mean* test between X and Y refers to the difference $\mathbb{E}[X]-\mathbb{E}[Y]$. It aims to quantify how much the means of X and Y differ. To quantify how much two random variables X and Y change together (*i.e.* have a linear dependency) one can compute the *covariance* between X and Y denoted by $\mathrm{cov}(X,Y)$ which satisfies $\mathrm{cov}(X,Y)=$

$\mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])]$. To measure the strength of such a linear dependency, the covariance can be normalized in the so-called *correlation coefficient* [111] denoted by $\rho(X,Y)$ and defined by:

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

For a more generic dependency, one can measure the information that a random variable Y reveals about a random variable X: the *mutual information* between X and Y, denoted by $\mathrm{I}(X\ ;\ Y)$ and defined by

$$\mathrm{I}(X\ ;\ Y) = \mathrm{H}[X] - \mathrm{H}[X\mid Y] \ . \tag{3.4}$$

It corresponds to the information contained by X minus the remaining information about X knowing Y. Thus, if Y does not bring any information about X, the conditional entropy of X given Y will be equal to the entropy of X which implies a zero mutual information. At the opposite, if Y brings the same information as X, the conditional entropy of X knowing Y will be null and the mutual information equals the entropy of X. In particular, X and Y are $independent$ iff $\mathrm{I}(X\ ;\ Y)=0$. The independence of X and Y implies $\rho(X,Y)=0$ but the converse is false (see Fig. 3-1-4 for an illustration).

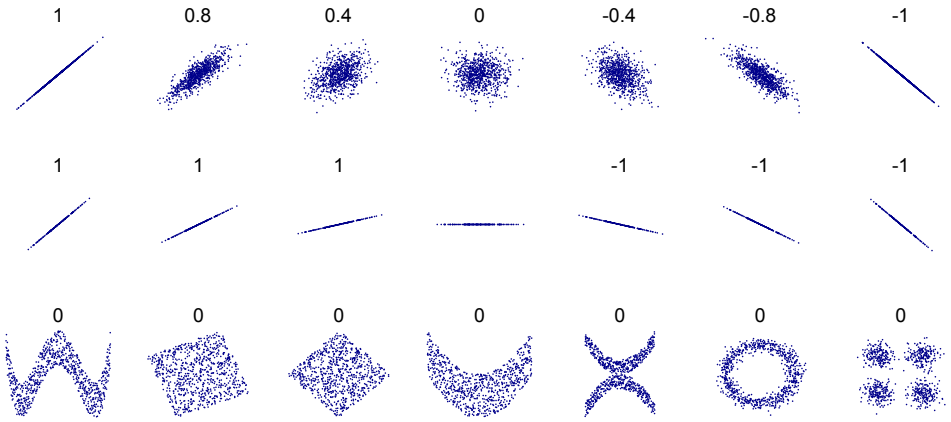


Fig. 3-1-4 – Examples of correlation coefficient for different sets of (x,y) points. The correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope (middle row) nor many aspects of nonlinear relationships (bottom row).

The estimators (*i.e.* experimental evaluations) of the mean, variance, standard deviation and entropy of X based on a sample of observations

are respectively denoted by $\widehat{\mathbb{E}}(X)$, $\widehat{\mathrm{var}}(X)$, $\widehat{\sigma}(X)$ and $\widehat{\mathrm{H}}[X]$. The estimators of the covariance and the correlation between X and Y based on samples of observations are denoted respectively by $\widehat{\mathrm{cov}}(X,Y)$ and $\widehat{\rho}(X,Y)$. Whereas robust estimators exist for the mean, the variance, the standard deviation, the covariance and the correlation, the entropy suffers from the lack of such estimators. More precisely, the estimation of entropy relies on the estimation of the underlying probability mass/density function which is less robust (for a quick review about density estimation, the reader can refer to [93]).

3-2 Basics on Algebra

As modern cryptography relies on the number theory, some basics on algebra are needed for the study and are recalled hereafter. Let $\mathscr{F}$ be a $\mathbb{R}$-vector space of functions defined over a vector space E (*e.g.* $E = \mathbb{F}_2^n$ for some integer n) *i.e.* the set of all real linear combinations of functions $E \mapsto \mathbb{R}$. For a set of d functions $\mathrm{g}_1, \ldots, \mathrm{g}_d$ in $\mathscr{F}$, we shall denote by $<\mathrm{g}_1, \ldots, \mathrm{g}_d>$ the vector space spanned by all the linear combinations of the g_i with coefficients in $\mathbb{R}$. For two functions f and g in $\mathscr{F}$, we call *distance between f and g* and we denote by $\mathrm{d}(f,g)$ the real value defined by:

$$\mathrm{d}(f,g) = \sqrt{\sum_{x \in E} (f(x) - g(x))^2} \quad . \tag{3.5}$$

It corresponds to the *Euclidean distance* between the vectorial representations of f and g.

Remark 2. If the sum in (3.5) is computed over a set of observations (x_i) (instead of E) in a statistical regression context, then $\mathrm{d}(f,g)$ can be interpreted as the square root of a *residual sum of squares* (RSS for short) between the set of variables $f(x_i)$ to be predicted and the predictions $g(x_i)$.

For a function f and a set $\mathscr{G}$, we call *distance between f and $\mathscr{G}$* the real value $\mathrm{d}(f,\mathscr{G})$ defined by:

$$\mathrm{d}(f,\mathscr{G}) = \min_{\substack{g \in \mathscr{G} \\ g \neq 0}} \mathrm{d}(f,g) \quad . \tag{3.6}$$

If $\mathscr{G}$ is the space $< g_1, \ldots, g_d >$, then (3.6) can be rewritten:

$$d(f, \mathscr{G}) = \min_{\substack{(a_1,\ldots,a_d)\in\mathbb{R}^d \\ (a_1,\ldots,a_d)\neq(0,\ldots,0)}} d(f, \sum_{i=1}^{d} a_i g_i) \ . \tag{3.7}$$

By analogy, for two sets $\mathscr{F}$ and $\mathscr{G}$, we call *distance between $\mathscr{F}$ and $\mathscr{G}$* the real value $d(\mathscr{F}, \mathscr{G})$ defined by:

$$d(\mathscr{F}, \mathscr{G}) = \min_{\substack{f\in\mathscr{F} \\ f\neq 0}} d(f, \mathscr{G}) \ . \tag{3.8}$$

3-3 Block Cipher Model

This section describes the modeling of a block-cipher as introduced in Sect. 1-3.1.2. A block cipher is parameterized by a *master key* and it transforms a plaintext block into a ciphertext block through the repetition of key-dependent *round transformations*. We denote by p, and we call *state*, the temporary value taken by the ciphertext during the algorithm. In practice, the cipher is *iterative*, which means that it applies several times the same round transformation φ to the state. This round transformation is parameterized by a *round key k* that is derived from the master key.

In our model, φ is composed of different operations: a key addition layer (by exclusive or), a non-linear layer γ and a linear layer λ:

$$\varphi[k](p) = [\lambda \circ \gamma](p \oplus k) \ .$$

For the sake of simplicity, we assume that the non-linear layer applies the same non-linear transformation S, called *S-box*, on N independent n-bit parts p_i of the state: $\gamma(p) = \big(S(p_1), \ldots, S(p_N)\big)$.
Remark 3. Some block ciphers (*e.g.* DES) have different S-boxes for each part of the state.

For efficiency reasons, the S-box is usually implemented by using Look-Up Table (LUT). The linear layer λ is composed of L linear operations λ_i that operate on L independent l-bit parts $p_{i(l)}$ of the state: $\lambda(p) = \big(\lambda_1(p_{1(l)}), \ldots, \lambda_L(p_{L(l)})\big)$.

We also denote by $l' \leqslant l$ the minimum number of bits of a variable manipulated during the processing of λ_i. For instance, the MixColumn layer of AES applies to columns of size $l = 32$ bits but it manipulates some elements of $l' = 8$ bits only. We further assume that the λ_i are sufficiently similar to be implemented by one *atomic operation* that is an operation which has the same execution flow whatever the index i is.

Remark 4. Linear and non-linear layers may involve different state indexes. In AES for instance, the state is usually represented as a 4×4 matrix of bytes and the non-linear layer usually operates on its elements $p_1, \ldots, p_{16}$ vertically (starting at the top) and from left to right. In this case, the operation λ_1 corresponding to the AES linear layer (that is composed of ShiftRows followed by MixColumns [1]) operates on $p_{1(8)} = (p_1, p_6, p_{11}, p_{16})$.

We shall consider that the key addition and the non-linear layer are merged in a *keyed substitution layer* that adds each key part k_i to the corresponding state part p_i before applying the S-box S.

Part II

Side Channel Attacks

CHAPTER 4

Side Channel Framework

4-1 Introduction

IDE *Channel Analysis* is a cryptanalysis method which uses physical observations leaked by the device during the execution of a given algorithm. Usual observations are timing [50], power consumption [51, 52] and electromagnetic radiations [41, 81]. These *leakages* depend on the performed operations and on the processed data. They can thus bring information about intermediate results. An intermediate result which jointly depends on a part of the secret key and a known value is called *sensitive* and allows an attacker to efficiently recover the secret key.

In the following sections we will first present a brief history of these attacks, then we will introduce a general framework to mount such attacks and we will discuss how to classify and compare them.

4-2 History

The purpose of a side channel attack is to take advantage of the key-dependent physical leakages provided by a cryptographic device, in order to recover secret information (key bytes, typically). Most of these attacks exploit the leakages by comparing them with *key-dependent* models that are available for the target device. Side channel analysis was first motivated by government services after the World War II. The first related side channel analysis appeared in a declassified NSA document [72] which revealed the use of oscilloscope to decipher a teletype encryption in 1943. Another famous example comes from Peter Wright [114] that broke the Egyptian Hagelin cipher machine (a rotor-based machine) by the UK government using microphones (in order to hear rotor manipulations). Subsequently in the early seventies, the *TEMPEST* program was launched by the US government to investigate and study compromising emission. The first academic (*i.e.* public) paper was published by Van Eck [105] in 1985 and the topic is about electromagnetic emissions of video display units. The first academic side channel analysis of a cryptographic implementation was mounted by Paul Kocher and his team in 1996 [50]. It explains how to exploit the computation time of a few executions to break a public key cryptosystem such as RSA. Two years later Kocher *et al.* described a side channel attack based on the power consumption produced by a cryptographic computation [51, 52]. The attack was later extended to electromagnetic radiations [41, 81]. All these attacks are shown to be very efficient in practice to break a large range of cryptosystems such as the widely used DES and RSA.

The emergence of side channel attacks plunged the security and cryptography community into a turbulent domain and a new area of research was created with its dedicated conferences such as *Cryptographic Hardware and Embedded Systems (CHES)*. This branch had a practical impact on all industries involved in embedded security, including the smart card industry. Nowadays, these industries must take into account side channel analysis and security certifications are delivered by independent laboratories in order to guaranty the robustness of final products against side channel attacks. Eventually, since the seminal work of Kocher *et al.* in the late 1990's [52], a large variety of statistical tests, also called distinguishers, have been introduced for this purpose. Their goal was to better take advantage of the available information, *e.g.*, by adapting the statistical test.

4-3 General Framework

Side channel attacks can be classified according to three criteria:

The knowledge of the attacker

- Profiling attacks [21, 31, 88] (*a.k.a. Template attacks*) which correspond to a powerful adversary who controls a copy of the attacked device and uses it to evaluate the distribution of the leakage according to the processed values. Once such an evaluation is obtained, a maximum likelihood approach is carried out to recover the secret data manipulated by the attacked device.

- Non-Profiling attacks (*a.k.a.* standard attacks) which correspond to a common adversary who has a limited access to the attacked device without the ability of learning from prior execution (*i.e.* changing/knowing any key used to build a training database).

The device operation targeted by the attack

- Attacks on the operation flow (*a.k.a. Simple Power Analysis (SPA)* in case of power consumption based attack) which consist in analysing directly – with possible averaging to lower the noise – the observation of an instruction flow (timings, power curves, *etc.*). This allows to retrieve information on a manipulated value (an example is given for instance in Fig. 4-3-1).

- Attacks on the processed data (*a.k.a. Differential Power Analysis (DPA)* in case of power consumption based attack) which consist in targeting an intermediate value depending on a secret value and a known value. In this case the attacker needs different observations for different known values and then apply advanced statistical tests to retrieve the secret value.
  Notation. To avoid ambiguity with the Kocher *et al.* attack named DPA [51], attacks on the processed data are referred by *standard side channel analysis* in the following.

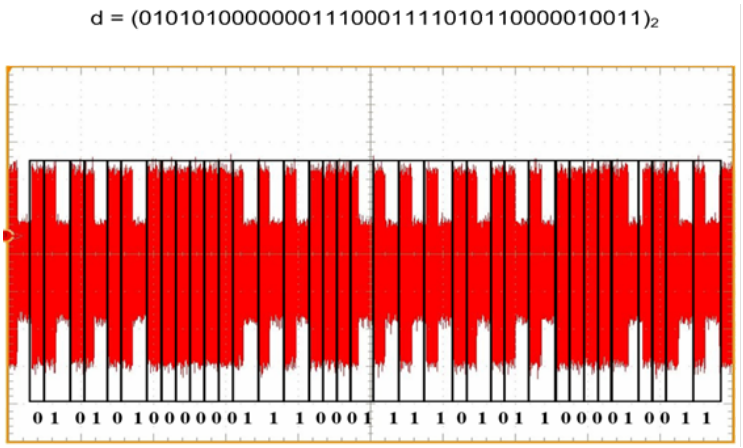


Fig. 4-3-1 – Example of a Simple Power Analysis on an RSA exponentiation using square-and-multiply algorithm: the bits of the exponent can be directly read on the power curve.

The arity of the attack

- *Univariate* attacks if targeting only one leakage point in time.
- Multivariate attacks if targeting $(d+1) > 1$ leakage points $L_0, \ldots, L_d$ (it can be bivariate, trivariate, *etc.*).

Remark 5. A special case of multivariate attacks arises when we assumed that the different leakage points (also called *shares*) are manipulated at the same time (*i.e.* $L_0, \ldots, L_d$ are stacked in time) which can be the case with some hardware configurations. In this case, the multivariate attack can be viewed as an univariate attack as it targets only one time instant. It is the so-called *zero-offset* attack [108].

In theory, profiling attacks are optimally efficient [31]. Nevertheless the adversary needs to be able to carry out a profiling stage on a perfect copy of the target device, which limits the attack pertinence in practice. In fact, it is difficult to have an open access to a copy of the device under attack and, even when it is possible, it remains difficult to exploit templates acquired on one device to attack another one. That implies that template attacks are not generally applied in an industrial context. More precisely, template attacks are usually used as a countermeasure efficiency measurement tool allowing to compare remain-

ing leakages with respect to the noise and thus to have a sketch of the countermeasure efficiency independently of a specific kind of attack. For those reasons, we focus on standard side channel analysis in this thesis (*i.e.* without profiling stage).

Remark 6. A *divide-and-conquer* method is considered in this thesis. That is although standard side channel analysis targets an intermediate value which depends on some parts of the key (*i.e.* subkey), in practice, we perform more global attacks which aim to recover the whole master key (*e.g.* targeting all possible subkeys).

In the following, we consider an adversary who has access to a physical implementation of a cryptographic algorithm and observes the side channel leakage of successive executions with known inputs.

Remark 7. The measurement aspect is not treated in this thesis. We assumed that the attacker is provided with synchronized curves and points of interest location. This is a worst-case scenario from a defender (designer) point of view.

Assumption 1 (Plaintext Uniformity)**.** The known variable sample $(x_i)_i$ is uniformly distributed.

Assumption 1 is a common assumption in side channel analysis as it usually corresponds to a targeted device where the attacker has no control over the processed data. It corresponds to the so called *known plaintext* paradigm where the plaintexts are randomly chosen.

During the cryptographic computation, it is assumed that an intermediate variable Z is manipulated. This intermediate variable is a known function F that combines a known variable X with a secret variable denoted by k (the subkey). The variable $Z = F_k(X)$ is *sensitive* since it depends on both a known value and a secret value. Variables X and k are assumed to be defined over $\mathbb{F}_2^n$ for some integer value n (*e.g.* $n = 8$) and the function $F : X, k \mapsto F_k(X)$ is from $\mathbb{F}_2^{2n}$ into $\mathbb{F}_2^m$ with m such that $m \leq n$ (*e.g.* F is an S-box and $F_k(X) = F(X \oplus k)$). We denote by F_k^{-1} a *reciprocal function* of F_k which maps each image of F_k to its set of preimages.

The analyses conducted in the following are done under the assumption that the leakages satisfy:

$$L = \delta(Z) + B \ , \tag{4.1}$$

where $\delta(\cdot)$ is a deterministic unknown function and the random variable B is Gaussian. Notice that in (4.1) we make the classical assumption

that $\delta(\cdot)$ only depends on the underlying hardware, independently of the time. For the rest of the thesis, the following assumption is made:
Assumption 2 (Independent Noise)**.** The noise B is independent of the targeted variable Z.

Assumption 2 is sound in a smart card context where the leakage is due to charge carriers within conductors [70]. When measuring, leakages usually contain an additional noise. This noise can come from an external source coupled with the device (*external noise*), from internal movements within conductors (*intrinsic noise*), from imperfection of measurement tools (particularly from Analog-to-Digital converters – *quantization noise*), or from variation of the data processed by the algorithm (*algorithmic noise*).

In the following we focus on *univariate* attacks. The multivariate case is treated in Chap. 7.

To mount an attack, the adversary measures leakages $(\ell^i)_i \hookleftarrow L$ from the targeted device using a sample $(x_i)_i \hookleftarrow X$ of plaintexts. Then, he computes the hypothetic value $F_{\hat{k}}(x_i)$ of the sensitive variable $F_k(x_i)$ for every x_i and for every possible $\hat{k}$. A *leakage model function* m is subsequently applied to map the hypothetic sensitive values toward estimated leakage values $m_{\hat{k},i} = \mathsf{m}(F_{\hat{k}}(x_i))$. Eventually, the adversary uses a distinguisher to compare the different model samples $(m_{\hat{k},i})_i \hookleftarrow M_{\hat{k}} = \mathsf{m}(F_{\hat{k}}(X))$ with the actual leakage sample $(\ell^i)_i$. If the attack is successful, the best comparison result (*i.e.* the highest – or lowest – value of the distinguisher) should be obtained for the model sample corresponding to the correct subkey candidate $\hat{k} = k$. This procedure can then be repeated for different subkeys in order to eventually recover the full master key.

We sum-up hereafter the different steps of a standard SCA (also recalled in Fig. 4-3-2):

1. Perform N measurements $(\ell^i)_i \hookleftarrow L$ on the cryptographic device using a sample $(x_i)_i \hookleftarrow X$ of plaintexts ;
2. Choose a function m to model the deterministic part of the leakage ;
3. For every key hypothesis $\hat{k}$, compute the model values $m_{\hat{k},i}$ from the plaintexts x_i and the model function m ;
4. Choose a statistical distinguisher Δ.

5. For every key hypothesis $\hat{k}$, compute the *distinguishing value* $\Delta_{\hat{k}}$ defined by:
$$\Delta_{\hat{k}} = \Delta\left((\ell^i)_i, (m_{\hat{k},i})_i\right) \ .$$
This results in a *score vector* $(\Delta_{\hat{k}})_{\hat{k}}$;
6. Output as the o most likely key candidates the o key hypotheses that maximize – or minimize – $\Delta_{\hat{k}}$.

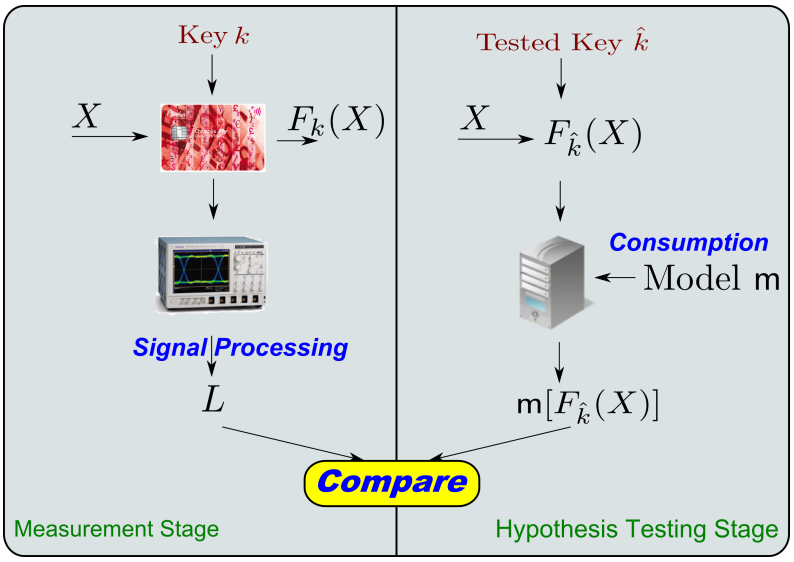


Fig. 4-3-2 – Main steps of a standard SCA

Notation. As it can be seen in the above list, a standard SCA on a given sensitive variable $Z = F_k(X)$ is only characterized by the model function m and the distinguisher Δ. For this reason we shall use in the following the notation (m, Δ)-*SCA* to differentiate one such an attack from another.

From the general attack description recalled above it is clear that two major choices are left to the adversary to perform a standard SCA attack on a given sensitive variable computed on some device:

- the choice of the distinguisher,
- the choice of the model.

In the following, we study the impact of both such choices in a SCA attack. We will first show that most of univariate SCA distinguishers that have been proposed in the literature are equivalent – under some conditions – to a same distinguisher. Namely, they lead to similar results up to a change of model. We will then discuss the importance of the model

for the attack soundness (*i.e.* the theoretically establishment of the attack) and we will investigate attacks that do not require any *a priori* choice of a model.

4-4 Main Univariate Side Channel Attacks Description

In the following, we describe the main univariate attacks targeting a unique time instant with leakage L. In this context, (4.1) became

$$L = \delta(Z) + B \ . \tag{4.2}$$

The first (m, Δ)-SCA introduced by Kocher *et al.* in [52] targets a single bit of the sensitive variable Z and shall be therefore referred to as *single-bit DPA* in the rest of the thesis. Since this bit usually depends on all bits of the subkey, the single-bit DPA may allow to unambiguously discriminate the correct subkey. However, for some kinds of algebraic relationships between the manipulated data and the subkey, several key candidates (including the correct one) may result in the same distinguishing value and the attack fails (this phenomenon is referred to as *ghost peaks* in [27]). To exploit more information from the leakage related to the manipulation of Z and to succeed when single-bit DPA does not, the attack was extended to several bits by Messerges in [67] in two ways: the *all-or-nothing DPA* and the *generalized DPA*. The original single-bit DPA of Kocher and its extensions by Messerges can all be defined in a similar way as follows:
Definition 1 (Differential Power Analysis – DPA –)**.** A *DPA* is a (m, Δ)-SCA, which involves a distinguisher Δ defined as a Difference of Means (DoM) between two leakage partitions defined according to the image set Im(m).

Depending on the definition of the leakage model function m, we connect the three DPA attacks listed above with Definition 1:

- In a *single-bit DPA*, the image set Im(m) is reduced to two elements w_0 and w_1 and for every $\hat{k}$ we have:

$$\Delta_{\hat{k}} = \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = w_0\right) - \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = w_1\right) \ . \tag{4.3}$$

- In an *all-or-nothing DPA*, the image set Im(m) can have a cardinality greater than 2. Two elements ω_0 and ω_1 are chosen in Im(m) and for every $\hat{k}$ we have:

$$\Delta_{\hat{k}} = \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega_0\right) - \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega_1\right) \ . \tag{4.4}$$

- In a *generalized DPA*, two subsets Ω_0 and Ω_1 of Im(m) are chosen and for every $\hat{k}$ we have:

$$\Delta_{\hat{k}} = \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} \in \Omega_0\right) - \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} \in \Omega_1\right) \ . \tag{4.5}$$

Notation. Distinguishers $\Delta_{\hat{k}}$ defined in (4.3) - (4.5) shall be denoted by SB-DPA($\hat{k}$), AON-DPA($\hat{k}$) and G-DPA($\hat{k}$) respectively, where $\hat{k}$ is the key hypothesis.
Example 1. Typical choices for the model functions in (4.3) - (4.5) are written hereafter. They are taken from the original papers [52] and [67]:

- *Single-bit DPA:* m is the function that maps $F_{\hat{k}}(x)$ to one of its bit-coordinates and we hence have $\text{Im}(\mathsf{m}) = \{\omega_0, \omega_1\} = \{0, 1\}$.
- *All-or-nothing DPA:* m is the Hamming weight and thus we have $\{\omega_0, \omega_1\} = \{0, n\}$ (n being the bit-size of $F_{\hat{k}}(x)$).
- *Generalized DPA:* m is the Hamming weight and thus we have $\{\Omega_0, \Omega_1\} = \left\{\{1, \ldots, \lfloor \frac{n}{2} \rfloor\}, \{\lceil \frac{n}{2} \rceil, \ldots, n\}\right\}$.

However, different choices for m, (ω_0, ω_1) and (Ω_0, Ω_1) may be arbitrary made by the attacker, hence we do not fix a particular choice in the following.

After Messerges' works, some other extensions of the DPA have been proposed respectively by Le *et al.* in [56], by Standaert *et al.* in [97] (and also Maghrebi *et al.* in [60]) and by Brier *et al.* in [27].

The generalization proposed in [56] starts from (4.5) and enables to involve more than 2 subsets to eventually compute a weighted sum of means instead of a simple DoM. We recall hereafter its definition:
Definition 2 (Partition Power Analysis – PPA –)**.** A *PPA* is a (m, Δ)-SCA, which involves a distinguisher Δ defined for every $\hat{k}$ by:

$$\Delta_{\hat{k}} = \sum_{\omega_i \in \text{Im}(\mathsf{m})} \alpha_i \cdot \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega_i\right) \ , \tag{4.6}$$

where the α_i are constant coefficients in $\mathbb{R}$.

Notation. A distinguisher $\Delta_{\hat{k}}$ defined such as in (4.6) shall be denoted $\mathsf{PPA}_{(\alpha_i)_i}(\hat{k})$. Moreover, when we shall need to exhibit the model m in the PPA, we shall use the notation $\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k})$ for the distinguisher.

As discussed in [56], the tricky part when specifying a PPA attack is the choice of the most suitable coefficients α_i. We will show in this section that this choice is naturally equivalent to a characterization of the leakage function δ in (4.2).

The generalization proposed in [97] (and also [60]) starts from (4.6) and proposes to use a weighted variance instead of a weighted mean. We recall hereafter its definition:
Definition 3 (Variation Power Analysis – VPA –)**.** A *VPA* is a (m, Δ)-SCA, which involves a distinguisher Δ defined for every $\hat{k}$ by:

$$\Delta_{\hat{k}} = \sum_{\omega_i \in \mathrm{Im}(\mathsf{m})} \alpha_i \cdot \widehat{\mathsf{var}}\left(L \mid M_{\hat{k}} = \omega_i\right) \ , \tag{4.7}$$

where the α_i are constant coefficients in $\mathbb{R}$.
Notation. A distinguisher $\Delta_{\hat{k}}$ defined such as in (4.7) shall be denoted $\mathsf{VPA}_{(\alpha_i)_i}(\hat{k})$. Moreover, when we shall need to exhibit the model m in the VPA, we shall use the notation $\mathsf{VPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k})$ for the distinguisher.

The generalization of the DPA proposed in [26] involves the linear correlation coefficient. We recall hereafter the definition of this attack:
Definition 4 (Correlation Power Analysis – CPA –)**.** A *CPA* is a (m, Δ)-SCA, which involves Pearson's correlation coefficient ρ as distinguisher. Namely, for every $\hat{k}$, we have:

$$\Delta_{\hat{k}} = \widehat{\rho}\left(L\,, M_{\hat{k}}\right) = \frac{\widehat{\mathrm{cov}}\left(L\,, M_{\hat{k}}\right)}{\widehat{\sigma}(L) \cdot \widehat{\sigma}\left(M_{\hat{k}}\right)} \ , \tag{4.8}$$

where $\widehat{\sigma}(L)$ and $\widehat{\sigma}\left(M_{\hat{k}}\right)$ denote the standard deviations of the samples $\ell_1^i \hookleftarrow L$ and $m_{\hat{k},i} \hookleftarrow M_{\hat{k}}$ respectively and where their covariance is denoted by $\widehat{\mathrm{cov}}\left(L\,, M_{\hat{k}}\right)$ which is $\widehat{\mathbb{E}}\left(L M_{\hat{k}}\right) - \widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(M_{\hat{k}}\right)$.
Notation. A distinguisher $\Delta_{\hat{k}}$ defined such as in (4.8) shall be denoted by $\mathsf{CPA}(\hat{k})$. Moreover, when we shall need to exhibit the model m used in the CPA, we shall use the notation $\mathsf{CPA}_{\mathsf{m}}(\hat{k})$ for the distinguisher.

Another kind of distinguisher, the mutual information which is based on a general dependency, not only a linear dependency, was proposed by Gierlichs *et al.* [42]. We recall hereafter its definition:

Definition 5 (Mutual Information Analysis – MIA –)**.** An *MIA* is a (m, Δ)-SCA, which involves the mutual information I as distinguisher. Namely, for every $\hat{k}$, we have:

$$\Delta_{\hat{k}} = \mathrm{I}(L \ ; \ M_{\hat{k}}) = \mathrm{H}[L] - \mathrm{H}[L \mid M_{\hat{k}}] \ . \tag{4.9}$$

Notation. A distinguisher $\Delta_{\hat{k}}$ defined such as in (4.9) shall be denoted by $\mathsf{MIA}(\hat{k})$. Moreover, when we shall need to exhibit the model m used in the MIA, we shall use the notation $\mathsf{MIA}_{\mathsf{m}}(\hat{k})$ for the distinguisher.

The attacks listed above (except VPA) have been applied in many papers, *e.g.* [35,44,67], and have even been sometimes experimentally compared one to another [55, 65, 97]. However, none of these works have enabled to draw definitive conclusions about the similarities and the differences of the attacks. Next chapters aim to overcome this lack.

Remark 8. The notion of weighted sum has been used with expectation and variance nevertheless this notion can be extended to other quantities such as entropy as in the so-called *Entropy Power Analysis – EPA –* introduced by Maghrebi *et al.* in [61]. The latter is not analysed in this manuscript as it is linked with mutual information which is not studied in this thesis.

Some other attacks are noticeable but they target an MIA-like distinguisher (such as EPA). For instance Le *et al.* [54] estimate the mutual information using higher-order cumulant method. Whitnall *et al.* [113] introduce the Kolmogorov-Smirnov test as a descriminant whereas Lyu *et al.* [59] introduce the partial Kolmogorov-Smirnov test. The link between these attacks and MIA remains to be analysed but it is not the purpose of this thesis.

4-5 Taxonomy

The SCA comparisons made during this thesis (and detailed in the next chapters) enable to draw a sketch of a classification tree (Fig. 4-5-1) and to thus clarify the non-profiled univariate side channel attacks zoology.

As we have focused on non-profiled SCA, only this branch is detailed. We denote by *"combining" univariate SCA*, a multivariate attack where leakages are preprocessed by a combination function to allow the application of an univariate SCA as explained in Sect. 7-1.2.

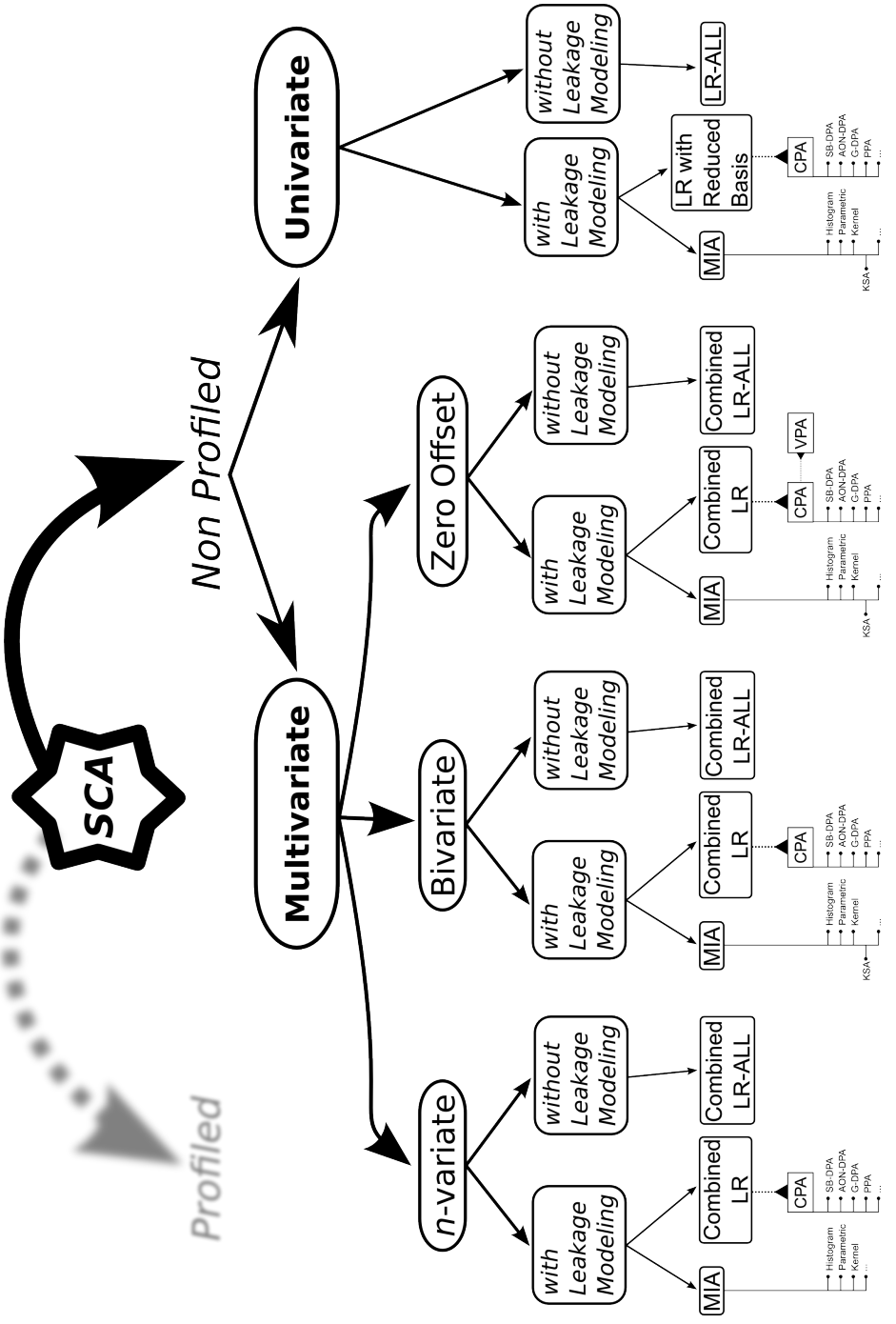


Fig. 4-5-1 – Taxonomy of non profiled SCA

4-6 Efficiency of Side Channel Attacks

The fair evaluation and comparison of side channel attacks need sound tools to measure and to quantify their efficiency. In [98], Standaert *et al.* proposed a framework based on two metrics. The first one is the *success rate* of an attack, defined as the probability of the right key to be ranked at the first place in the score vector. By extension an o^{th} order success rate is the probability of the good key to be ranked among the o^{th} first places in the score vector. The second one is the *guessing entropy* of an attack, defined as the expectation of the rank of the good key. The success rate is adapted in an attacker point of view with a fixed workload as it directly quantifies the success of an attack. At the opposite, the guessing entropy better fits with a designer point of view as it quantifies the workload needed for an efficient attack. Nevertheless both values need empirical measures to be estimated and thus quickly need an huge computational effort. Some works (for instance [83]) deal with this problem and propose solutions with the addition of more restrictive assumptions. Nevertheless these values quantify the attacks at one point (with for instance a fixed noise standard deviation) and do not permit to draw conclusions about the behavior of the attacks in a slightly different context. To overcome this drawback, some other quantities have been already mentioned by Messerges [67] and to a lesser extent by Prouff [78]. They have been formalized by Oswald *et al.* in [112]. In particular, Oswald *et al.* formalized the *relative distinguishing margin* which measures the – normalized – distance between the correct key distinguisher value and the value for the highest ranked alternative. We can expect that higher this distance, lower the noise sensitivity. This measure is sound only when the good key is ranked first. [112] also formalizes the *absolute distinguishing margin* which aims to measure the efficiency loss from the optimal context (*e.g.* with no noise) to a practical context. Theses metrics in fact permit to quantify some properties of an attack and depending on the context (attacker, chip designer *etc.*) one can favor one property over the others.

A pecular work of Mangard [63], dedicated to a specific distinguisher (the correlation coefficient), uses the Fisher transformation to evaluate its efficiency. It permits to deduce directly from the distinguishing value, the success rate of an attack w.r.t. the noise standard deviation.

In this thesis we will mainly uses the notion of success rate (*a.k.a.* the

number of messages needed to achieve some success rate) as it reflects the useful information from an attacker point of view (*i.e.* it corresponds to the minimal effort an attacker must do to have a probability p of success). Moreover, we notice that in [62], Mangard *et al.* show that the correlation for the good key is a sound estimator of the efficiency of the CPA. The mutual information for the good key is also a sound estimator of the efficiency of the MIA [99, 106]. Therefore both CPA and MIA distinguisher values are directly related to the success rate values even if the distinguishing value for the good key depends on the given leakage model w.r.t. the leakage function. Thus CPA (respectively MIA) can evaluate an attack in a very particular context for instance when the error made by the leakage model on the leakage function is exactly known.

It must be noticed in case of an attack using a – given or computed – leakage model (such as CPA) that the closeness of the leakage model to the real leakage function is not directly linked to the efficiency of the attack. That is, a more accurate leakage model does not imply a more efficient underlying attack. In fact the leakage model must be closer to the real leakage function for the good key hypothesis than for wrong hypothesis. It is usually the case when the leakage model is not parametrized by the key hypothesis and is close to the real function (for instance in CPA). In this case, the closest is the leakage model, more efficient is the attack. At the opposite, if the leakage model is parametrized by the key hypothesis (for instance in a linear regression attack) a model instantiated with the wrong hypothesis can be closer to the real leakage function than the model instantiated with the good key. A good example of the latter is the linear regression with full basis as explained in Sect. 7-3.

4-7 Notion of SCA-equivalency

The study shall be conducted under the following assumption that is added to Assumption 2 introduced in Sect. 4-3:
Assumption 3 (Target Uniformity)**.** Under Assumption 1, the predicted variable sample $(F_{\hat{k}}(x_i))_i$ is balanced for every key hypothesis $\hat{k}$.
Remark 9. Assumption 3 is realistic in the SCA context. Indeed, the $(F_{\hat{k}}(x_i))_i$ result from the evaluation of a *balanced* cryptographic primitive (*e.g.* an S-box or a linear operation over a small vector space), and we

can fairly assume when the sample size N is large enough that $(F_{\hat{k}}(x_i))_i$ is a balanced sample.

Remark 10. Since m is defined over the definition set of the values $F_{\hat{k}}(x_i)$ and since the distribution over $(F_{\hat{k}}(x_i))_i$ is balanced whatever $\hat{k}$, Assumption 3 implies that the mean and the standard deviation of $M_{\hat{k}} = \mathsf{m}(F_{\hat{k}}(X))$ are always estimated from a balanced sample. As a consequence, those estimations are constant with respect to the key hypothesis $\hat{k}$ and correspond exactly to the mean $\mathbb{E}\left[M_{\hat{k}}\right]$ and the standard deviation $\sigma\left(M_{\hat{k}}\right)$ of $M_{\hat{k}}$.

Remark 11. Assumption 3 can be intentionally relaxed by using non-uniform distributed plaintexts. Usually this is done using a – possibly *adaptive – chosen plaintext* paradigm which permits to bias the distribution of $(F_{\hat{k}}(x_i))_i$ (see [107] for more details).

Remark 12. In some hardware context the deterministic part of the leakage (*i.e.* the function δ) can be key-dependent as shown in [45] even if Assumption 1 is fulfilled and $(F_{\hat{k}}(x_i))_i$ is balanced for every key hypothesis. Since the function m is not parameterized by the key hypothesis, in such a setting some keys can be more easily recovered than others (see [47]).

Under these assumptions, we aim to compare different distinguishers targeting the same intermediate variable. For this purpose, we introduce hereafter the notion of *reduction between two SCAs*:

Definition 6 (SCA-reduction)**.** A (m, Δ)-SCA is said to be *SCA-reducible* to a $\left(\mathsf{m}', \Delta'\right)$-SCA if there exists a function f such that $\mathsf{m} = \mathsf{f} \circ \mathsf{m}'$ and for every key $\hat{k}$ and every sample $(\ell_1^i)_i$ and $(x_i)_i$, there exists a strictly monotonous function g such that:

$$\Delta\left((\ell_1^i)_i, (m_{\hat{k},i})_i\right) = \mathsf{g} \circ \Delta'\left((\ell_1^i)_i, (m'_{\hat{k},i})_i\right) \ ,$$

where $m_{\hat{k},i} = \mathsf{m}(F_{\hat{k}}(x_i))$ and $m'_{\hat{k},i} = \mathsf{m}'(F_{\hat{k}}(x_i))$.

This definition implies that one attack is SCA-reducible to another, if and only if the first one ranks the key in the same (or reverse) order as the second one does. That is, either the success rate or the guessing entropy remains unchanged by the transformation. If one attack is SCA-reducible to another, it does not imply that the second one is SCA-reducible to the first one. If the case arises, we will use the notion of *SCA-equivalence*.

Definition 7 (SCA-equivalence)**.** Let A be a (m, Δ)-SCA and let B be a $\left(\mathsf{m}', \Delta'\right)$-SCA. A is said to be *SCA-equivalent* to B if and only if A is

SCA-reducible to B and B is SCA-reducible to A.

Remark 13. This definition extends the one in [62] in a non-asymptotic context and thus must deal with estimation problematic. It implies that this definition is estimation-dependent. Moreover, whereas [62] is about equivalent efficiency, this definition is about effectiveness for a given fixed sample.

In what follows, we establish the SCA-reductions from DPA to PPA and from PPA to CPA. We show that each of those attacks can be reformulated to reveal a correlation coefficient computation and that they only differ in the involved model function. A direct consequence of this result is that comparing those attacks simply amounts to compare the accuracy of the underlying models. Afterward, we analyse the special case of VPA which involves variance computation instead of mean and we show that it is in fact a zero-offset CPA. We also address attacks that consist in summing distinguishers and we show that they are also SCA-reducible to a CPA. These results emphasize the importance of making a good choice for the model according to the attack context specificities, which is eventually discussed in Sect. 5-7.

CHAPTER 5

Univariate Side Channel Analysis and Linear Correlation

5-1 Introduction

In view of the large variety of distinguishers available in the literature, a natural question is to determine the exact relations between them and the conditions upon which one of them would be more efficient. Closely related to this question, Mangard *et al.* showed in [65] that for a category of attacks, denoted as *standard univariate SCA*, a number of distinguishers (namely, those using a Difference-of-Means test or a Pearson's correlation coefficient or Gaussian templates) are in fact asymptotically equivalent given that they are provided with the same *a priori* information about the leakages (*i.e.* if they use the same model). More precisely, [65] shows that these distinguishers only differ in terms that become asymptotically key-independent once properly estimated. While this result is limited to

first-order (a.k.a. standard univariate) SCAs, it clearly underlines that the selection (or construction) of a proper leakage model in SCA is at least as important as the selection of a good distinguisher.

A natural extension of Mangard *et al.*'s work is to study whether their statement holds in non-asymptotic contexts (*i.e.* when the number of measurements is reasonably small). Such a study is of particular importance since it corresponds to a practical issue from both the attacker and the security designer side. Indeed the latter ones often need to precisely determine which of the numerous existing attacks is the most suitable one in a given context, or reciprocally, which context is the most appropriate one for a given attack.

The results in this chapter can be seen as a complement to the state-of-the-art analysis. We focus on the main used non-profiled side channel distinguishers (see Sect. 4-4). We prove that they are not only asymptotically equivalent but also, that they can be explicitly re-written one in function of another, by only changing the leakage model. In other words, we show that all these distinguishers exploit essentially the same statistics and that any difference can be expressed as a change of model. This provides us with a unified framework to study and to compare the attacks. Moreover, it emphasizes how strong is the impact of the model choice on the attack efficiency, not only in an asymptotic contex but also in contexts with limited sample sizes.

5-2 From DPA to PPA

As the PPA is a generalization of the DPA, based on the same statistical tool (namely a DoM test), we can reasonably expect that all the DPA presented in Section 4-4 can be rewritten in terms of a PPA. We give in the following a formal proof of this intuition. Note that our proof is constructive and it exhibits how to reformulate any DPA in terms of a PPA.

Proposition 1. Let $\mathsf{DPA}(\hat{k})$ be one of the DPA defined in (4.3) - (4.5). There exist coefficients $(\alpha_i)_i$ such that $\mathsf{DPA}(\hat{k}) = \mathsf{PPA}_{(\alpha_i)_i}(\hat{k})$.

Proof. Let us first focus on the $\mathsf{SB\text{-}DPA}(\hat{k})$ distinguisher and let us denote by α_0 and α_1 respectively the coefficients 1 and -1. Relation

(4.3) can be rewritten:

$$\text{SB-DPA}(\hat{k}) = \alpha_0 \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = w_0\right) + \alpha_1 \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = w_1\right) \ . \tag{5.1}$$

The right part of (5.1) defines a PPA distinguisher $\text{PPA}(\hat{k})$ involving the same 2-valued model m as $\text{SB-DPA}(\hat{k})$ and the pair of coefficients (α_0, α_1). The same reasoning holds for an all-or-nothing DPA and its distinguisher $\text{AON-DPA}(\hat{k})$ defined in (4.4), by stating $\alpha_0 = 1$, $\alpha_1 = -1$ and $\alpha_i = 0$ for every $\omega_i \in \text{Im(m)} \backslash \{\omega_0, \omega_1\}$.

Let us now focus on the generalized DPA distinguisher $\text{G-DPA}(\hat{k})$. It can be easily checked that (4.5) can be rewritten:

$$\begin{aligned}\text{G-DPA}(\hat{k}) = \sum_{\omega \in \Omega_0} \frac{\widehat{\text{P}}(M_{\hat{k}}=\omega)}{\widehat{\text{P}}(M_{\hat{k}} \in \Omega_0)} \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega\right) - \sum_{\omega \in \Omega_1} \frac{\widehat{\text{P}}(M_{\hat{k}}=\omega)}{\widehat{\text{P}}(M_{\hat{k}} \in \Omega_1)} \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega\right) \\ + \sum_{\omega \in \text{Im(m)} \backslash \Omega_0 \cup \Omega_1} 0 \cdot \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega\right) \ . \end{aligned} \tag{5.2}$$

Let us denote by $(\omega_i)_i$ the elements in Im(m) and let $(\alpha_i)_i$ be a family of coefficients defined such that:

$$\alpha_i = \begin{cases} \frac{\widehat{\text{P}}(M_{\hat{k}}=\omega_i)}{\widehat{\text{P}}(M_{\hat{k}} \in \Omega_0)} & \text{if } \omega_i \in \Omega_0, \\ -\frac{\widehat{\text{P}}(M_{\hat{k}}=\omega_i)}{\widehat{\text{P}}(M_{\hat{k}} \in \Omega_1)} & \text{if } \omega_i \in \Omega_1, \\ 0 & \text{otherwise.} \end{cases}$$

Under Assumption 3, coefficients α_i are constant (namely independent of the sample size and of the key hypothesis). After replacing the coefficients in (5.2) by those α_i, we recognize in (5.2) the definition of a PPA distinguisher involving the same model m as $\text{G-DPA}(\hat{k})$ and the family $(\alpha_i)_i$ as coefficients. ◇

As a direct consequence of Proposition 1, we get the following corollary:

Corollary 1. Under Assumption 3, a DPA is SCA-reducible to a PPA.

In the next section, we compare the PPA with the CPA.

5-3 From PPA to CPA

It is already well known in statistics that a linear correlation coefficient can be written as a weighted sum of means over a partition of a proba-

bility space. As a straightforward consequence and as mentioned by Le *et al.* in [56], a CPA can be viewed as a particular case of a PPA (*i.e.* a CPA is SCA-reducible to a PPA). What we prove in this section is that a PPA can be conversely re-stated as a CPA. Eventually, we argue that both attacks are SCA-equivalent under Assumption 3.

Proposition 2. Let $\mathsf{PPA}_{(\alpha_i)_i}(\hat{k})$ be a PPA distinguisher defined with respect to a model function m and a family of coefficients $(\alpha_i)_i$. Then, there exists a function f and two constant coefficients a and b such that $\mathsf{PPA}_{(\alpha_i)_i}(\hat{k}) = a \cdot \mathsf{CPA}(\hat{k}) + b$, where $\mathsf{CPA}(\hat{k})$ is a CPA distinguisher involving the model function $\mathsf{f} \circ \mathsf{m}$.

Proof. We recall that, in the definition of $\mathsf{PPA}_{(\alpha_i)_i}(\hat{k})$ (see (4.6)), every $\omega_i \in \mathrm{Im}(\mathsf{m})$ is associated with the coefficient α_i. From those ω_i and α_i we define a function f on $\mathrm{Im}(\mathsf{m})$ by:

$$\mathsf{f}(\omega_i) = \frac{\alpha_i}{\widehat{\mathrm{P}}\left(M_{\hat{k}} = \omega_i\right)} \quad . \tag{5.3}$$

Under Assumption 3, probabilities $\widehat{\mathrm{P}}\left(M_{\hat{k}} = \omega_i\right)$ and thus coefficients $\mathsf{f}(\omega_i)$ are constant (namely independent of the sample size and of the key hypothesis $\hat{k}$). With those new notations, (4.6) can be rewritten as:

$$\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \sum_{\omega_i \in \mathrm{Im}(\mathsf{m})} \mathsf{f}(\omega_i) \cdot \widehat{\mathrm{P}}\left(M_{\hat{k}} = \omega_i\right) \cdot \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} = \omega_i\right) \quad . \tag{5.4}$$

We therefore get the following relation:

$$\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \sum_{\alpha \in \mathrm{Im}(\mathsf{f})} \alpha \cdot \widehat{\mathrm{P}}\left(M_{\hat{k}} \in \mathsf{f}^{-1}(\alpha)\right) \cdot \widehat{\mathbb{E}}\left(L \mid M_{\hat{k}} \in \mathsf{f}^{-1}(\alpha)\right) \tag{5.5}$$

i.e.

$$\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \sum_{\alpha \in \mathrm{Im}(\mathsf{f})} \widehat{\mathrm{P}}\left(\mathsf{f}(M_{\hat{k}}) = \alpha\right) \cdot \widehat{\mathbb{E}}\left(\alpha \cdot L \mid \mathsf{f}(M_{\hat{k}}) = \alpha\right) \quad . \tag{5.6}$$

After denoting by $M'_{\hat{k}}$ the random variable $\mathsf{f}(M_{\hat{k}})$ and thanks to the law of total expectation, we eventually deduce:

$$\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \widehat{\mathbb{E}}\left(L M'_{\hat{k}}\right) \quad . \tag{5.7}$$

On the other hand, we have:

$$\mathsf{CPA}_{\mathsf{m}'}(\hat{k}) = \frac{1}{\widehat{\sigma}(L)\widehat{\sigma}\left(M'_{\hat{k}}\right)} \cdot \widehat{\mathbb{E}}\left(L M'_{\hat{k}}\right) - \frac{\widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(M'_{\hat{k}}\right)}{\widehat{\sigma}(L)\widehat{\sigma}\left(M'_{\hat{k}}\right)} \quad ,$$

where m' denotes the function $\mathsf{f} \circ \mathsf{m}$. Under Assumption 3, values $\widehat{\mathbb{E}}(L)$, $\widehat{\sigma}(L)$, $\widehat{\mathbb{E}}\left(M_{\hat{k}}\right)$ and $\widehat{\sigma}\left(M_{\hat{k}}\right)$ are constant with respect to $\hat{k}$. This implies that the CPA distinguisher $\mathsf{CPA}(\hat{k})$ associated with the model function $\mathsf{f} \circ \mathsf{m}$ satisfies the following equality:

$$\widehat{\mathbb{E}}\left(L M'_{\hat{k}}\right) = a \cdot \mathsf{CPA}_{\mathsf{m}'}(\hat{k}) + b \ , \tag{5.8}$$

where a and b are two constant values satisfying

$$a = \widehat{\sigma}(L)\widehat{\sigma}\left(M'_{\hat{k}}\right) \ \text{ and } \ b = \frac{\widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(M'_{\hat{k}}\right)}{\widehat{\sigma}(L)\widehat{\sigma}\left(M'_{\hat{k}}\right)} \ .$$

From (5.7) and (5.8) we deduce that there exist two constant terms a and b and a model transformation f such that

$$\mathsf{PPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = a \cdot \mathsf{CPA}_{\mathsf{m}'}(\hat{k}) + b \ , \tag{5.9}$$

with $\mathsf{m}' = \mathsf{f} \circ \mathsf{m}$. ◇

As a straightforward consequence of Proposition 2 we get the following corollary:

Corollary 2. Under Assumption 3, a PPA is SCA-equivalent to a CPA.

Proposition 2 implies that a PPA and a CPA only differ in the model which is involved to correlate the leakage signal. As a consequence, if a PPA with model m and coefficients α_i is more efficient than a CPA with model m', this simply means that the model $\mathsf{f} \circ \mathsf{m}$ (for f defined as in the proof of Proposition 2) is more linearly related to the deterministic leakage function $\delta(\cdot)$ than m' does. In such a case, the CPA must be performed with the most accurate model between both, namely $\mathsf{f} \circ \mathsf{m}$. In other terms, we have:

Fact. The problem of finding the most pertinent coefficients α_i is equivalent to the problem of finding the model with maximum linear correlation with the deterministic leakage function.

5-4 A (not so) Special Case: VPA

Due to the similarities with PPA definition, we can expect to have a similar rewritting of the VPA in terms of CPA. We prove that VPA is in fact a bivariate zero-offset product centered combining correlation power

analysis. In other words, VPA can be restated as a CPA targeting the square of a centered leakage.

Proposition 3. Let $\mathrm{VPA}_{(\alpha_i)_i}(\hat{k})$ be a VPA distinguisher defined with respect to a model function m and a family of coefficients $(\alpha_i)_i$. Then, there exist a function f and two constant coefficients a and b such that $\mathrm{VPA}_{(\alpha_i)_i}(\hat{k}) = a \cdot \mathrm{CPA}(\hat{k}) + b$, where $\mathrm{CPA}(\hat{k})$ is a CPA distinguisher involving the model function $\mathsf{f} \circ \mathsf{m}$ and applied to the square of the centered leakage $\left(L - \widehat{\mathbb{E}}(L)\right)^2$.

Proof. VPA can be rewritten in the same manner as PPA. That is, we can define a function f such that $\mathsf{f}(\omega_i) = \frac{\alpha_i}{\widehat{\mathrm{P}}(M_{\hat{k}} = \omega_i)}$. Then using the same rewritten procedure from (5.4) to (5.6) we obtain the following relation:

$$\mathrm{VPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \sum_{\alpha \in \mathrm{Im}(\mathsf{f})} \widehat{\mathrm{P}}\left(\mathsf{f}(M_{\hat{k}}) = \alpha\right) \cdot \alpha \cdot \widehat{\mathrm{var}}\left(L \mid \mathsf{f}(M_{\hat{k}}) = \alpha\right) \ , \tag{5.10}$$

which is equivalent to:

$$\mathrm{VPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \sum_{\alpha \in \mathrm{Im}(\mathsf{f})} \widehat{\mathrm{P}}\left(\mathsf{f}(M_{\hat{k}}) = \alpha\right) \cdot \alpha \cdot \widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2 \mid \mathsf{f}(M_{\hat{k}}) = \alpha\right) \ . \tag{5.11}$$

With the same trick – as in PPA rewriting – of renaming the random variable $\mathsf{f}(M_{\hat{k}})$ by $M'_{\hat{k}}$ and thanks to the law of total expectation, we eventually deduce:

$$\mathrm{VPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = \widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2 M'_{\hat{k}}\right) \ . \tag{5.12}$$

On the other hand, in a bivariate zero-offset product centered combination setting we have:

$$\mathrm{CPA}_{2,\mathsf{m}'}(\hat{k}) = \frac{\widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2 M'_{\hat{k}}\right)}{\widehat{\sigma}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\sigma}\left(M'_{\hat{k}}\right)} - \frac{\widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\mathbb{E}}\left(M'_{\hat{k}}\right)}{\widehat{\sigma}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\sigma}\left(M'_{\hat{k}}\right)} \ ,$$

where m' denotes the function $\mathsf{f} \circ \mathsf{m}$.

Under Assumption 3, values $\widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)$, $\widehat{\sigma}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)$, $\widehat{\mathbb{E}}\left(M_{\hat{k}}\right)$ and $\widehat{\sigma}\left(M_{\hat{k}}\right)$ are constant with respect to $\hat{k}$. This implies that the CPA distinguisher $\mathrm{CPA}_{2,\mathsf{m}'}(\hat{k})$ satisfies the following equality:

$$\widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2 M'_{\hat{k}}\right) = a \cdot \mathrm{CPA}_{2,\mathsf{m}'}(\hat{k}) + b \ , \tag{5.13}$$

where a and b are two constant values satisfying

$$a = \widehat{\sigma}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\sigma}\left(M'_{\hat{k}}\right) \text{ and } b = \frac{\widehat{\mathbb{E}}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\mathbb{E}}\left(M'_{\hat{k}}\right)}{\widehat{\sigma}\left(\left(L - \widehat{\mathbb{E}}(L)\right)^2\right)\widehat{\sigma}\left(M'_{\hat{k}}\right)} \ .$$

From (5.12) and (5.13) we deduce that there exist two constant terms a and b and a model transformation f such that

$$\mathsf{VPA}_{(\alpha_i)_i,\mathsf{m}}(\hat{k}) = a \cdot \mathsf{CPA}_{2,\mathsf{m}'}(\hat{k}) + b \ , \tag{5.14}$$

with $\mathsf{m}' = \mathsf{f} \circ \mathsf{m}$. ◇

Proposition 3 implies the following corollary:
Corollary 3. Under Assumption 3, a VPA is SCA-reducible to a CPA_2.

5-5 Summing Distinguishers

In previous sections, we have established the SCA-reduction of DPA and PPA to CPA. Namely, we have shown that for every DPA or PPA with model m, there exists a new model $\mathsf{m}' = \mathsf{f} \circ \mathsf{m}$ such that a CPA with m' leads to a similar key-guess classification. This shows that, when performing such attacks, the real issue is the choice of the model and not the choice of the distinguisher. To deal with this issue when the best model is not known, an approach could consist in applying one of the distinguishers recalled in previous sections to a family of models $(\mathsf{m}_i)_i$ and to sum the results to define a new distinguisher. Actually, this distinguisher is still reducible to a CPA-distinguisher involving a model defined with respect to $(\mathsf{m}_i)_i$ and the "new" attack is thus nothing more than a CPA attack with a new model. This comes down as a consequence of the following lemma:
Lemma 1. Let $\mathsf{CPA}_{\mathsf{m}_1}(\hat{k})$ and $\mathsf{CPA}_{\mathsf{m}_2}(\hat{k})$ be two CPA distinguishing values defined for the same samples $(\ell_{k,i})_i$ and $(v_{\hat{k},i})_i$, and with two different model functions m_1 and m_2 respectively. Then, denoting by m_3 the function $\frac{\mathsf{m}_1}{\widehat{\sigma}\left(M_{\hat{k},1}\right)} + \frac{\mathsf{m}_2}{\widehat{\sigma}\left(M_{\hat{k},2}\right)}$, we have:

$$\mathsf{CPA}_{\mathsf{m}_1}(\hat{k}) + \mathsf{CPA}_{\mathsf{m}_2}(\hat{k}) = a\ \mathsf{CPA}_{\mathsf{m}_3}(\hat{k}) \ ,$$

where $M_{\hat{k},1}$, $M_{\hat{k},2}$ and $M_{\hat{k},3}$ denote the model variables associated with the model functions m_1, m_2 and m_3 respectively, and where $a = \widehat{\sigma}\left(M_{\hat{k},3}\right)$.

The idea consisting in summing several distinguishers to define a new one has been for instance applied by Bévan and Knudsen in [24] to enhance original Kocher's DPA. The authors propose to perform a single-bit DPA for each bit of the sensitive variable $Z_{\hat{k}}$ and then to sum the results. We call this attack a *Multiple-DPA* attack hereafter and we denote the involved distinguisher by M-DPA($\hat{k}$). It is defined as follows:

$$\text{M-DPA}(\hat{k}) = \sum_{j=0}^{t} \text{SB-DPA}(\hat{k})_j \tag{5.15}$$

where t is any integer lower than or equal to the dimension of v_k viewed as a binary-vector and where SB-DPA($\hat{k}$)$_j$ denotes the single-bit DPA with a model function m_j defined w.r.t. two real values $\omega_{0,j}$ and $\omega_{1,j}$ by $\mathsf{m}_j(v_{\hat{k}}) = (1 - v_{\hat{k}}[j]) \cdot \omega_{0,j} + v_{\hat{k}}[j] \cdot \omega_{1,j}$. As argued at the beginning of this section (and as a consequence of Propositions 1 and 2 and Lemma 1), this attack is SCA-reducible to a CPA. We state this in the following proposition and, for completeness, we exhibit in its proof the way how to define the CPA-distinguisher of this reduced CPA.

Proposition 4. Under Assumption 3, an M-DPA attack is SCA-reducible to a CPA.

Proof. Let us focus on Relation (5.15). Due to Proposition 1, for every j the single-bit DPA distinguisher SB-DPA($\hat{k}$)$_j$ is affinely reducible to the CPA-distinguisher CPA($\hat{k}$)$_j$ involving the model function $\mathsf{f}_j \circ \mathsf{m}_j$ where f_j is defined on $\text{Im}(\mathsf{m}_j) = \{\omega_{0,j}, \omega_{1,j}\}$ by $\mathsf{f}_j(\omega_{0,j}) = 1/\widehat{\mathrm{P}}\left(\mathsf{m}_j(Z_{\hat{k}}) = \omega_{0,j}\right)$ and $\mathsf{f}_j(\omega_{1,j}) = -1/\widehat{\mathrm{P}}\left(\mathsf{m}_j(Z_{\hat{k}}) = \omega_{1,j}\right)$. Let $M_{\hat{k},j}$ denote the random variable $\mathsf{f}_j \circ \mathsf{m}_j(Z_{\hat{k}})$. As a consequence of Proposition 1, we have:

$$\text{SB-DPA}(\hat{k})_j = \frac{\text{CPA}(\hat{k})_{\mathsf{m}_j} + b}{a} \ ,$$

with $a = \frac{1}{\widehat{\sigma}(L)\widehat{\sigma}\left(M_{\hat{k},j}\right)}$ and $b = \frac{\widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(M_{\hat{k},j}\right)}{\widehat{\sigma}(L)\widehat{\sigma}\left(M_{\hat{k},j}\right)}$. It can be checked that under Assumption 3, a and b are constant with respect to j and $\hat{k}$. We therefore deduce that (5.15) is equivalent to:

$$\text{M-DPA}(\hat{k}) = \frac{1}{a} \sum_{j=0}^{t} \text{CPA}(\hat{k})_{\mathsf{m}_j} + \frac{t \cdot b}{a} \ .$$

Lemma 1 then implies the following equality:

$$\text{M-DPA}(\hat{k}) = \frac{\widehat{\sigma}\left(M^{\star}_{\hat{k}}\right)}{a} \text{CPA}(\hat{k})_{\mathsf{m}^{\star}} + \frac{t \cdot b}{a} \ , \tag{5.16}$$

with $\mathsf{m}^\star$ being the function $\sum_{j=1}^{t} \frac{\mathsf{f}\circ\mathsf{m}_j}{\widehat{\sigma}\left(M_{\hat{k},j}\right)}$ and where $M_{\hat{k}}^\star$ denotes the model variable associated with $\mathsf{m}^\star$. ◇

5-6 A Brief Look at MIA Distinguisher

Previously we have analysed several distinguishers based on linear dependency (linear correlation). In Definition 5, a more generic distinguisher based on mutual information is described. From its mathematical definition (4.9) we see that only the term

$$\mathrm{H}[L \mid M_{\hat{k}}] = \sum_{y \in \mathrm{Im}(\mathsf{m})} \mathrm{P}\left[M_{\hat{k}} = y\right] \mathrm{H}[L \mid M_{\hat{k}} = y]$$

is key-dependent. Moreover the value $\mathrm{H}[L \mid M_{\hat{k}} = y]$ depends on the pdf of the conditional leakage $L \mid M_{\hat{k}} = y$ which is unknown. Thus an adversary can try to estimate this pdf to compute an estimation of the mutual information. Several pdf estimators exist in the literature [93] which leads to different mutual information estimations with different efficiency when used in MIA. Some of these pdf estimators have already been studied by Prouff *et al.* in [79], namely the *histogram*, the *parametric* and the *kernel method* estimators. Some other mutual information estimation methods exist that do not rely on estimation and are for instance based on polynomial density expansions. For a brief survey of mutual information estimation, we refer to [109]. When applied in an MIA, one of the most efficient attack in practice seems to be the histogram based one (introduced by Gierlichs *et al.* in [42]) as analysed in [22].

Histogram based MIA consists in grouping the samples into bins. The number of bins and their width are chosen w.r.t. the context of application and the nature of the samples. Several rules exist to empirically deduce these values (for instance in [104], [91] or [110]). Thus the bins are a partition of the range of the samples. In [42] it is suggested to use a number of – identical width – bins equal to the number of distinct model values (*i.e.* the number of expected components in the distribution). The histogram method can be straightforwardly extended into a multivariate context using multidimensional bins (see *e.g.* [79]).

As MIA defines another class of distinguishers, we mainly focused our study on linear correlation and not on MIA. In fact our main purpose

was to compare the existing attacks w.r.t. the CPA techniques which are the most widely used in practice.

Remark 14. Mutual information can detect any statistical dependency between $M_{\hat{k}}$ and L. Therefore, if $\mathsf{m} \circ F_{\hat{k}}$ is injective, $\mathrm{H}[L \mid M_{\hat{k}}]$ will lead to the same value for all $\hat{k}$ (for more details see [79]). Moreover, although the model $M_{\hat{k}}$ does not impact the asymptotic behavior of the MIA attack, it has a strong impact w.r.t. the efficiency of the MIA attack in terms of the number of messages needed (same as for CPA).

Remark 15. In some particular context, one can derive a link between the correlation factor and the mutual information. For instance in [62] it is shown that if X and Y are normally distributed then

$$\mathrm{I}(X\ ;\ Y) = -\frac{1}{2}\log_2\left(1-\rho(X\,,Y)^2\right)\ .$$

5-7 On the Choice of the Model

In previous sections we argued that most of existing linear power analysis attacks are reducible to CPAs that only differ in the model they involve. As a first important consequence, one of those attacks is more efficient than another one if and only if the corresponding SCA-reduced CPA involves a better model. This naturally raises the question of defining the model that optimizes the CPA efficiency. It has been proven in [80] that the model function $\mathsf{m} : v \mapsto \mathbb{E}\left[L \mid F_{\hat{k}}(X) = v\right]$ maximizes the amplitude of the correlation coefficient (4.8) when the good key is tested and hence optimizes the attack efficiency (as argued in [64]). In the context of univariate SCA with leakage satisfying (4.2), this function is the deterministic leakage function $\delta(\cdot)$. Note that any model $\mathsf{m}(\cdot) = a\ \delta(\cdot) + b$ where $a \neq 0$, b are constant will also maximize the amplitude of the correlation. As a particular observation, when all the bits of the targeted variable Z impact the leakage expectation, the result in [80] implies that the model must take into account all the bits of Z and that attacks exploiting only a limited number of bits (such as *e.g.*, the single-bit DPA) are sub-optimal. It is worth noticing that if the model is perfect (*i.e.* if $\mathsf{m}(\cdot) = \delta(\cdot)$), then under the *Gaussian Noise Assumption* (*i.e.* the noise B in (4.2) is drawn from a Gaussian distribution), the CPA is equivalent to a maximum likelihood attack [65], which is known to be optimal for key-recovery. Unfortunately, computing $\mathsf{m} : v \mapsto \mathbb{E}\left[L \mid F_{\hat{k}}(X) = v\right]$ is not possible with no *a priori* knowledge about L (*e.g.* without a profiling stage).

This implies that the adversary model is often not perfect and the resulting attacks are thus most of the time sub-optimal. In the next chapter, we investigate a family of side channel attacks that makes weaker assumptions on the device behavior than the CPA-like attacks.

CHAPTER 6

Linear Regression

In the previous chapter, we have shown the SCA-equivalence (or SCA-reduction) between the main used univariate SCAs and we have shown that the leakage modeling is a crucial point. In this chapter we introduce and analyse SCAs which need weaker knowledge on the leakage modeling. That is in classical univariate SCA we restrict the leakage modeling to one fixed function whereas in this new approach a set of functions sharing some algebraic properties is fixed, the attack will find and use the most relevant function in the set. To succeed, those attacks, termed *robust*, do not require a good affine estimation of the deterministic part $\delta(\cdot)$ of the device leakage. Actually, they only require some general assumptions on the algebraic properties of $\delta(\cdot)$ (namely the output value of the function is any linear combination of the bits of the input value). In particular, their efficiency does not rely on the adversary ability to find a model m which is a good affine approximation of $\delta(\cdot)$ as it was the case for CPA-like attacks.

In the following we first present a robust extension of DPA and then we introduce the linear regression which encompasses and formalizes the first latter attack. The leakage is assumed to be defined as in (4.2).

6-1 Robust Side Channel Attacks

In this section, we investigate robust side channel attacks that are able to succeed with only a very limited knowledge (compared to a CPA-based attack) on how the device leaks information. The starting point is to replace the requirement that the deterministic part of the leakage $\delta(\cdot)$ is greatly correlated to the attack model m by the weaker requirement that $\delta(\cdot)$ belongs to a set of functions sharing some algebraic properties. Thus the aim of robust attacks is to overcome the drawback of a – bad – leakage model selection and thus to perform a more generic attack.

Before presenting the attacks and in order to determine the kind of algebraic properties of $\delta(\cdot)$ we focus on, let us have a closer look at this function. As any real function defined over $\mathbb{F}_2^m$, it can be represented by a multivariate polynomial in $\mathbb{R}[z_1,\ldots,z_m]/({z_1}^2-z_1,\ldots,{z_m}^2-z_m)$ (*i.e.* the degree of every z_i in every monomial is at most 1). Consequently, there exists a unique set of real coefficients $(\alpha_u)_{u\in\mathbb{F}_2^m}$ such that for every $z\in\mathbb{F}_2^m$ we have:

$$\delta(z) = \sum_{u=(u_1,\ldots,u_m)\in\mathbb{F}_2^m} \alpha_u \cdot z^u \quad , \tag{6.1}$$

where each term z^u denotes the *monomial* (function) $z \longmapsto z_1^{u_1} z_2^{u_2} \cdots z_m^{u_m}$ with values in $\{0,1\}$ [28]. The *degree* of such a monomial is hence the Hamming weight of u. Equation 6.1 is called the *algebraic normal form* of the function δ. In view of (6.1), a side channel adversary could use his knowledge of the device technology to make an assumption on the degree d of $\delta(\cdot)$ viewed as a polynomial with coefficients in $\mathbb{R}$. This amounts to make the following assumption on the device.

Assumption 4 (Leakage Interpolation Degree)**.** The multivariate degree of the deterministic part $\delta(\cdot)$ of the leakage is upper bounded by d, for some d lower than or equal to m.

In practice and for most of devices such as smart cards, the coefficients α_u with $\mathrm{HW}(u)\leqslant 1$ are significantly greater than the others. This implies that the value of $\delta(x)$ is very close to the value of the linear part in (6.1), the other non-linear terms playing a minor role [57]. In this case,

it makes sense for the adversary to make Assumption 4 for $d = 1$. It is sometimes referred as the *Independent Bit Leakage* (IBL) Hypothesis in the literature [82] since it amounts to assume that the leakages related to the manipulation of two different bit-coordinates of Z are independent. This assumption fits well with the physical reality of numerous electronic devices. Indeed, the power consumption and electromagnetic emissions both result from logical transitions occurring on the circuit wires. Thus, assuming that every bit of a processed variable contributes independently to the overall instantaneous leakage is therefore realistic.

From an attacker point of view, assuming the IBL hypothesis is often a good strategy in practice since it enables to define an attack which, without being optimal, has an adequate efficiency. However, from the security designer perspective the IBL hypothesis may be considered as too restrictive. In this case indeed, the security analysis must include the largest class of adversaries as possible and proving resistance under the IBL hypothesis is therefore no longer sufficient. Moreover, for some new devices (*e.g.*, based on architectures using 65 nm manufacturing technology), it has been observed [37, 71, 82] that the coefficients of the quadratic terms in (6.1) are not negligible compared to those of the linear terms: the leakages related to the manipulation of two different bit-coordinates of Z are no longer independent. In this case, Assumption 4 for $d \geqslant 2$ shall yield a better representation of δ.

To sum up our discussion, even if making the Assumption 4 for $d = 1$ may be sufficient for an attacker to perform a succesfull attack, one (typically a device designer) must choose d as large as possible if the purpose is to test a device resistance in the worst case scenario.

In the next two sections we present two side channel attacks that are able to successfully recover the expected secret k with no other assumption on the deterministic part of the leakage than Assumption 4 for some fixed value of d. The two attacks are described in the particular case $d = 1$. This situation is indeed sufficient for most of practical attack contexts and it has the advantage to allow a simple description of the attacks. Eventually, in Section 6-1.1 we briefly explain how they can be simply extended to deal with degree $d > 1$ (*e.g.*, when neglecting the terms of degree greater than 1 leads to attack failure). This case ($d > 1$) is deeper analysed in Chap. 7.

6-1.1 Absolute Sum DPA

It may first be noticed that the multi-bit DPA (M-DPA($\hat{k}$)) recalled in Sect. 5-5 is not a robust extension of the binary single-bit DPA. Indeed, if we take a closer look at (5.1), we can check that the sign of each single-bit DPA distinguisher in the sum depends on the choice of the values $\omega_{0,j}$ and $\omega_{1,j}$. Hence, depending on the models m_j chosen for the attack, the sum of the values returned by the single-bit DPA distinguishers when the good key is tested may be very close to zero, which may result in a wrong-key discrimination. As already pointed out in [19], a straightforward solution to circumvent this issue consists in replacing the sum in (5.15) by a sum of absolute values – or a sum of squares – of single-bit DPA distinguishers. This leads to define the following AS-DPA distinguisher:

$$\mathsf{AS\text{-}DPA}(\hat{k}) = \sum_{i=0}^{t} \left| \mathsf{SB\text{-}DPA}(\hat{k})_i \right| \quad . \tag{6.2}$$

Contrary to what happens for M-DPA($\hat{k}$), the value of each element in the sum in AS-DPA($\hat{k}$) stays unchanged if we replace a family of bijective model functions $(\mathsf{m}_j)_j$ by another one. We can therefore choose any m which shows that our new AS-DPA attack is robust.

Illustration of the Differences Between AS-DPA, M-DPA and CPA
Let us focus on an adversary targeting the manipulation of a 2-bit intermediate value $Z = F_k(X)$ having a uniform distribution. For illustration purpose, we assume here that the attacked device leaks exactly the difference between the two bit-coordinates of Z. Namely we assume that L satisfies $L = \delta(Z)$, with $\delta(Z) = Z[0] - Z[1]$. As explained in [96], such a situation typically occurs when the leakage is measured by electromagnetic analysis. If the adversary performs a single-bit DPA to exploit L, a natural choice for $M_{\hat{k}}$ is either $F_{\hat{k}}(X)[0]$ or $F_{\hat{k}}(X)[1]$ (namely in (4.3) the model function m is the projection related to one of the bit-coordinates of $F_{\hat{k}}(X)$ and w_0 and w_1 equal 0 and 1 respectively). We denote by $\mathsf{SB\text{-}DPA}(\hat{k})_0$ (respectively $\mathsf{SB\text{-}DPA}(\hat{k})_1$) the distinguisher defined with respect to $M_{\hat{k}} = F_{\hat{k}}(X)[0]$ (respectively $M_{\hat{k}} = F_{\hat{k}}(X)[1]$). Under Assumption 3 which implies $\mathsf{var}[Z[0]] = \mathsf{var}[Z[1]]$ and the independency

between $Z[0]$ and $Z[1]$, we have

$$\text{SB-DPA}(\hat{k})_0 = \begin{cases} \mathbb{E}[Z[0]] - \mathbb{E}[1 - Z[0]] = 1 & \text{if } \hat{k} = k \\ 0 & \text{otherwise} \end{cases},$$

and

$$\text{SB-DPA}(\hat{k})_1 = \begin{cases} \mathbb{E}[Z[1]] - \mathbb{E}[1 - Z[1]] = 1 & \text{if } \hat{k} = k \\ 0 & \text{otherwise} \end{cases}.$$

Since we have $\text{SB-DPA}(\hat{k})_0 = -\text{SB-DPA}(\hat{k})_1$ for every $\hat{k}$, the distinguisher $\text{M-DPA}(\hat{k})$ in (5.15) always equals 0 whereas $\text{AS-DPA}(\hat{k}) = 2$ if $\hat{k} = k$ and 0 otherwise.

Let us now focus on the case where the adversary performs a CPA with the Hamming weight as a model function. When computing the correlation between the leakage L and the model random variable $M_{\hat{k}} = \text{HW}(F_{\hat{k}}(X)) = F_{\hat{k}}(X)[0] + F_{\hat{k}}(X)[1]$, we have:

$$\text{cov}\left(L, M_{\hat{k}}\right) = \text{cov}\left(Z[0] - Z[1], F_{\hat{k}}(X)[0] + F_{\hat{k}}(X)[1]\right)$$

which can be rewritten:

$$\begin{aligned} \text{cov}\left(L, M_{\hat{k}}\right) = \text{cov}\left(Z[0], F_{\hat{k}}(X)[0]\right) + \text{cov}\left(Z[0], F_{\hat{k}}(X)[1]\right) \\ - \text{cov}\left(Z[1], F_{\hat{k}}(X)[0]\right) - \text{cov}\left(Z[1], F_{\hat{k}}(X)[1]\right), \end{aligned}$$

from which we deduce $\text{CPA}(\hat{k}) = 0$ whatever the relation between $\hat{k}$ and k (since $\text{var}[Z[0]] = \text{var}[Z[1]]$).

To sum-up, this section gives an example of a leakage on a 2-bit variable for which the M-DPA and the CPA (with Hamming weight model function) fail, whereas the AS-DPA still succeeds.

Extension of the Attack to Non-linear Contexts If we relax Assumption 4 and assume that the leakage also depends on some monomials z^u with $d \geqslant \text{HW}(u) \geqslant 2$, then the corresponding SB-DPA cross-product $|\text{SB-DPA}(\hat{k})_1^{u_1} \times \text{SB-DPA}(\hat{k})_2^{u_2} \times \cdots \times \text{SB-DPA}(\hat{k})_m^{u_m}|$ can be added to the initial AS-DPA .

6-1.2 Linear Regression

In [88], Schindler *et al.* describe an efficient profiling method for SCA. Assuming that the attacker knows the subkey k, they explain how to

recover the leakage function δ (*i.e.*, the coefficients α_j under the IBL assumption) using linear regression. As mentioned by the authors, their approach could also enable the recovering of k (but neither details nor experiments are provided). We develop hereafter this idea that leads to a robust SCA.

The core idea is to discriminate the key-candidates by processing a *linear regression* on a key-dependent variable, denoted Y hereafter. To apply such a linear regression, the adversary must have chosen a basis $(\mathrm{g}_i)_{i=1,\ldots,d}$ of functions beforehand (see Sect. 7-2.2 on the basis choice). With this basis on hand, he then computes for each key-candidate a discriminating value and finally outputs the key-candidate which gives rise to the smallest value. For the sake of explanations, the linear regression at Step 3 of the attack below, is expressed in terms of distance from a function to a subspace of functions as introduced in Sect. 3-2. More precisely, the new attack is composed of the following steps:

1. **[Basis choice]** Choose a family of functions $(\mathrm{g}_i)_{1\leqslant i\leqslant d}$ defined from $\mathbb{F}_2^m$ into $\mathbb{R}$. The set spanned by the functions g_i is denoted by $\mathscr{H}$.
2. **[Measurement step]** For N plaintexts, collect measurements together with corresponding plaintexts sub-parts: $(\ell^i, x_i)_i \hookleftarrow (L, X)$. Then we define the vector y_N such that $y_N(i) = \ell^i$
3. **[Linear regression]** For every key hypothesis $\hat{k}$, compute:

$$\Delta_{\hat{k}}(N) = \mathrm{d}\left(y_N, \mathscr{G}_{\hat{k}}\right)^2 \quad , \tag{6.3}$$

   where $\mathscr{G}_{\hat{k}}$ denotes the space $< \mathrm{g}_1 \circ F_{\hat{k}}, \ldots, \mathrm{g}_d \circ F_{\hat{k}} >$.
4. **[Key candidate decision]** Select the key hypothesis for which $\Delta_{\hat{k}}(N)$ is minimal.

In the following, we shall associate the value $y_N(i)$ with the random variable $(Y \mid X = x_i)$, with Y being defined by:

$$Y = L \quad .$$

The computation of the minimum distance at Step 3 involves a linear regression to model the functional relationship between Y and X. The function is searched into a set which basis is constructed by composing the functions g_i with the key-hypothesis dependent function $F_{\hat{k}}$ defined in Sect. 4-3. This point is detailed hereafter while the way how to choose

the family of functions $(\mathrm{g}_i)_i$ is discussed in Sect. 7-2.2. The linear regression technique itself together with its link with the distance $\mathrm{d}(\cdot)$ between functions is detailed below.

Therefore, for a basis of functions $(\mathrm{g}_i)_{1\leq i\leq d}$, a set of noisy observations $(y_N(j))_{0<j\leqslant N}$ and a key candidate $\hat{k}$, the goal is to estimate:

$$\begin{aligned}\Delta_{\hat{k}}(N) &= \min_{\substack{(a_1,\ldots,a_d)\in\mathbb{R}^d \\ (a_1,\ldots,a_d)\neq(0,\ldots,0)}} \mathrm{d}\left(y_N, \left(\sum_i a_i \mathrm{g}_i\right)\circ F_{\hat{k}}\right)^2 \\ &= \min_{\substack{(a_1,\ldots,a_d)\in\mathbb{R}^d \\ (a_1,\ldots,a_d)\neq(0,\ldots,0)}} \sum_j \left(y_N(j) - \left[\left(\sum_i a_i \mathrm{g}_i\right)\circ F_{\hat{k}}\right](x_j)\right)^2 \quad .\end{aligned}$$

Note that the square root in the distance computation has no importance as we search for the minimum of a positive value (a sum of square). The linear regression technique involved in this paper starts by building the following *regression matrix*:

$$\mathbf{M} = \begin{pmatrix} \mathrm{g}_1(F_{\hat{k}}(x_1)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(x_1)) \\ \mathrm{g}_1(F_{\hat{k}}(x_2)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(x_2)) \\ \vdots & \ddots & \vdots \\ \mathrm{g}_1(F_{\hat{k}}(x_i)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(x_i)) \\ \vdots & \ddots & \vdots \\ \mathrm{g}_1(F_{\hat{k}}(x_N)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(x_N)) \end{pmatrix} ,$$

where the value x_i in $F_{\hat{k}}(x_i)$ is represented as an integer corresponding to the binary representation of $x_i \in \mathbb{F}_2^n$.

From the vector $\boldsymbol{y_N} = (y_N(0), y_N(1), \ldots, y_N(N))$ and $\mathbf{M}$, the following column vector $\boldsymbol{\alpha}_{\hat{k}}$ is computed:

$$\boldsymbol{\alpha} = {}^t(\alpha_1, \ldots, \alpha_d) = \left({}^t\mathbf{M}\cdot\mathbf{M}\right)^{-1}\cdot {}^t\mathbf{M}\cdot {}^t\boldsymbol{y_N} \quad . \tag{6.4}$$

Under the Gaussian assumption, it can be proved [39] that $[(\mathrm{g}_1,\ldots,\mathrm{g}_d)\cdot \boldsymbol{\alpha}]\circ F_{\hat{k}}$ is the function in $<\mathrm{g}_i>_{1\leq i\leq d}$ that is the closest one to y_N for the – squared – Euclidean distance.

Contrary to the attacks analysed in Chap. 5, which involve a fixed model function, regression attacks output a different model function $\mathrm{m}_{\hat{k}}$ for each key candidate $\hat{k}$. For the key discrimination step, an Euclidean distance is processed in place of a correlation coefficient with the leakage sample.

Remark 16. In the literature, *goodness of fit* is the common way to describe how well a model fits a set of observations. Different measures of goodness of fit can be used depending on the context. The *coefficient of determination* or the *Akaike information criterion* are examples of such a measure. In this paper, we privileged the following coefficient of determination:

$$R^2(\hat{k}) = \frac{\|L - \mathbf{M}\cdot\boldsymbol{\alpha}\|^2}{\mathsf{var}[L]} = \frac{\mathbb{E}\left[(\mathbf{L} - \mathbf{M}\cdot\boldsymbol{\alpha})^2\right]}{\mathsf{var}[L]} \quad . \tag{6.5}$$

It first permits to have a value in the range $[0,1]$. Moreover, it is closely related to the correlation coefficient. Note that in our specific case, all models result from a linear regression with the same basis functions set and with the same observations. This implies that in this particular case the main known estimators are equivalent to the Euclidian distance estimator.

6-2 Improvement

6-2.1 Averaging over Plaintexts

In the previous section, a linear regression techniques based attack is described. The main computation effort lies in the manipulation (*e.g.* multiplication and inversion) of the $N \times d$ regression matrix $\mathbf{M}$ and has thus at least a quadratic complexity (both matrix must be read at least once). This implies that for a large N the attack becomes very time consuming if not unfeasible. Nevertheless, one can remark that the random variable X is over $\mathbb{F}_2^n$ and thus take only 2^n values. The core idea of the improvement proposed here is to average the leakages according to the value of the corresponding plaintext. This leads to make a linear regression with an averaged leakage vector containing at most 2^n elements, namely the matrix size (and thus the computation complexity) is independent of N. With this practical trick, the linear regression reachs the same complexity than CPA (see Sect. 8-3 for practical results). More precisely, the improved attack is composed of the following steps:

1. **[Basis choice]** Choose a family of functions $(\mathrm{g}_i)_{1\leqslant i\leqslant d}$ defined from $\mathbb{F}_2^m$ into $\mathbb{R}$. The set spanned by the functions g_i is denoted by $\mathscr{H}$.
2. **[Measurement step]** For N plaintexts, collect measurements together with corresponding plaintexts sub-parts: $(\ell_1^i, x_i)_i \hookleftarrow (L, X)$.

3. **[Averaging step]** Partition the leakage measurements into sets $\mathscr{L}_x$ defined for every x such that $\mathscr{L}_x = \{\ell_1^i; x_i = x\}$. Then we define the function γ_N such that

$$\gamma_N(x) = \frac{1}{|\mathscr{L}_x|} \sum_{\ell_1 \in \mathscr{L}_x} (\ell_1) \ . \tag{6.6}$$

4. **[Linear regression]** For every key hypothesis $\hat{k}$, compute:*

$$\Delta_{\hat{k}}(N) = \mathrm{d}\left(\gamma_N, \mathscr{G}_{\hat{k}}\right)^2 \ , \tag{6.7}$$

   where $\mathscr{G}_{\hat{k}}$ denotes the space $< \mathrm{g}_1 \circ F_{\hat{k}}, \ldots, \mathrm{g}_d \circ F_{\hat{k}} >$.

5. **[Key candidate decision]** Select the key hypothesis for which $\Delta_{\hat{k}}(N)$ is minimal.

In this case, the regression matrix became

$$\mathbf{M} = \begin{pmatrix} \mathrm{g}_1(F_{\hat{k}}(0)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(0)) \\ \mathrm{g}_1(F_{\hat{k}}(1)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(1)) \\ \vdots & \ddots & \vdots \\ \mathrm{g}_1(F_{\hat{k}}(x)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(x)) \\ \vdots & \ddots & \vdots \\ \mathrm{g}_1(F_{\hat{k}}(2^n-1)) & \cdots & \mathrm{g}_d(F_{\hat{k}}(2^n-1)) \end{pmatrix} ,$$

and by analogy with the previous attack, from $\mathbf{M}$ and the vector $\boldsymbol{\gamma_N} = (\gamma_N(0), \ldots, \gamma_N(2^n-1))$, the following column vector $\boldsymbol{\alpha}$ is computed:

$$\boldsymbol{\alpha} = {}^t(\alpha_1, \ldots, \alpha_d) = \left({}^t\mathbf{M} \cdot \mathbf{M}\right)^{-1} \cdot {}^t\mathbf{M} \cdot {}^t\boldsymbol{\gamma_N} \ . \tag{6.8}$$

Remark 17. We assumed that the function γ_N is defined for every value in $\mathbb{F}_2^n$. Nevertheless in some cases (*e.g.* for a small N) it may happen that γ_N is defined only on a strict subset E of $\mathbb{F}_2^n$. In this case, the linear regression processing remains the same, except that lines corresponding to the values in $\mathbb{F}_2^n \backslash E$ are discarded from the matrix $\mathbf{M}$.

Proposition 5. Under Assumption 1 linear regression with averaged leakage vector as observation will output the same model as linear regression with non-averaged leakage vector.

*It must be observed that the distance in (6.7) is computed over 2^n values, whereas the distance in (6.3) is computed over N values.

Proof. When applying the ordinary least-square algorithm described in Sect. 6-1.2, we minimize the least-square error function defined as $f(\boldsymbol{\alpha}) = \left(\mathrm{d}(y_N, \sum_{j=1}^{d} \alpha_j \mathrm{g}_j \circ F_{\hat{k}})\right)^2 = \sum_i [y_N(i) - \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x_i))]^2$ which can be rewritten in matrix notation as $f(\boldsymbol{\alpha}) = {}^t(\boldsymbol{y_N} - \mathbf{M}\cdot\boldsymbol{\alpha})\cdot(\boldsymbol{y_N} - \mathbf{M}\cdot\boldsymbol{\alpha})$. By definition, this "real value" function admits a unique global minimum (because it is quadratic in $\boldsymbol{\alpha}$ with positive-definite Hessian) which can be explicitly computed as the unique solution which zeroizes the partial derivatives. Thus, finding this minimum is equivalent to solve the system of d partial derivatives

$$\mathrm{Eq}_k: \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2\sum_i \left(\left[y_N(i) - \sum_j^d \alpha_j \mathrm{g}_j(F_{\hat{k}}(x_i)) \right] \cdot \mathrm{g}_k(F_{\hat{k}}(x_i)) \right) = 0 \ , \quad (6.9)$$

for $k = 1, \ldots, d$.

This is equivalent to the following matrix system:

$$\frac{\partial f(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = -2\cdot {}^t\mathbf{M}\cdot\boldsymbol{y_N} + 2\cdot {}^t\mathbf{M}\cdot\mathbf{M}\cdot\boldsymbol{\alpha} = 0 \ . \quad (6.10)$$

Assuming ${}^t\mathbf{M}\cdot\mathbf{M}$ is invertible, we deduce straightforwardly (6.4) from (6.10). By analogy, (6.8) can be deduced from the following system:

$$\mathrm{Eq}'_k: \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2\sum_{x=0}^{2^n} \mathrm{g}_k(F_{\hat{k}}(x)) \cdot \left(\gamma_N(x) - \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x)) \right) = 0 \ , \quad (6.11)$$

Taking a closer look to (6.9), it can be rewritten as:

$$\mathrm{Eq}_k: \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2\sum_{x=0}^{2^n} \sum_{i;x_i=x} \left(\left[y_N(i) - \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x)) \right] \cdot \mathrm{g}_k(F_{\hat{k}}(x)) \right) = 0 \ ,$$

which gives

$$\mathrm{Eq}_k: \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2\sum_{x=0}^{2^n} \mathrm{g}_k(F_{\hat{k}}(x)) \cdot \left(\sum_{i;x_i=x} y_N(i) - \sum_{i;x_i=x} \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x)) \right) = 0 \ .$$

After denoting by c_x the cardinal of the set $\{i; x_i = x\}$, we obtain

$$\mathrm{Eq}_k: \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2\sum_{x=0}^{2^n} c_x \cdot \mathrm{g}_k(F_{\hat{k}}(x)) \cdot \left(\gamma_N(x) - \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x)) \right) = 0 \ .$$

Under Assumption 1, c_x is constant w.r.t. x thus the system becomes:

$$\mathrm{Eq}_k : \quad \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} = -2 \sum_{x=0}^{2^n} \mathrm{g}_k(F_{\hat{k}}(x)) \cdot \left(\gamma_N(x) - \sum_{j=1}^{d} \alpha_j \mathrm{g}_j(F_{\hat{k}}(x)) \right) = 0 \ ,$$

The system of equations $(\mathrm{Eq}_k)_{k=1,\ldots,d}$ exactly corresponds to the system $(\mathrm{Eq'}_k)_{k=1,\ldots,d}$ resolved in (6.11). Thus both systems are equivalent and will lead to the same solutions, which concludes this proof. ◇

Remark 18. We assumed that c_x is constant which implies that each plaintext value appears the same number of times in the sample. This can be straightforwardly done by omitting some values when the sample size is sufficiently large. Notice that if this condition is not fulfilled, the variance of the averaged traces are different.

This practical trick can be applied as soon as a mean computation (more generally a conditional moment computation) according to the plaintexts is involved (for instance in a CPA attack). Although the gain is not visible on one single execution, this trick can be viewed as a preprocessing step of the leakage which can be interesting when several attacks are performed on the same leakage set (for instance to test different models on CPA or to compare some attacks).

6-2.2 Adaptive basis

As pointed out in Sect. 7-2.2, the linear regression attack needs a choice of a suitable basis of functions which largely impacts the attack efficiency. The size of such a basis becomes a real drawback when few samples are available. To overcome this problem an idea for further works could be to study the *least angle regression* introduce by Efron *et al.* [38]. The core principle consists in computing a linear regression by adding basis elements to one by one (instead of taking into account the whole basis directly) from the most to the least correlated coefficient with the observation. This allows us to bypass the drawback of the basis size and moreover permits to converge more quickly to the solution as the basis elements are inserted in order of importance. Eventually, least angle regression algorithm is a least-square refinement which can perform better in some contexts. Nevertheless the size of the basis is still a limiting factor and the idea to use a very large basis and let the least angle regression choose the good elements seems not to be a valid approach with an “out-of-the-box” algorithm. Additional optimizations (such as *lasso*

optimization [38]) can perhaps bypass this drawback. The principle of lasso is straightforward and consists in limiting the number of basis elements to add during the least angle regression computation. In other words, instead of fixing a basis, we let to the algorithm the choice of the most pertinent elements from a larger set. For instance the attacker has just to fix the size of the basis. Even if we have performed some encouraging experiments, more investigations must be conducted.

More formally, the least angle algorithm is based on the fact that if the basis is orthogonal, elements have no effect on each others coefficient estimations and thus we can estimate separately each coefficient. That is, after the orthogonalisation of the basis, we compute one coefficient, subtract it from the residue and do it again until all coefficients are computed or the residue becomes null. Theoretical foundations and detailed algorithms can be found in the original paper [38].

CHAPTER 7

High-order

7-1 Introduction to High-Order

7-1.1 The Sharing Concept

In the previous chapters we focused on univariate SCAs, that is we focused on an unprotected implementation. A common countermeasure against such an attack is the use of *masking*. When the cryptosystem is protected thanks to a $(d)^{\text{th}}$-order masking scheme, the sensitive variable Z is randomly split into $d+1$ (at least) shares $V_0, V_1, \ldots, V_d$ that are manipulated at different times [30]. The manipulation of the shares results in $d+1$ observable physical leakages denoted by $L_0, L_1, \ldots, L_d$. The parameter d is usually called the *masking order*. The analyses conducted in the following are done under

the assumption that the leakages satisfy:

$$L_i = \delta(V_i) + B_i,\ 0 \leqslant i \leqslant d \quad , \tag{7.1}$$

where $\delta(\cdot)$ is a deterministic unknown function and the random variables B_i are independent but identically distributed unidimensional Gaussian variables. Notice that in (7.1) we made the classical assumption that $\delta(\cdot)$ only depends on the underlying hardware, independently of the time. Usually the random variables $V_1, \ldots, V_d$ are independent and uniformly distributed and V_0 is defined such as $V_0 = Z \star V_1 \star \cdots \star V_d$ where $\star$ is an operation law such that $(\mathbb{F}_2^n, \star)$ is a group*. For the rest of the thesis, Assumption 2 is extended to the following assumption:
Assumption 5 (Independent Noise)**.** The noises B_i are independent of the shared variables V_i.

7-1.2 High-Order Side Channel Attacks

To defeat a d^{th}-order masking, an attacker has to exploit at least $d+1$ leakages corresponding to $d+1$ shares manipulation. In this multivariate context the framework presented in Sect. 4-3 becomes:

1. Perform N measurements $(\ell_0^i, \ldots, \ell_d^i)_i \hookleftarrow (L_0, \ldots, L_d)$ on the cryptographic device using a sample $(x_i)_i \hookleftarrow X$ of plaintexts.
2. Choose a function m to model the deterministic part of the leakage.
3. Compute the model values $m_{\hat{k},i} = \mathsf{m}(F_{\hat{k}}(x_i))$ from the plaintexts x_i and the model function m, for every key hypothesis $\hat{k}$.
4. Choose a statistical distinguisher Δ.
5. For every key hypothesis $\hat{k}$, compute the *distinguishing value* $\Delta_{\hat{k}}$ defined by:
$$\Delta_{\hat{k}} = \Delta\left((\ell_0^i, \ldots, \ell_d^i)_i, (m_{\hat{k},i})_i\right) \quad .$$
This results in a *score vector* $(\Delta_{\hat{k}})_{\hat{k}}$.
6. Output as the o most likely key candidates the o key hypotheses that maximize – or minimize – $\Delta_{\hat{k}}$.

*for instance $\star$ may be the bitwise addition $\oplus$ or the addition + modulo 2^n where Z and V_i are viewed as elements of $\mathbb{Z}/2^n\mathbb{Z}$.

In a multivariate context, the choice of the distinguisher and the choice of the model are still crucial points. Moreover the choice of the distinguisher can hide the choice of a combination function to pass from a multivariate to an univariate context which permits to apply univariate statistical tools.

Several combining functions have been proposed in the literature. Two of them are commonly used: the *product combining* [30] which consists in multiplying the different signals and the *absolute difference combining* [69] which computes the absolute value of the difference between two signals. As noted in [33, Sect. 1], the latter can be extended to higher-orders by induction. Other combining functions have been proposed in [48, 73]. In a recent paper [80], the different combining functions are compared for second-order DPA in the Hamming weight model. An improvement of the product combining called *normalized product combining* or *centralized product combining* is proposed and it is shown to be more efficient than the other combining functions*.

7-2 A Particular Case: Second-Order

In the following, we describe a bivariate attack targeting two different time instants with leakage L_0 and L_1. In this context, (7.1) becomes

$$L_0 = \delta(Z \star V) + B_0 \quad \text{and} \quad L_1 = \delta(V) + B_1 \ . \tag{7.2}$$

In what follows, a new second-order attack is introduced, extending to a masked context the strategy described in the previous chapter. The core idea is to discriminate the key-candidates by processing a *linear regression* on a key-dependent variable which combines the two leakages defined in (7.2). More precisely, the new attack is composed of the following six steps – derived from the ones in Sect. 6-1.2:

1. **[Basis choice]** Choose a family of functions $(\mathrm{g}_i)_{1 \leqslant i \leqslant d}$ defined from $\mathbb{F}_2^m$ into $\mathbb{R}$. The set spanned by the g_i is denoted by $\mathscr{H}$.
2. **[Measurement step]** For N plaintexts, collect measurements together with the corresponding plaintexts sub-parts: $(\ell_0^i, \ell_1^i, x_i)_i \hookleftarrow (L_0, L_1, X)$.

*This assertion is true while considering a noisy model. In a fully idealized model, other combining may provide better results (see [80]).

3. **[Partitioning step]** Partition the pair of leakage measurements into sets $\mathscr{L}_x$ defined for every $x \in \mathbb{F}_2^n$ such that $\mathscr{L}_x = \{(\ell_0^i, \ell_1^i); x_i = x\}$.
4. **[Combining step]** For every $x \in \mathbb{F}_2^n$, compute:

$$y_N(x) = \frac{1}{|\mathscr{L}_x|} \sum_{(\ell_0,\ell_1)\in\mathscr{L}_x} \big(\ell_0 - \mu_0(x)\big)\big(\ell_1 - \mu_1(x)\big) \quad , \tag{7.3}$$

   where y_N is a function of x parameterized by the number of collected measurements, and where $\mu_0(x)$ and $\mu_1(x)$ respectively denote $\mathbb{E}[L_0 \mid X = x]$ and $\mathbb{E}[L_1 \mid X = x]$. For analysis purpose, $y_N(x)$ is viewed* as an approximation of $y(x) = \mathrm{cov}(L_0 \mid X = x, L_1 \mid X = x)$.
5. **[Linear regression]** For every key hypothesis $\hat{k}$, compute:

$$\Delta_{\hat{k}}(N) = \mathrm{d}\left(y_N, \mathscr{G}_{\hat{k}}\right)^2 \quad , \tag{7.4}$$

   where $\mathscr{G}_{\hat{k}}$ denotes the space $< \mathrm{g}_1 \circ F_{\hat{k}}, \ldots, \mathrm{g}_d \circ F_{\hat{k}} >$.
6. **[Key candidate decision]** Select the key hypothesis for which $\Delta_{\hat{k}}(N)$ is minimal.

A discussion of the new attack rationale will be conducted in the next section. We can however sum-up its main steps in the following way. First, and due to the univariate aspect of the linear regression, the leakages L_0 and L_1 are combined in Step 4 to form an univariate random variable Y which can be viewed as estimation of the covariance between L_0 and L_1 knowing $X = x$. The latter covariance, viewed as a function of the random variable X is denoted by Y in the following. The computation of the minimum distance during the fifth step involves linear regression to find a good model for the functional relationship between Y and X. The model is searched into a set of functions which basis is constructed by composing the g_i with the key-hypothesis dependent function $F_{\hat{k}}$ defined in Sect. 4-3. This point is detailed in the next section while the way how to choose the family of functions $(\mathrm{g}_i)_i$ is discussed in Sect. 7-2.2. The linear regression technique and its link with the distance $\mathrm{d}(\cdot)$ can be found in Sect. 6-1.2.

Remark 19. The distinguisher in (7.4) is equivalent to the maximization of the so-called *coefficient of determination* between the $y_N(x_i)$ and the $\mathrm{g}_i(x_i)$ for g_i ranging over $\mathscr{G}_{\hat{k}}$ (see [39] for more details about this

*The pertinence of this definition of Y (and hence of the construction of $y_N(x)$ in the attack) is discussed in Sect. 7-2.3.

coefficient). Hence, another way of interpreting our linear regression based SCA is as follows. The attack attempts to express the trace measurements $(\ell_0^i, \ell_1^i)_i$ in terms of a polynomial function of the bits of the $\hat{k}$-dependent predicted values $\left(F_{\hat{k}}(x_i)\right)_i$ with additive noise. The polynomial is searched into the set generated by the basis functions $(\mathrm{g}_i)_{i=1\cdots d}$ themselves viewed as polynomial functions with coefficients in $\mathbb{R}$. The idea is that only the correct predictions will give a good least-squares approximation. The natural measure of goodness-of-fit is the coefficient of determination, which can be interpreted as the proportion of the variance in the traces which is accounted by the model. So, for a well-chosen regression function, only the correctly predicted bits will give a good explanation of the variance.

7-2.1 Rationale Behind the New Attack

In this section we analyse the theoretical foundation of such an attack. Since L_0 and L_1 satisfy (7.2), and random variables B_0, B_1 and V are independent, the definition of Y can be rewritten

$$Y = \varphi[F_k(X)] \ , \tag{7.5}$$

where φ denotes the function

$$z \mapsto \mathrm{cov}(\delta(z \star V), \delta(V)) \ .$$

By construction, the function y_N defined in Step 4 tends toward y as the number of measurements increases. Therefore from (7.5) and some terms rearrangements, one deduces the following limit of $\Delta_{\hat{k}}(N)$, where we recall that y is considered as a function of x:

$$\begin{aligned} \lim_{N\to\infty} \Delta_{\hat{k}}(N) &= \lim_{N\to\infty} \mathrm{d}\left(y_N, \mathscr{G}_{\hat{k}}\right)^2 = \mathrm{d}(Y, \mathscr{G}_{\hat{k}})^2 \\ &= \min_{\substack{h\in\mathscr{H} \\ h\neq 0}} \mathrm{d}\left(\varphi\circ F_k \circ F_{\hat{k}}^{-1} \circ F_{\hat{k}}, h\circ F_{\hat{k}}\right)^2 \ . \end{aligned} \tag{7.6}$$

Assuming that $F_{\hat{k}}$ is balanced, (7.6) simplifies to

$$\lim_{N\to\infty} \Delta_{\hat{k}}(N) = 2^{n-m} \cdot \mathrm{d}\left(\varphi\circ F_k \circ F_{\hat{k}}^{-1}, \mathscr{H}\right)^2 \ . \tag{7.7}$$

Now, depending on whether $\hat{k}$ equals k or not, we have the two following situations:

Good hypothesis ($\hat{k} = k$): Equation (7.7) becomes $\lim_{N\to\infty} \Delta_{\hat{k}}(N) = 2^{n-m} \cdot \mathrm{d}\left(\varphi, \mathscr{H}\right)^2$.

Wrong hypothesis ($\hat{k} \neq k$): Equation (7.7) cannot be simplified.

From those two situations, we deduce that the new attack outputs the correct key if the distance between $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ and $\mathscr{H}$ is minimized when $\hat{k} = k$ (*e.g.* when $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ equals φ). This implies that the choice of $\mathscr{H}$ must be relevant. In other words, it highlights the importance of the choice of the basis $(\mathrm{g}_i)_i$. This choice is discussed in the next section.
Remark 20. Notice that in this case, $F_{\hat{k}}^{-1}$ can refer to a set of functions and thus the distance is computed over two sets (see (3.7)).

7-2.2 Basis Choice

As pointed out in previous sections, the basis choice is essential since it directly impacts the attack efficiency. Ideally, the basis should guarantee the adversary that $\mathrm{d}\left(\varphi \circ F_k \circ F_{\hat{k}}^{-1}, \mathscr{H}\right)$ is minimal when $\hat{k} = k$. In this section, we propose a strategy for the adversary to choose it.

By definition, the function φ to be approximated belongs to the space $\mathscr{F}$ of all the functions from $\mathbb{F}_2^m$ into $\mathbb{R}$. We recall that any function in $\mathscr{F}$ can be represented in algebraic normal form (see Eq . (6.1)). It can moreover be checked that the family of functions $(z^u)_{u\in\mathbb{F}_2^m}$ spans $\mathscr{F}$ [28]. In the following, we denote by $\mathscr{F}_d$ the subset of $\mathscr{F}$ that contains all the functions of degree lower than or equal to d. This set is spanned by the basis $(z^u)_{u\in\mathbb{F}_2^m, \mathrm{HW}(u)\leq d}$.

Let us now come back to the attack described in Sect. 6-1.2 and extended in Sect. 7-2. If the set $\mathscr{H}$ spanned by the functions $(\mathrm{g}_i)_i$ equals $\mathscr{F}$ (*i.e.* $(\mathrm{g}_i)_i$ is also a basis of $\mathscr{F}$), then for any $F_{\hat{k}}$ and F_k it is obvious that $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ is in $\mathscr{H}$. As a consequence, the distance $\mathrm{d}\left(\varphi \circ F_k \circ F_{\hat{k}}^{-1}, \mathscr{H}\right)$ is always null, the key hypothesis $\hat{k}$ being equal to k or not. This implies that choosing the basis $(\mathrm{g}_i)_i$ as large as possible is not a sound approach for our attack (see Sect. 7-3 for more details). Let us now denote by $\mathscr{J}$ the set of functions $\{F_k \circ F_{\hat{k}}^{-1}; k \neq \hat{k}\}$.

A much better strategy an adversary can follow is to look for a subspace $\mathscr{H}$ such that $\varphi \in \mathscr{H}$ (*i.e.* the distance between φ and $\mathscr{H}$ is null) while the distance between the two sets $\mathscr{H}$ and $\mathscr{H} \circ \mathscr{J}$ is as high as possible

(Fig. 7-2-1 illustrates it). For such a purpose, we propose here to make an assumption on the degree d of φ and to set $\mathscr{H} = \mathscr{F}_d$. This amounts to choose the basis such that $(\mathrm{g}_i)_i = (z^u)_{u \in \mathbb{F}_2^m, \mathrm{HW}(u) \leq d}$. Since the composition of functions $F_k \circ F_{\hat{k}}^{-1}$ is very likely to have a high degree (close to m) due to the cryptographic properties* of F, then none of the functions $\varphi \circ F_k \circ F_{\hat{k} \neq k}^{-1}$ is in $\mathscr{F}_d$ for d small enough, whereas $\varphi \circ F_k \circ F_{\hat{k}=k}^{-1} = \varphi$ does (by hypothesis).

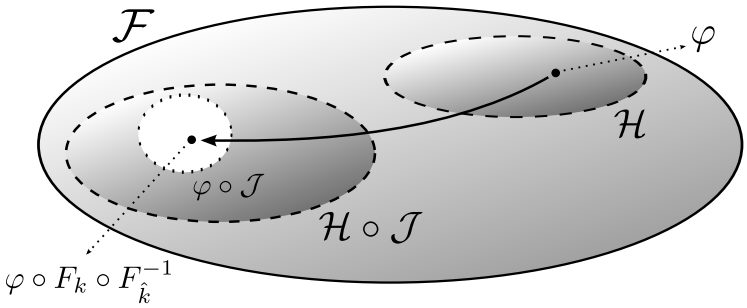


Fig. 7-2-1 – Relationship between the different spaces.

Remark 21. In our strategy, we assumed that the attacker targets the result of a non-linear transformation (*e.g.* an S-box) and thus that the function F is likely to have a high degree. Nevertheless, one can choose to target the result of a linear transformation (typically the manipulation of the sensitive variable just *before* the non-linear transformation). In this case, the choice of the basis is less obvious and will be very dependent on the algebraic properties of φ. Therefore the choice of a basis must be adapted to the knowledge or assumptions on both φ and F (*i.e.* it depends on both the nature of the leakage and the nature of the targeted sensitive variable).

To conclude this section, we give hereafter an example of our strategy in a realistic attack context.

Example. Let us assume that F_k is an AES S-box. Then the set $\mathscr{J}$ contains all the functions that are the composition of AES SubByte with

*These properties relate to the fact that, by construction, functions F_k and $F_{\hat{k}}$ must be as independent as possible when parameterized by different keys. Moreover, the family of functions F_k must have a high algebraic degree (close to m) to defeat linear and differential cryptanalyses. As a consequence, the composition of functions F_k and $F_{\hat{k}}$, with $k \neq \hat{k}$, must act as a random composition of functions with high algebraic degrees. With very high probability, such a composition results in a function with high degree. If required, this hypothesis may be tested for a target function F by computing the minimim degree of the functions in $\mathscr{J}$.

an AES InvSubByte, the two S-boxes being parameterized by different keys. By property of the AES S-box, every function in $\mathscr{J}$ will be far, in terms of distance, from the set of affine functions (this relates to the high *non-linearity* of the S-box). Hence, a good strategy is to assume that φ belongs to the set of linear functions $\mathscr{F}_1$ (*i.e.* $(g_i)_i = (z^u)_{u \in \mathbb{F}_2^m, \mathrm{HW}(u) \leq 1}$). Indeed, in this case the linear regression will compute a good approximation of φ in $\mathscr{F}_1$, while by definition of $\mathscr{J}$, it will not be able to compute a good approximation of $\varphi \circ j$ for any $j \in \mathscr{J}$.

7-2.3 Relationship with Other Attacks

7-2.3.1 Relationship with Second-Order CPA

A second-order CPA using the *centered product combining* function has been introduced in [80] and compared favorably to other attacks based on the correlation coefficient. In fact, this CPA may be viewed as a particular case of our attack where the space spanned by the basis (g_i) is reduced to a single function $\hat{\varphi}$ that is assumed to approximate the function φ defined in (7.5) (*e.g.* the Hamming weight function is chosen for $\hat{\varphi}$). Indeed, in such a particular case, the distance computation (7.4) can be rewritten:

$$\Delta_{\hat{k}}(N) = \mathrm{d}(y_N, \mathscr{H} \circ F_{\hat{k}})^2 = \mathrm{d}(y_N, \hat{\varphi} \circ F_{\hat{k}})^2 \ , \tag{7.8}$$

since $\mathscr{H} = \{\hat{\varphi}\}$.

Now asymptotically (7.8) becomes:

$$\lim_{N \to \infty} \Delta_{\hat{k}}(N) = \mathrm{d}(y, \hat{\varphi} \circ F_{\hat{k}})^2 = \mathrm{d}(y, \hat{y})^2 \ , \tag{7.9}$$

where we have denoted $\hat{\varphi} \circ F_{\hat{k}}$ by $\hat{y}$ and where we recall that y denotes $\varphi \circ F_k$. Equation (7.9) can be rewritten:

$$\begin{aligned} \lim_{N \to \infty} \Delta_{\hat{k}}(N) &= \sum_{x \in \mathbb{F}_2^n} \left([\varphi \circ F_k](x) - [\hat{\varphi} \circ F_{\hat{k}}](x) \right)^2 \\ &= 2^n \cdot \mathbb{E}\left[(Y - \hat{Y})^2 \right] \ . \end{aligned} \tag{7.10}$$

After developing (7.10), we get:

$$\begin{aligned} &\lim_{N \to \infty} \Delta_{\hat{k}}(N) \\ = \ & 2^n \cdot \left(\mathbb{E}\left[Y^2 \right] + \mathbb{E}\left[\hat{Y}^2 \right] - 2 \cdot \mathbb{E}\left[Y \cdot \hat{Y} \right] \right) \ . \end{aligned} \tag{7.11}$$

As a consequence, if $\rho(Y,\hat{Y})$ denotes the correlation coefficient between Y and $\hat{Y}$ viewed as random variables functionally dependent on X, we recall that $\rho(Y,\hat{Y})$ satisfies:

$$\begin{aligned}\rho(Y,\hat{Y}) &= \frac{\mathrm{cov}\left(Y,\hat{Y}\right)}{\sigma_Y \cdot \sigma_{\hat{Y}}} \\ &= \frac{1}{\sigma_Y \cdot \sigma_{\hat{Y}}} \cdot \left(\mathbb{E}\left[Y \cdot \hat{Y}\right] - \mathbb{E}[Y] \cdot \mathbb{E}\left[\hat{Y}\right]\right) \ ,\end{aligned} \tag{7.12}$$

From (7.11) and (7.12), we deduce:

$$\lim_{N\to\infty} \Delta_{\hat{k}}(N) = a \cdot \rho + b \ , \tag{7.13}$$

where

$$\begin{aligned} a &= -2^{n+1} \cdot \sigma_Y \cdot \sigma_{\hat{Y}} \text{ and} \\ b &= 2^n \cdot \left(\mathbb{E}\left[Y^2\right] + \mathbb{E}\left[\hat{Y}^2\right] - 2 \cdot (\mathbb{E}[Y] \cdot \mathbb{E}\left[\hat{Y}\right])\right) \end{aligned}$$

are independent of the key hypothesis provided σ_Y, $\sigma_{\hat{Y}}$, $\mathbb{E}\left[Y^2\right]$, $\mathbb{E}\left[\hat{Y}^2\right]$, $\mathbb{E}[Y]$ and $\mathbb{E}\left[\hat{Y}\right]$ are also independent of the key hypothesis*.

Equation (7.13) above shows that our new attack with space $\mathscr{H}$ reduced to a single function $\hat{\varphi}$ is asymptotically equivalent to a second-order CPA involving the centered product as combining function and $\hat{\varphi}$ as prediction function.

7-2.3.2 Relationship with Maximum Likelihood Approach

In a second-order attack based on a maximum likelihood approach [31, 43, 73, 88], the adversary knows for every z a good estimation of the pdf f_z of the random variable $((L_1,L_2)|Z=z)$. With such a knowledge and a sample $(\ell_1^i,\ell_2^i,x_i)_i \hookleftarrow (L_1,L_2,X)$ measured on the targeted device, the adversary then computes for each key candidate $\hat{k}$, a set of predictions $(\hat{z}_i)_i = (F_{\hat{k}}(x_i))_i$ and selects the key that maximizes the product $\prod_i f_{\hat{z}_i}(\ell_1^i,\ell_2^i)$. This class of attack, which has first been introduced in [31] under the name of *template attacks*, can be very efficient if the profiling phase is done precisely enough. However, as previously observed in many papers, the assumption that the adversary has a good approximation of f_z in hand strongly limits the attack practicability and raises the

*This is clearly the case with typical first-order masking schemes involving an addition, like Boolean and arithmetic masking schemes.

need for alternative approaches. To some extent, the attack presented in Sect. 7-2 can be viewed as such an alternative. More precisely, it may be viewed as an application of the template attacks principle in a context where the adversary has no *a priori* knowledge of the f_z but tries to reconstruct them on-the-fly. To further discuss on this statement, let us develop the pdfs f_z under Gaussian assumption.

When the leakage is defined as in (7.2), the f_z are *mixture* of *elliptic normal distributions* [76]. Namely they are defined such that:

$$f_z = \frac{1}{2^n} \sum_{v \in \mathbb{F}_2^n} \Phi_{\boldsymbol{m}_{z,v},\Sigma} \ , \tag{7.14}$$

where $\Phi_{\boldsymbol{m}_{z,v},\Sigma}$ denotes the pdf of the multivariate Gaussian distribution with mean $\boldsymbol{m}_{z,v}$ and covariance matrix Σ, where $\boldsymbol{m}_{z,v}$ and Σ satisfy:

$$\boldsymbol{m}_{z,v} = (\delta(z \star v), \delta(v)) \text{ and } \Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} .$$

Our attack implicitly tries to approximate the distribution f_z by a bivariate Gaussian pdf and this is actually the main difference between it and template attacks. The use of such an approximation is known in the literature as the technique of *merging* the mixture components [86] with a limited and fixed number of components (here 2). It leads us to make the following approximation:

$$f_z \sim \Phi_{\boldsymbol{m},\Sigma_z} \ , \tag{7.15}$$

where $\boldsymbol{m} = (\mathbb{E}[\delta(z \star V)], \mathbb{E}[\delta(V)])$ and*

$$\Sigma_z = \begin{pmatrix} \sigma^2 & y(x) \\ y(x) & \sigma^2 \end{pmatrix} ,$$

where x corresponds to one pre-image of z through F_k and where y satisfies (7.3).

In view of the definitions of $\boldsymbol{m}$ and Σ_z it is clear that the only key-dependent parameter of the pdf approximation (7.15) is $y(x)$. Thus, testing whether an observation (ℓ_1, ℓ_2) comes from a distribution $\Phi_{\boldsymbol{m},\Sigma_z}$ reduces to test whether (ℓ_1, ℓ_2) comes from a bivariate distribution with

*Note that $\boldsymbol{m}$ exactly corresponds to the development of the mean vector $(\mathbb{E}[L_1|Z = z], \mathbb{E}[L_2])$ when using the linearity of the expectation and the fact that the noise is assumed to have zero mean.

covariance $y(x)$. As explained in Sect. 7-2.1, our new attack computes an estimation of this variable, the estimation being parametrized by a key hypothesis. Then, to validate the hypothesis (or equivalently the quality of the approximation of $y(x)$ for every x), a mean-of-square test is computed. It is well known that this test is equivalent to a maximum likelihood computation under the Gaussian Assumption. Some simulations can be found in Sect. 8-2.5.

Remark 22. Another more precise way of approximating the distributions may be to look for approximations by mixtures of Gaussian distributions. This approach has already been suggested in [58] but its soundness is still under discussion, it involves a class of algorithms, called *expectation-maximization* (EM) algorithms.

7-3 Models and metrics

As pointed out in Sect. 7-2.2, using a basis of the full space $\mathscr{F}$ as basis in the linear regression attack is not a good strategy to follow. In fact, the linear regression will always find the approximated function $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for every $\hat{k}$ and thus the Euclidean distance does not permit to discriminate k. In other words, we are interested in discriminating φ from $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for $\hat{k} \neq k$. Indeed we can expect that φ and $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for $\hat{k} \neq k$ have a different behavior which can be directly noticed in their algebraic normal form. We recall that linear regression with a basis of the full space will output the algebraic normal form of the approximated function (see Sect. 7-2.2).

We suggest that the algebraic normal form can permit to discriminate φ from $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for $\hat{k} \neq k$. Namely the distribution of the coefficients in algebraic normal form reveals some information on the algebraic nature of the function.

For instance, if we assume that the deterministic part of the leakage δ is not an (algebraic) complex function (*e.g.* an AES) – which is a relatively common assumption – we can hope that φ will not be an (algebraic) complex function too. At the opposite, the function F is generally a cryptographic primitive *i.e.* an (algebraic) complex function (*e.g.* AES). In this case, $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for $\hat{k} \neq k$ still remains (algebraically) complex. In other terms, F is generally a highly non-linear function designed to be indistinguishable from a random function. At the opposite the function φ is

assumed to be a simple function resulting from the manipulation of a data (typically, a linear combination of the bits of the manipulated function). In this case, the algebraic normal form of φ will contain several zero coefficients whereas $\varphi \circ F_k \circ F_{\hat{k}}^{-1}$ for $\hat{k} \neq k$ will not.

Moreover, taking a closer look at φ, in the second order case, we noticed that it directly depends on δ and the masking scheme (Eq. (7.5)). Thus the algebraic form of φ can bring information about the masking scheme. Namely using a full basis linear regression can also serve as a reverse engineering tool to identify a masking scheme and some leakage properties.

Eventually, in an attacker point of view, the masking scheme is important and the attack must be adapted in consequence. For instance, the number of zero coefficients can be a good discriminating tool whereas in other conditions it will not.

This way of discriminating needs a deeper analysis. At the moment only few experiments reports have been done and are recorded as-is in App. C

7-4 To infinity... and beyond

In Sect. 7-2 we have treated the special case of second-order side channel attacks and we have exhibited a relevant way of exploiting linear regression with a link to a maximum likelihood approach (Sect. 7-2.3.2). This method cannot be extended to higher order as easily as previously done. Namely a multivariate Gaussian approximation is still parameterized by the mean vector and the covariance matrix (3.2) but the covariance matrix only contains covariance between two leakage points which is, by definition of the masking countermeasure, independent of the key.

Indeed, to our knowledge, no special analysis have been done on third and higher order SCA. Currently only MIA is multivariate and still face estimations difficulties [22]; otherwise univariate attacks with a combination function are used. Some new directions have already been mentioned that need deeper analysis. For instance computing the parameters of the Gaussian mixture using the EM algorithm [58] or using optimization algorithm such as the gradient descent [87]. In the same way as the maximum likelihood approach, we can try to approximate

the Gaussian mixture by a more complicated set of function. Namely, instead of merging the mixture component into a multivariate Gaussian pdf, one can merge them into a mixture of two Gaussian distributions *etc.* Finally, high-order SCAs still need a lot of analysis. In a defender point of view to have relevant countermeasures in practice, and in a attacker point of view to have efficient attacks.

CHAPTER 8

Simulations and Experiments

In the previous chapters we have analysed from a theoretical point of view several univariate side channel Attacks and proposed an unified framework (Chap. 5 and Chap. 6) which has been extended to second-order (Sect. 7-2). In this chapter we put in practice these analyses to reinforce them. These experiments show the practicability of our attack and permit to quantify the gain w.r.t. the state of the art.

8-1 Univariate SCA

In the previous sections we have shown that common univariate side channel attacks based on a restrictive model are equivalent to a CPA. At the opposite, we have exhibited two pertinent ways of attacking where some constraints on the model can be relaxed. It involves as a distin-

guisher either AS-DPA($\hat{k}$) or linear regression techniques. In the following we aim at confronting our theoretical analyses with simulations in realistic scenarios. Simulation parameters are described below.

Attacks Target. The 8-bit output of the AES S-box, denoted by S, is targeted. Namely the variable Z_k in (4.2) [p. 46] satisfies:

$$Z_k = S(P \oplus k) \ , \tag{8.1}$$

where P is an 8-bit value known by the adversary.

Attack Types. We list below the attacks we have performed:

1. Single-bit DPA (SB-DPA)
2. All-Or-Nothing DPA (AON-DPA)
3. Generalized DPA (G-DPA)
4. Correlation Power Analysis (CPA)
5. Partition Power Analysis (PPA)
6. Absolute-Sum DPA (AS-DPA)
7. Regression Attack with $(v_{\hat{k}}[i])_{0 \leqslant i \leqslant 7}$ as basis functions (this corresponds to Assumption 4 with $d = 1$).

Attacks 1 to 5 are described in Sect 4-4 and attacks 5 and 6 are described in Sect 6-1.

Model Choice. We recall that AON-DPA, G-DPA, CPA and PPA require the choice of a model function m, whereas SB-DPA, AS-DPA and the regression attack do not (for the latter the basis function is fixed). In our simulation, we have assumed that the definition of the function $\delta(\cdot)$ in (4.2) is not known by the adversary and we thus systematically used the Hamming weight function when a model was required to perform the attack. Namely, in AON-DPA, G-DPA, CPA and PPA the model m satisfies:

$$\mathsf{m}(Z_{\hat{k}}) = \mathrm{HW}(Z_{\hat{k}}) = \sum_i Z_{\hat{k}}[i] \ . \tag{8.2}$$

This model choice is very classical and has been experimentally validated in several papers *e.g.*, [57]. Once the model function has been

specified, parameters (ω_0, ω_1) in AON-DPA and (Ω_0, Ω_1) in G-DPA still need to be chosen in order to determine the distinguishers defined in (4.4) and (4.5) respectively. We chose

$$(\omega_0, \omega_1) = (\min_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}}), \max_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}})) = (0, 8)$$

and if we denote by $\mathrm{med}_X f(X)$ the *median* of the sample $f(X)$ with respect to X, we chose

$$\begin{aligned}(\Omega_0, \Omega_1) = \\ ([\min_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}}); \operatorname*{med}_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}})[,] \operatorname*{med}_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}}); \max_{Z_{\hat{k}}} \mathsf{m}(Z_{\hat{k}})]) = \\ ([0;4[,]4;8]) \ . \quad (8.3)\end{aligned}$$

Note that this choice is optimal and exactly corresponds to the attacks performed by Messerges in his original papers [67, 69]. Additionally, we chose the coefficients α_i of the PPA distinguisher such that (5.7) is satisfied for the model function m defined in (8.2) (*i.e.*, $\mathsf{PPA}_{(\alpha_i)_i}(\hat{k}) = \widehat{\mathbb{E}}\big(L \cdot \mathrm{HW}(Z_{\hat{k}})\big)$).

Leakage Simulations. Leakages have been simulated in accordance with (4.2) [p. 46], with the noise variable B being a Gaussian random variable with mean 0 and standard deviation σ.

Remark 23. For instance, algorithmic noise can results from an hardware with a parallel S-boxes computation design and can thus have an high standard deviation (*e.g.* in case of AES, one S-box computation in parallel of the targeted S-box will bring an additive noise with a standard deviation of $\sqrt{2}$)

As explained in the following sections, we launched our attack simulations for different definitions of the function $\delta(\cdot)$ in (4.2), leading to two different scenarios:

- *Scenario* 1*:* we chose $\delta(\cdot)$ in (4.2) to be the Hamming weight function. Namely, the leakage variable L satisfies:

  $$L = \mathrm{HW}(Z_k) + B \ , \quad (8.4)$$

  In our attack settings, this first scenario is ideally suited for AON-DPA, G-DPA, CPA and PPA since the model function m used by the adversary exactly corresponds to the deterministic function $\delta(\cdot)$. It will be referred as the *perfect model* scenario.

- *Scenario* 2*:* we chose $\delta(\cdot)$ to be a linear combination of the $Z_{\hat{k}}[i]$ with randomly generated coefficients. Namely the leakage variable L satisfies:

$$L = \alpha_{-1} + \sum_{i=0}^{7} \alpha_i \cdot Z_k[i] + B \ , \tag{8.5}$$

  with coefficients $(\alpha_i)_{-1 \leqslant i \leqslant 7}$ uniformly picked in $[-1, 1]$. This scenario is used to observe the distinguishers behavior when the deterministic part of the leakage differs from the model used by the adversary. We restricted ourselves to functions $\delta(\cdot)$ that are linear combinations in $\mathbb{R}$ of the bit-coordinates of the targeted value $Z_{\hat{k}}$ *i.e.* as in Assumption 4 (p. 68) with $d = 1$. It will be referred as the *random linear leakage* scenario.

Remark 24. We do not restrict ourselves to Assumption 3 (p. 52). That is we do not ensure that the size of the plaintext sample is a multiple of 256. Nevertheless plaintexts are drawn from a uniform distribution.

Attack Efficiency. In the following, an attack is said to be *successful* if the good key is output by the attack, that is if the key corresponding to the first element in the score vector is the key used in the simulated cryptographic device. An attack is said to be *more efficient than* another if it needs less messages to achieve the same success rate. Success rate is measured over $1,000$ tries.

We report and analyse in next two sections our attack simulations results for Scenario 1 (Section 8-1.1) and Scenario 2 (Section 8-1.2).

8-1.1 Attack Results in the Perfect Model Scenario

In this section we assume that L satisfies (8.4). In Fig. 8-1-1, the number of messages needed to achieve a success rate of 90% is recorded for each attack mentionned before*. Note that a success rate threshold has been fixed at 90% but in this configuration each attack can reach 100%.

*We inform the reader that the curves are fitted with a fourth degree polynomial to ease the reading of the figure. Fitted curves permit to observe the general behavior. Raw data can be found in Appendix A.

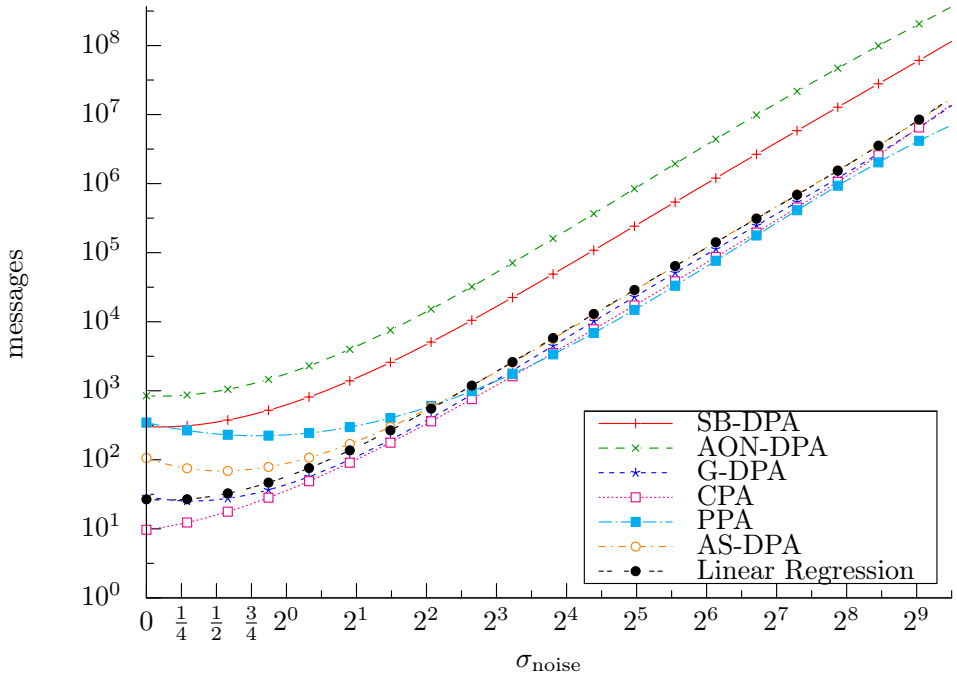


Fig. 8-1-1 – Evolution of the number of messages (y-axis logscaled) to achieve a success rate of 90% according to the noise standard deviation (x-axis logscaled) – Fitted curves.

Curves in Fig. 8-1-1 can be split in two parts depending on the noise standard deviation: the *oversampling* part, where a huge number of observations are needed to deal with the important noise effects and the *undersampling* part, where a small number of observations is sufficient. The two situations are analysed separately in the following. In both cases, the most relevant observations are listed and discussed.

Oversampling. When the noise standard deviation is strictly greater than 2^3, each distinguisher needs a large number of messages (greater than 500) to reach a success rate of 90%. In this case the curves have the same shape for each distinguisher, which is compliant with the asymptotical results in [65]. Our observations are detailled below:

- The efficiency curves of each attack have the same gradient. This suggests us that the noise similarly impacts the efficiency of the attacks.
- The curves corresponding to G-DPA, CPA, PPA, AS-DPA and the regression attack are stacked. Note that the logscaling implies that those attacks share *approximatively* the same efficiency and that none of them is emerging as better candidate than the others. In fact, in the perfect model scenario, the distinguishers corresponding to these attacks are equivalent to a maximum likelihood test and the attacks therefore perform in a similar (and optimal) way [65]. This pinpoints the equivalence between the distinguishers when the model function used in the model-based attacks (*i.e.*, AON-DPA, G-DPA, CPA and PPA) is optimal (*i.e.*, perfectly corresponds to the function $\delta(\cdot)$ in (4.2)).
- As expected, SB-DPA and AON-DPA are less powerful than the others (around 100 and 30 times less efficient than G-DPA, CPA, PPA, AS-DPA and the regression attack for the SB-DPA and the AON-DPA respectively). Indeed, by nature they do not exploit all the information contained in the leakage signal: in SB-DPA only one output bit is targeted over the 8 output bits of the AES, whereas the AON-DPA only exploits a limited part of the leakage measurements.

Remark 25. The good result of G-DPA can be surprizing as the involved model is not the Hamming weight model. The G-DPA model only takes two values -1 and 1 depending on the Hamming weight of the sensitive variable is lower than 4 or not. In fact the linear correlation between the

G-DPA model and the Hamming weight model is high (greater than 0.9). That implies an efficiency ratio of 1.2 (0.08 in a $\log_{10}$ scale) according to [63]. This explains why G-DPA's curve appears stacked with CPA's curve.

Undersampling. When the noise standard deviation is lower than 2^3, the number of messages needed to perform an attack is quite small (lower than 500). In this case, the statistical stability of the involved distinguisher plays a role. To better understand how the different attacks perform in this context we redrew in Fig. 8-1-2 the curves with a thiner resolution than in Fig. 8-1-1. We detail our observations below:

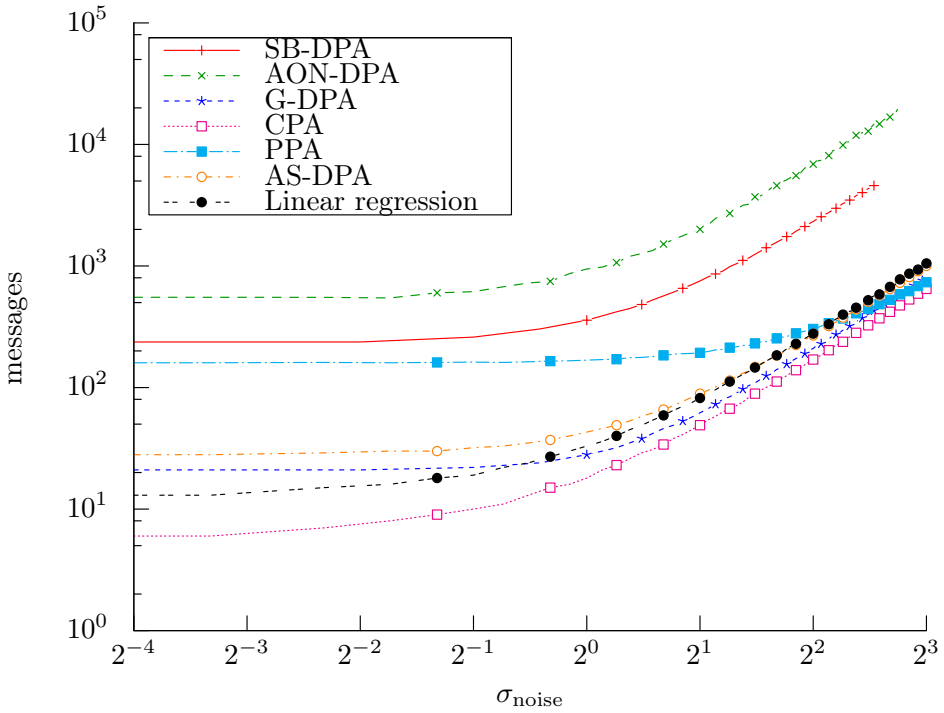


Fig. 8-1-2 – Evolution of the number of messages (y-axis logscaled) to achieve a success rate of 90% according to the noise standard deviation (x-axis logscaled) – Higher resolution.

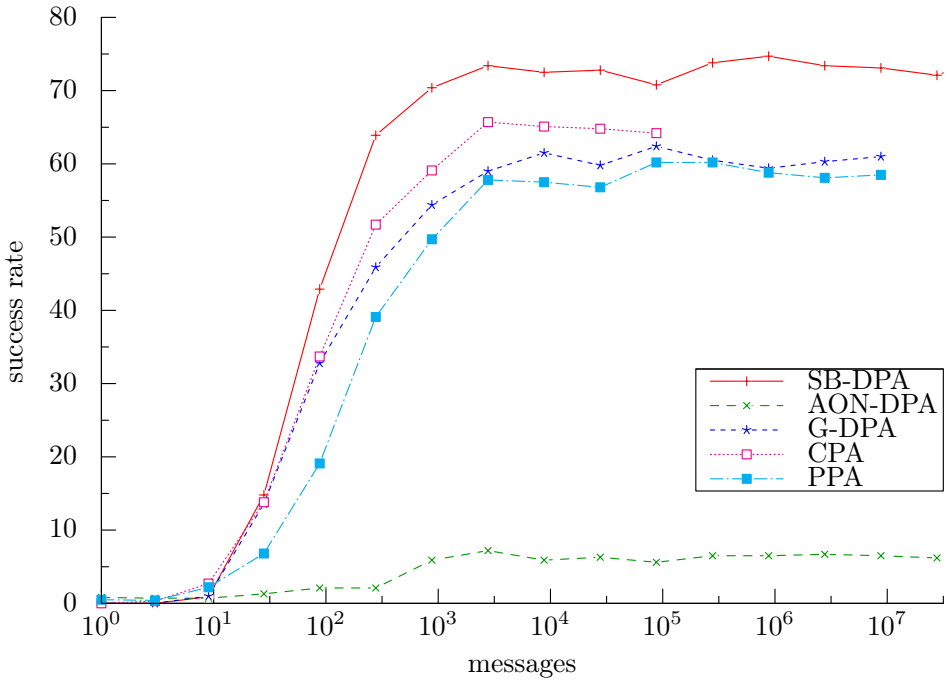
- An important efficiency difference occurs between the CPA, the DPAs and the PPA. For example with a noise standard deviation of 1, CPA needs only 30 messages to reach a success rate of 90%, whereas PPA needs 280 messages to achieve the same threshold.

- CPA is the most efficient attack. This confirms that Pearson's coefficient is the good tool to measure a linear correlation.
- In comparison, the PPA is much less efficient than the CPA (and even also than the DPAs). This result was actually expected. Indeed, *centering* the leakage and the model random variables (*i.e.* computing the value $\widehat{\mathbb{E}}\left(L \cdot \mathsf{m}(Z_{\hat{k}})\right) - \widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(\mathsf{m}(Z_{\hat{k}})\right)$ in place of the value $\widehat{\mathbb{E}}\left(L \cdot \mathsf{m}(Z_{\hat{k}})\right)$ in the PPA attack) and then *normalizing* the centered mean by the standard deviations of the random variables (*i.e.* dividing $\widehat{\mathbb{E}}\left(L \cdot \mathsf{m}(Z_{\hat{k}})\right) - \widehat{\mathbb{E}}(L)\widehat{\mathbb{E}}\left(\mathsf{m}(Z_{\hat{k}})\right)$ by $\widehat{\sigma}(L) \cdot \widehat{\sigma}\left(\mathsf{m}(Z_{\hat{k}})\right)$ thus getting the CPA distinguisher $\mathsf{CPA}(\hat{k})$) is useful to reduce the linear dependency estimation errors when the number of observations is small (*i.e.* undersampling), which is the case when the attacks are performed for a small amount of noise.
- G-DPA, CPA and PPA are more efficient than AS-DPA and regression attacks. It may be noted that this situation is the opposite of the one occuring in the oversampling case.

Hence, our results corroborate our theoretical analysis: the SB-DPA and the AON-DPA are less efficient than the other simulated attacks for any amount of noise in the leakage. This highlights the fact that targeting a subspace of the model (*i.e.*, a single bit over eight or targeting 2 values over 256) is suboptimal when the adversary uses a model that well corresponds to the function $\delta(\cdot)$ (G-DPA, CPA and PPA) or when an AS-DPA or a regression attack is performed. Whatever the signal-to-noise ratio, CPA is always the best attack. However its efficiency is very close to that of G-DPA and PPA when the noise standard deviation reaches the threshold 4. Actually CPA is mainly better than the other tested attacks when the leakage is not very noisy (*i.e.*, when the noise standard deviation is between 0 and 4). Eventually, it can be noted that the efficiency of AS-DPA and linear regression attack tends to be close to that of the CPA while the perfect model scenario is optimally suited for CPA.

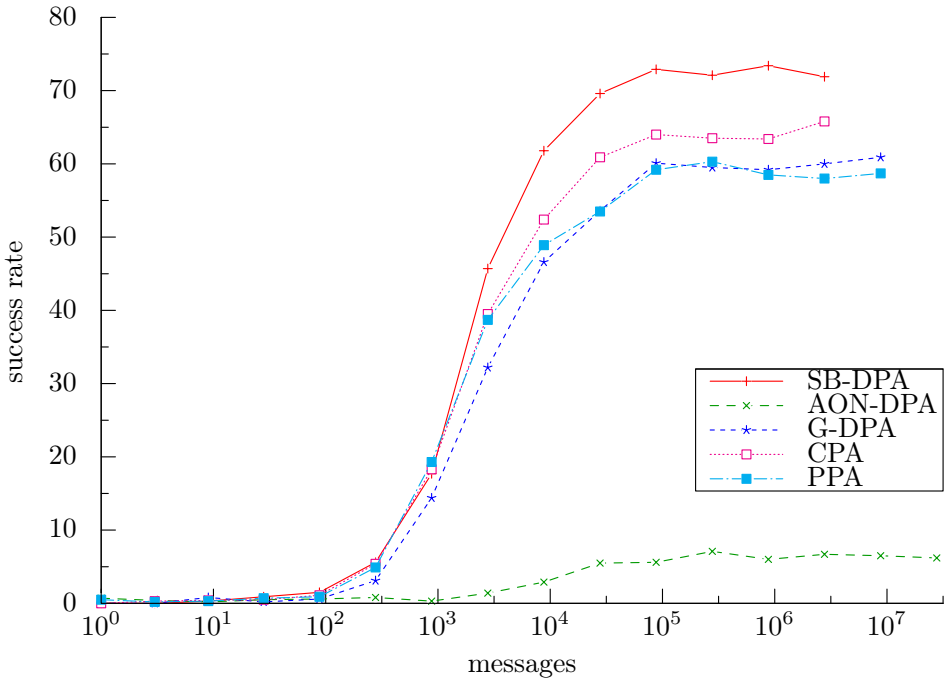
8-1.2 Attack Results in the Random Linear Leakage Scenario

In this section we assume that L satisfies (8.5). In Fig. 8-1-3, we recorded the success rate for different numbers of messages and for different values of noise standard deviation.



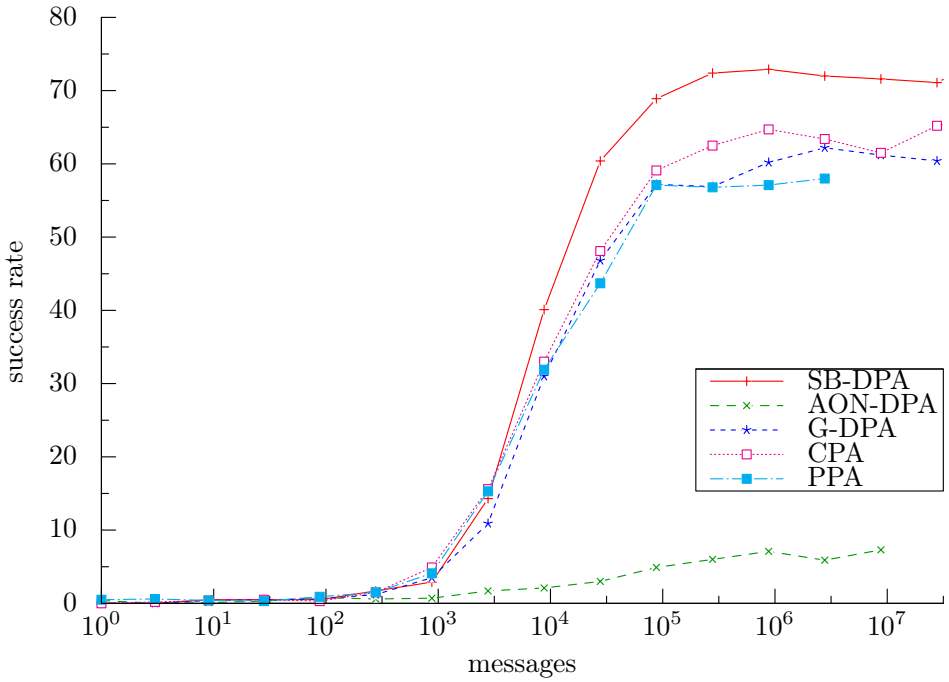
(a) No noise

Fig. 8-1-3 – Evolution of the success rate (1,000 tries) for different numbers of messages and according to some critic noise standard deviations – whole data can be found in Appendix A.



(b) Mid noise (4.00)

Fig. 8-1-3 – Evolution of the success rate (1,000 tries) for different numbers of messages and according to some critic noise standard deviations – whole data can be found in Appendix A.



(c) High noise (8.00)

Fig. 8-1-3 – Evolution of the success rate (1,000 tries) for different numbers of messages and according to some critic noise standard deviations – whole data can be found in Appendix A.

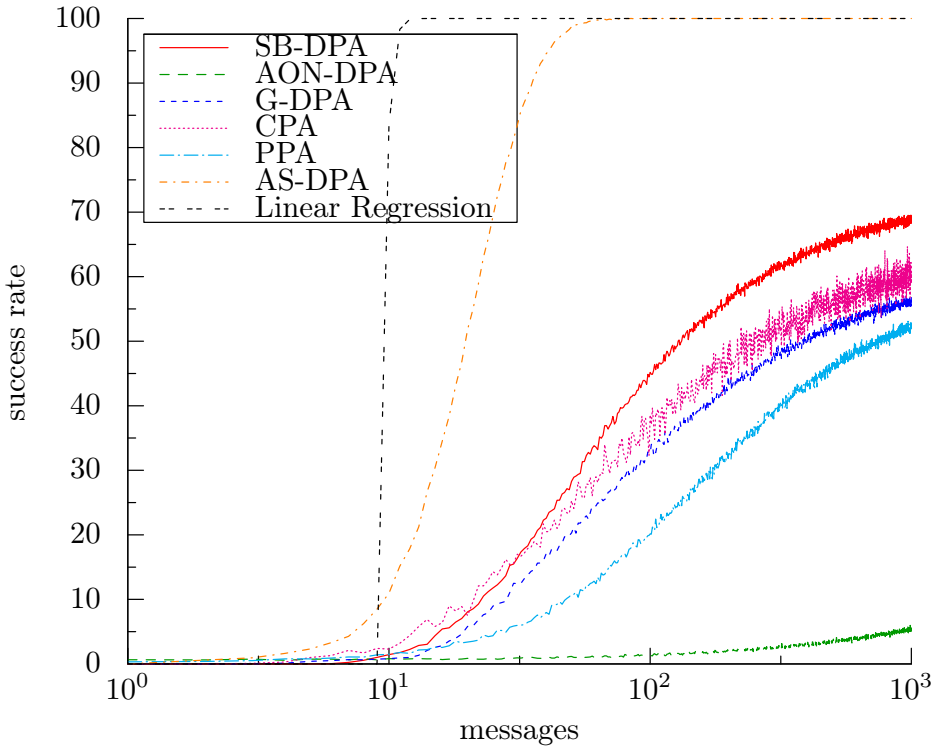
Observations are reported below. As in the perfect model scenario we can split our observations in two parts.

Oversampling. When the number of messages available is greater than approximately $10^5 \times \sigma^2$, the curves have the same shape for each distinguisher but contrary to what happened in the perfect model scenario, all the attacks do not reach a success rate of 100%.

- The maximum success rate achieved by the model-based attacks is lower than 75% (*e.g.*, CPA achieves a success rate of 62% while G-DPA and PPA are still less efficient with a success rate limit of 58%) independent of the noise standard deviation. In other terms, for some linear functions $\delta(\cdot)$, those attacks do not succeed in discriminating the good key candidate when the Hamming weight function is involved as model. In Appendix 8-1.5, we give a theoretical explanation of the CPA ineffectiveness for some linear functions $\delta(\cdot)$ and we argue that it is related to the algebraic properties of the S-box S that is targeted.
- AON-DPA only reaches a maximal success rate of 6% which is very low compared to the others. A possible explanation for the AON-DPA poor effectiveness resides in the fact that the design of the sets Ω_0 and Ω_1 under the hypothesis $\mathsf{m} = \mathrm{HW}$ is not relevant when $\delta(\cdot)$ is far away from the Hamming weight function
- At the opposite SB-DPA reaches a maximal success rate of 72% which is better than CPA. This observation is not surprising since SB-DPA targets only one bit (independently of the model choice) over eight, which lowers the impact of the model choice on the remaining seven bits.

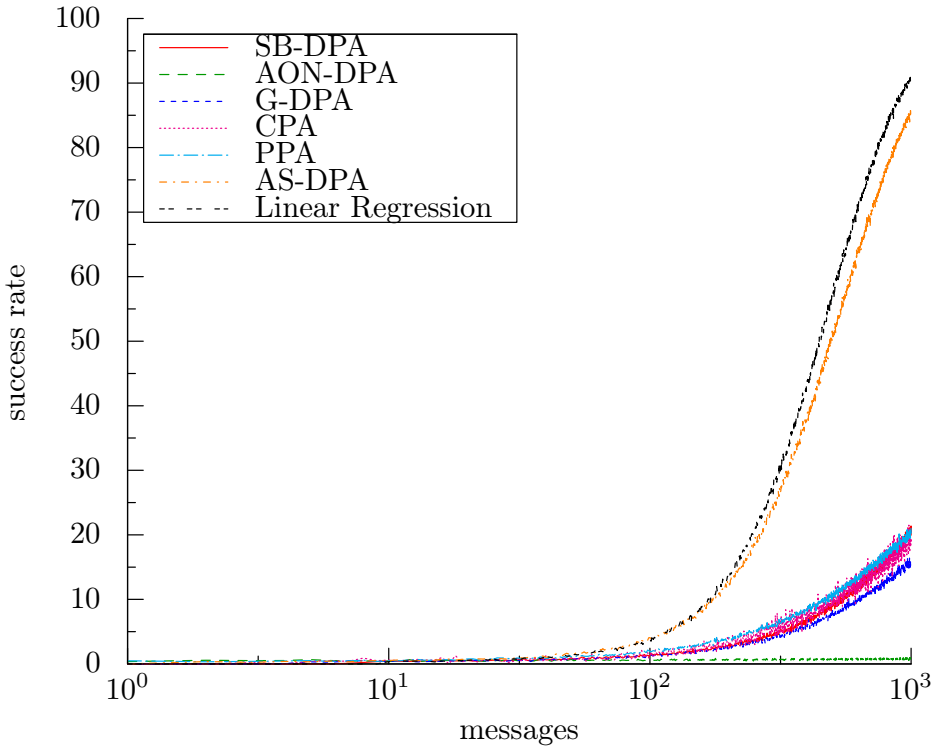
Undersampling. Let us focus on the critic values when a small number of messages is involved in the attack (lower than 500). In this case, the statistical stability of the involved distinguisher plays a role. To better understand how the different attacks perform in this context we redrew in Fig. 8-1-4 the curves with a thiner resolution than in Fig. 8-1-3.

Our observations are detailled below:



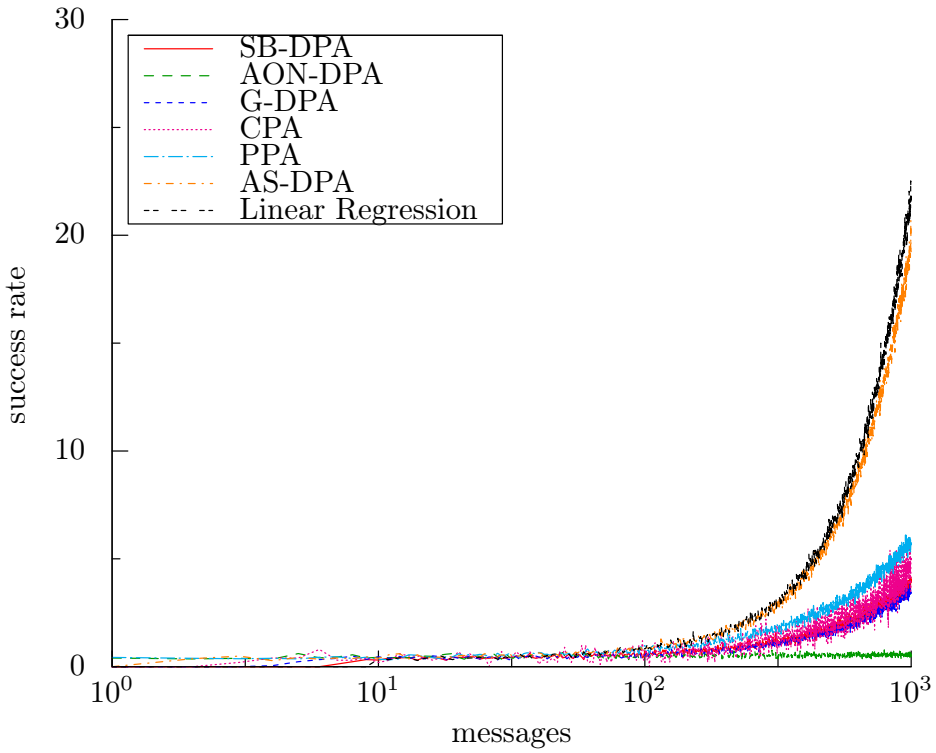
(a) No noise

Fig. 8-1-4 – Evolution of the success rate (10,000 tries) for number of messages from 1 to 1,000 with some noise values.



(b) Mid noise (4.00)

Fig. 8-1-4 – Evolution of the success rate (10,000 tries) for number of messages from 1 to 1,000 with some noise values.



(c) High noise (8.00)

Fig. 8-1-4 – Evolution of the success rate (10,000 tries) for number of messages from 1 to 1,000 with some noise values.

- In this situation, each distinguisher has the same ranking as in oversampling.
- G-DPA, CPA and PPA are relatively less efficient than in the perfect model scenario. Indeed, in the latter model scenario they are more efficient than AS-DPA and regression attack which is not the case here.
- SB-DPA and AON-DPA still have a different behavior than other model based attacks due to the use of a suboptimal model (with respect to the attacker choice in (8.2)).

The impact of the noise on the attacks efficiency in our linear random model scenario is very close to what we observed in the perfect model context. Namely the maximal success rate is the same whatever the noise deviation but more messages are needed to achieve it. In fact, we confirm the theoretical analysis in [63], where the author shows that doubling the noise deviation just increases the number of needed messages by its square root to reach the same success rate.

Among the attacks we simulated in the random model scenario, the linear regression attack and the AS-DPA are clearly the most efficient ones and they are the only ones that reach a success rate of 100%.

8-1.3 Attacks Experiments in Real Life

In previous sections, we have confronted our theoretical analyses with simulations in realistic scenarios. Two attacks emerged, the CPA and the linear regression. In the following, we aim to confront our results against real measurements. Thus we only focus on CPA and linear regression attacks. Attack parameters are described below:

Attacks Target. The 8-bit output of the AES S-box, denoted by S, is targeted. Namely the variable Z_k in (4.2) satisfies:

$$Z_k = S(P \oplus k) \ , \tag{8.6}$$

where P corresponds to an 8-bit value known by the adversary.

Attack Types. We list below the attacks we have performed:

- CPA with m satisfying (8.2) (Hamming weight model).
- Regression Attack with $\mathscr{B}_{\text{lin}} = (v_{\hat{k}}[i])_{0 \leqslant i \leqslant 7}$ as basis functions (Assumption 4 with $d = 1$).
- Regression Attack with $\mathscr{B}_{\text{quad}} = (v_{\hat{k}}[i] \cdot v_{\hat{k}}[j])_{0 \leqslant i \leqslant j \leqslant 7}$ as basis functions (Assumption 4 with $d = 2$).

Leakage Measurements. A sample of 400,000 power consumption measures have been done on a 8051 8-bit micro-controller. In each measurement curve, the part related to the manipulation of Z_k is composed of 200 points. We suppose the curves to be synchronized (a glitch is used to be synchronized at the beginning of the manipulation processing). Before mounting the attacks, a pre-processing step has been performed on the curves to determine the most pertinent *point of interest* for each attack. By definition, this point is the one among the 200 points per curve that optimizes the attack efficiency. As argued in Section 5-7, it corresponds for the CPA to the point when the error resulting from the approximation of the leakage by the attack model (*i.e.* the Hamming weight function) is minimum. For the regression attacks, the point of interest is the point on which the error resulting from the approximation of the leakage by a linear (resp. quadratic) combination of the coordinates of the manipulated variable is minimum. During the pre-processing, we have used the fact that we knew the values $v_{k,i}$ manipulated by the device. Even if this does not correspond to a real life adversary, pre-processing in this context allows us to determine the time/point when an attack performed by an adversary with no such a knowledge is the most efficient. In the following, we sum-up the pre-processing step for the three attacks.

- CPA: the coefficient $\text{CPA}_{\text{HW}}(k)^2$ has been estimated for each of the 200 points of the curve – the estimation being performed for a sample of size 400,000 – to determine the best attack time location.
- Regression Attack: a model function m_{lin} (resp. m_{quad}) corresponding to the correct k has been computed for each of the 200 points of the curve, the estimation being performed for a sample of size 400,000. Then, 200 determination coefficients R^2 have been performed (one for each model M_k and the corresponding leakage point) to determine the best attack time location corresponding to the basis functions $\mathscr{B}_{\text{lin}}$ (resp. $\mathscr{B}_{\text{quad}}$)

Figure 8-1-5 illustrates the results of the pre-processing step for each attack and each of the 200 points.

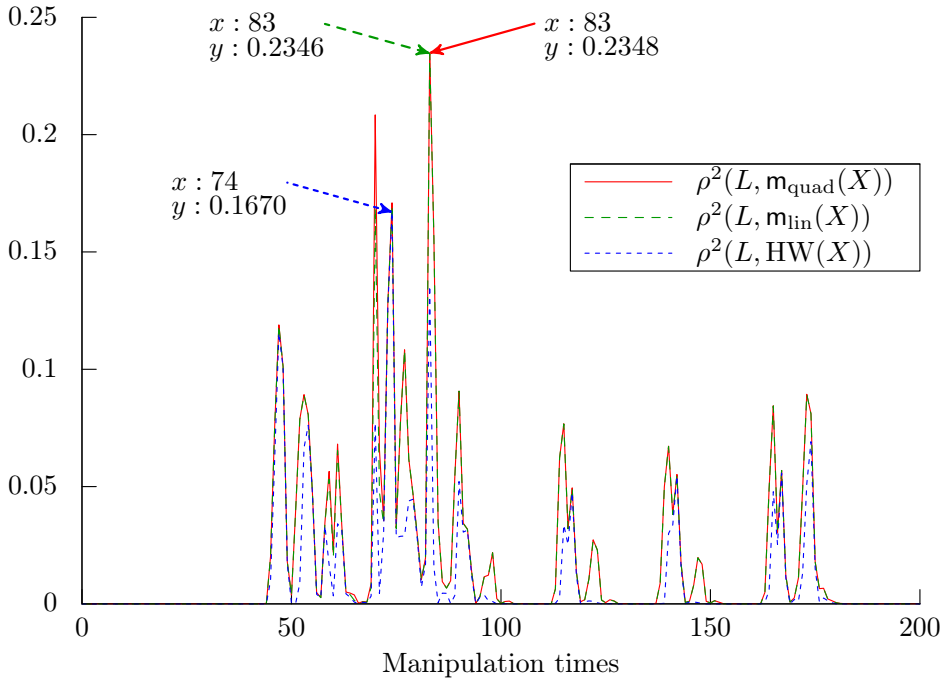


Fig. 8-1-5 – Characterisation Timing Diagram. Max values are pinpointed by an arrow.

For the attack comparisons, only the point of interest resulting in the maximal distinguishing value has been considered for each attack.

Attack Comparaison. For each attack, the distinguishing coefficient (in y-axis) has been computed for each key candidate and for a given (increasing) number of power traces (x-axis). We recorded the minimal number of messages needed to have the real key ranked first (*i.e.* emerging from others). Results are drawn in Fig. 8-1-6, 8-1-7 and 8-1-8. As expected linear regression with linear basis is clearly more efficient than CPA *i.e.*, a lower number of messages is required for the real key to emerge (68 messages is sufficient for the first one while 95 at least are needed for the CPA). As expected, the linear regression with quadratic basis needs more messages. In fact the information contained in the quadratic part of the leakage is not enough to compensate for the increase of noise resulting from the multiplication of leakage points (which

is necessary to process the linear regression). Moreover the quadratic regression has to build a larger model (*i.e.*, from a larger basis) from data. We can remark that even with quadratic basis, the minimum number of messages needed to discriminate the real key is still very close to the one for CPA (≈ 95).

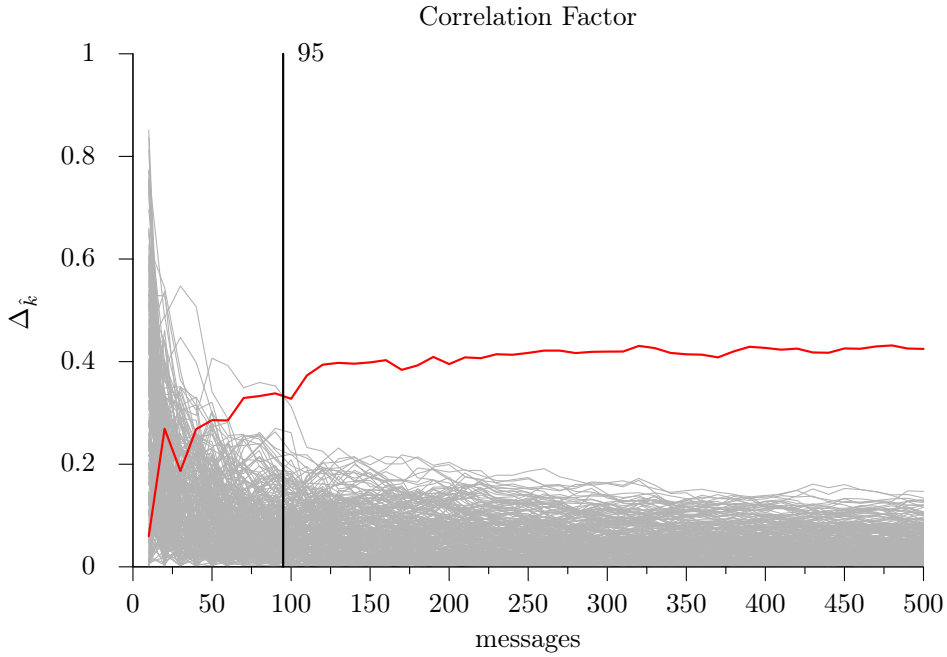


Fig. 8-1-6 – Evolution of the distinguishing value (y-axis) with the number of messages (x-axis) for all key candidates for CPA. The curve of the real key used in the device is plotted in red.

8-1.4 Conclusion on the Attack Simulations and Experiments

When the chosen model exactly corresponds to the leakage function (perfect model case), each distinguisher reveals the key and the CPA and regression attacks are among the most efficient ones (actually except SB-DPA and AON-DPA all tested attacks have an equivalent efficiency when the noise increases). Nevertheless in case of undersampling, CPA is ranked first. This can be explained by the fact that the linear regression attack has to rebuild the model from data while CPA is directly

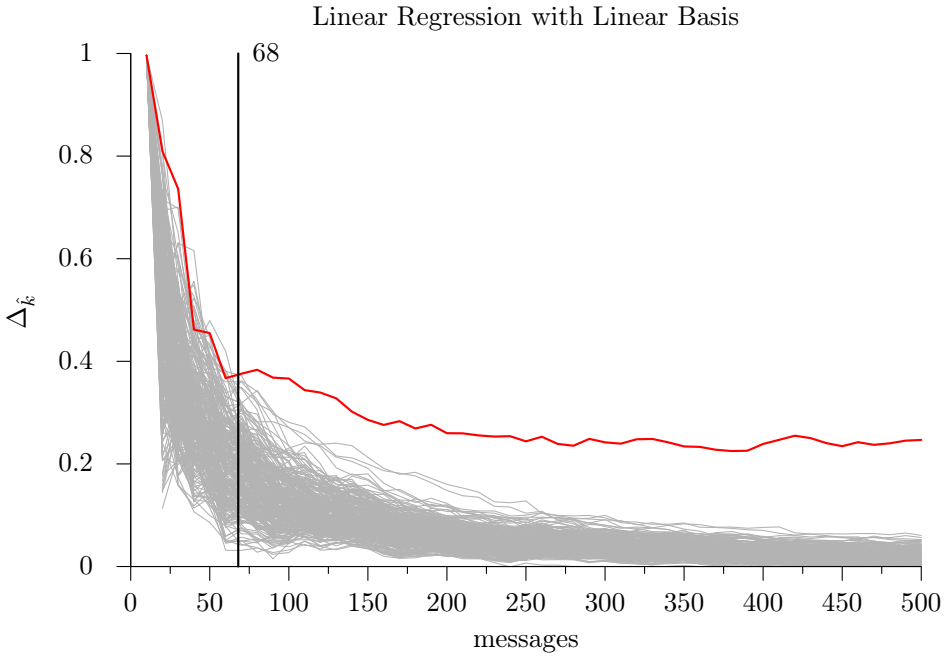


Fig. 8-1-7 – Evolution of the distinguishing value (y-axis) with the number of messages (x-axis) for all key candidates for linear regression with linear basis. The curve of the real key used in the device is plotted in red.

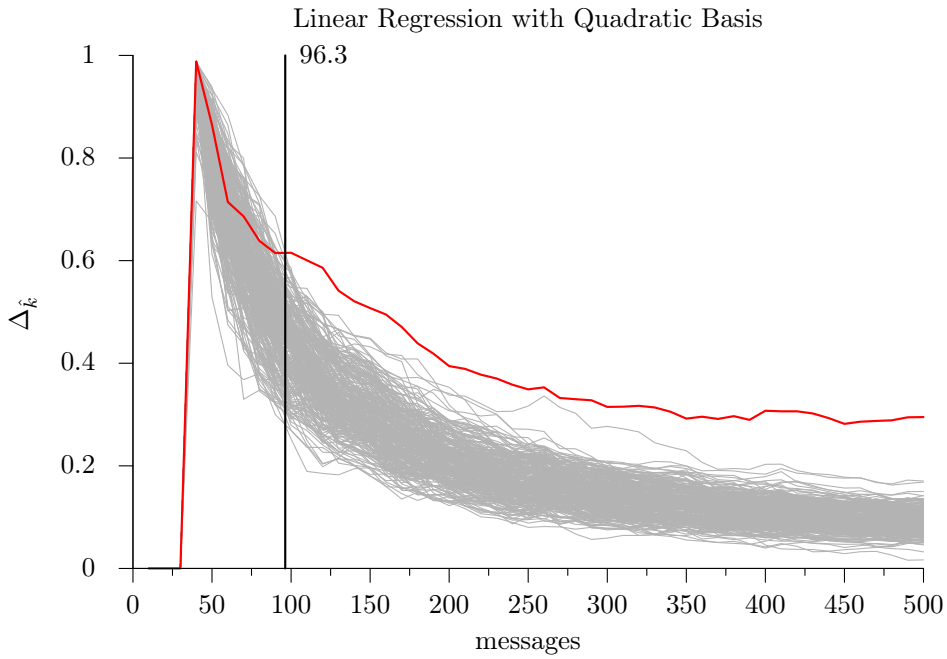


Fig. 8-1-8 – Evolution of the distinguishing value (y-axis) with the number of messages (x-axis) for all key candidates for linear regression with quadratic basis. The curve of the real key used in the device is plotted in red.

provided with the optimal model function and uses the observations only to corroborate a linear dependency.

When the model is unknown, the linear regression attack and the AS-DPA always succeed in revealing the key. they both are moreover more efficient than the model-based attacks. Nevertheless, collating both, the linear regression is always better than AS-DPA. That is, at a cost of a little computational overhead, linear regression attack shall be preferred to the other distinguishers.

Notice that a more sophisticated model or basis did not necessarily lead to better distinguishability as it will bring more noise than useful information.

Finally, if one has a good linear approximation of $\delta(\cdot)$ then CPA is an optimal way to perform an attack. In other cases, the linear regression attack will always perform better.

8-1.5 Why CPA can fail?

This section aims at explaining why the CPA fails in discriminating the correct key for some linear leakage models. Before starting our discussion, let us first have a look on the definition of the CPA distinguisher (4.8). Under Assumption 3, it involves standard deviations that tend to be independent of the key hypothesis when the sample size increases. As a consequence, the distinguisher in (4.8) discriminates key hypotheses in a similar way as the covariance $\mathrm{cov}\left(L\,,M_{\hat{k}}\right)$. Explaining the CPA failure hence amounts to explain the covariance failure when involved as a key-distinguisher.

Our analysis will be merely related to the following proposition.
Proposition 6. Let f and g be two Boolean functions defined over $\mathbb{F}_{2^n}$. If f and g are balanced, then we have:

$$\mathrm{cov}(f\,,g)=\frac{1}{4}W(f\oplus g)\ , \tag{8.7}$$

where $W(f\oplus g)$ denotes the value $2^{-n}\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)\oplus g(x)}$.

Proof. The result is a direct consequence of the following equality:

$$f+g=f\oplus g+2fg\ . \tag{8.8}$$

◇

Due to Assumption 2 and the fact that the leakage satisfies (4.2), we recall that $\mathrm{cov}\left(L\,,M_{\hat{k}}\right)$ equals $\mathrm{cov}\left(\delta(Z_k),M_{\hat{k}}\right)$ independent of the targeted key k and the key hypothesis $\hat{k}$. If the model function m is the Hamming weight and if $\delta(\cdot)$ satisfies (6.1) with $d=1$ (*i.e.* Assumption 4), then $\delta(Z_k)$ and $M_{\hat{k}}=\mathsf{m}(V_{\hat{k}})$ respectively equal $\alpha_{-1}+\sum_i \alpha_i Z_k[i]$ and $\sum_j Z_{\hat{k}}[j]$. Under those two assumptions, we hence get:

$$\mathrm{cov}\left(L\,,M_{\hat{k}}\right)=\mathrm{cov}\left(\alpha_{-1}+\sum_i \alpha_i Z_k[i],\sum_j Z_{\hat{k}}[j]\right)\ , \tag{8.9}$$

i.e.,

$$\mathrm{cov}\left(L\,,M_{\hat{k}}\right)=\sum_i \alpha_i\left(\sum_j \mathrm{cov}\left(Z_k[i],Z_{\hat{k}}[j]\right)\right)\ . \tag{8.10}$$

Since functions $Z_k[i]$ and $Z_{\hat{k}}[j]$ are both balanced under Assumption 3, Proposition 6 can be applied to develop the covariances in (8.10):

$$\mathrm{cov}\left(L\,,M_{\hat{k}}\right)=\frac{1}{4}\sum_i \alpha_i\sum_j W(Z_k[i]\oplus Z_{\hat{k}}[j])\ , \tag{8.11}$$

That is we have

$$\mathrm{cov}\left(L\,,M_{\hat{k}}\right)=\frac{1}{4}\sum_i \alpha_i w_i(k,\hat{k}) \tag{8.12}$$

after denoting the term $\sum_j W(Z_k[i]\oplus Z_{\hat{k}}[j])$ by $w_i(k,\hat{k})$.

Let us study (8.12) when the correct key hypothesis is tested, *i.e.*, when $\hat{k}$ equals k. As Z_k is balanced, the term $W(Z_k[i]\oplus Z_{\hat{k}}[j])$ is always zero except for $i=j$ where it equals 1. Equation (8.11) can thus be rewritten as:

$$\mathrm{cov}\left(L\,,M_{\hat{k}}\right)=\frac{1}{4}\sum_i \alpha_i\ . \tag{8.13}$$

In view of (8.13), $\mathrm{argmax}_{\hat{k}}\left|\mathrm{cov}\left(L\,,M_{\hat{k}}\right)\right|$ is not equal to the expected key (*i.e.*, the covariance distinguisher fails at discriminating the correct key), if there exists at least one key hypothesis $\hat{k}\neq k$ such that $Z_{\hat{k}}$ satisfies:

$$\left|\sum_i \alpha_i\right|<\left|\sum_i \alpha_i w_i(k,\hat{k})\right|\ . \tag{8.14}$$

Actually, for the type of variables Z_k involved in the attack simulations reported in Section 8-1.2, the condition (8.14) is often satisfied. Indeed, in those simulations, Z_k corresponds to the output of the AES S-box S parameterized by the key k. Namely, Z_k takes the form $S(X \oplus k)$. In this context, $Z_k[i] \oplus Z_{\hat{k}}[j]$ corresponds to the function $X \mapsto S_i(X \oplus k) \oplus S_j(X \oplus \hat{k})$, where $S_1, \ldots, S_n$ denote the boolean coordinate functions of S. When X has a uniform distribution, the latter function shares the same distribution as the function $S^a_{i,j}$ defined by $S^a_{i,j}(X) = S_i(X \oplus a) \oplus S_j(X)$, with a denoting $k \oplus \hat{k}$. After denoting by $w_i(a)$ the sum $\sum_j W(S^a_{i,j})$, we therefore conclude on the equivalency between (8.14) and

$$\left| \sum_i \alpha_i \right| < \left| \sum_i \alpha_i w_i(a) \right| \quad . \tag{8.15}$$

Since coefficients $(\alpha_i)_i$ and $(w_i(a))_{i,a}$ have an amplitude upper bounded by 1 and the right hand side of (8.15) is itself upper-bounded by the value $\min(\sum_i |\alpha_i|, \sum_i |w_i(a)|)$, we deduce two sufficient conditions for (8.15) to be never satisfied for $a \neq 0$ (*i.e.* for another key candidate than the correct one):

- All the terms α_i have the same sign.
- $\max_{a \neq 0} \sum_i |w_i(a)|$ is lower than or equal to $\sum_i |\alpha_i|$.

The first sufficient condition condition is device dependent and the second condition relies on the S-box properties. For the AES S-box for instance, it can be checked that $\max_a \sum_i |w_i(a)|$ equals 1.9375 for $a = 53$ (the absolute sum of $w_i(a)$ has been computed for AES for each 256 values of a, results can be found in the Appendix B). Thus, if $\sum_i |\alpha_i|$ is greater than 1.9375, then (8.15) cannot be satisfied for a value $a \neq 0$ and we deduce that the CPA is theoretically able to succeed in this case. Otherwise, when $\sum_i |\alpha_i| < 1.9375$ then for some $a \neq 0$ the distinguisher value is greater than the one got for the good hypothesis and the attack thus fails.

In the following, we give an example of such a case (*i.e.*, when CPA fail to discriminate the good key):
Example 2. Let $\{\alpha_i\}_{0 \leqslant i < 7} = \{0.5, 0.2, -0.5, 0.2, -0.5, 1, -0.8, 0.5\}$ be the coefficients of the leakage model, that is for every $x \in \mathbb{F}_{2^8}$:

$$\delta(x) = 0.5x_0 + 0.2x_1 - 0.5x_2 + 0.2x_3 - 0.5x_4 + x_5 - 0.8x_6 + 0.5x_7$$

where $(x_7,\ldots,x_0)$ is the binary decomposition of x. In this case we have $|\sum_i \alpha_i| = 0.6 < 1.9375$ and at least ten values (see Table 8-1-1) of a are such that $|\sum_i \alpha_i w_i(a)| > |\sum_i \alpha_i|$.

Tab. 8-1-1 – Eleven highest values of $|\sum_i \alpha_i w_i(a)|$ obtained in Example 2.

a	$\lvert\sum_i \alpha_i w_i(a)\rvert$
101	0.8875
228	0.84375
109	0.775
25	0.7625
30	0.721875
176	0.66875
19	0.6578125
151	0.6515625
66	0.634375
158	0.6203125
0	0.6

8-2 Second-Order Side Channel Attack: Application on Masking Schemes

In Sect. 7-2, we exhibited a way to attack a masked implementation by using linear regression techniques. In the following, we aim at confronting our analyses with simulations in realistic scenarios (Sect. 8-2.1 and 8-2.2) and experiments (Sect. 8-2.3). To ease the comparison, several attack parameters are considered: (1) the underlying masking scheme used to protect the sensitive variable, (2) the distinguisher involved in the key discrimination, (3) some related parameters to customize the attack, (4) the nature of the leakage (simulation or real curves) and (5) the attack efficiency (number of messages, *etc.*).

Remark 26. Our main purpose is to compare the new attack with the CPA techniques which are the most widely used in practice. However, in order to have an analysis as exhaustive as possible, we also implemented second-order MIA attacks. Among the different techniques to process the MIA, we chose to implement the one which is based on histogram since it seems to be the most efficient in practice [22]. Further works may

consist in deeper comparing the new attack with all the various MIA techniques [112] and also with the recently introduced attacks based on Kolmogorov-Smirnov distance estimator [106, 113]. Those attacks indeed also aim to target masked implementations when the leakage has unpredictible behaviour.

Attack Target. The attacks exploit the leakage related to the manipulation of two shares that jointly depend on a sensitive variable Z satisfying

$$Z = F_k(X) = F(X \oplus k) \ , \tag{8.16}$$

where F denotes the AES S-box and where X corresponds to a 8-bit uniformly distributed random value known by the adversary and F denotes the AES S-box. Depending on the underlying masking scheme, the definition of the two shares differ. The following masking schemes are considered in our attacks:

1. 1^{st}-order Boolean masking: the operation $\star$ in (7.2) is the bitwise addition over $\mathbb{F}_2^8$. The two shares are $Z \oplus V$ and V, with V a uniformly distributed random variable independent of Z.

2. 1^{st}-order arithmetic masking: the operation $\star$ in (7.2) is the modular addition over $\mathbb{Z}/256\mathbb{Z}$. The two shares are $Z + V \mod 256$ and V, with V a uniformly distributed random variable independent of Z.

Leakage Simulations. Leakages have been simulated in accordance to (7.2) for different definitions of $\delta(\cdot)$, leading to the three following scenarios:

Scenario 1 *(Hamming Weight Leakage):* Equation (7.2) becomes:

$$L_1 = \underbrace{\mathrm{HW}(Z \star V)}_{\delta(Z \star V)} + B_1 \quad \text{and} \quad L_2 = \underbrace{\mathrm{HW}(V)}_{\delta(V)} + B_2 \ . \tag{8.17}$$

In our attack settings, this first scenario is ideally suited for CPA since the model used by the adversary exactly corresponds to the deterministic function $\delta(\cdot)$.

Scenario 2 *(Linear Leakage):* Equation (7.2) becomes:

$$\begin{aligned} L_1 &= \underbrace{\alpha_0 + \sum_{i=1}^{8} \alpha_i \cdot (Z \star V)[i]}_{\delta(Z \star V)} + B_1 \quad \text{and} \\ L_2 &= \underbrace{\alpha_0 + \sum_{i=1}^{8} \alpha_i \cdot V[i]}_{\delta(V)} + B_2 \ , \end{aligned} \tag{8.18}$$

with coefficients $(\alpha_i)_{0 \leqslant i \leqslant 8}$ uniformly picked from $[-1, 1]$. This scenario is used to observe the distinguishers behaviour when the deterministic part of the leakage differs from the model used by the adversary. We restricted ourselves to functions $\delta(\cdot)$ that are linear combinations in $\mathbb{R}$ of the bit-coordinates of the shared values.

Scenario 3 *(Quadratic Leakage):* Equation (7.2) becomes:

$$\begin{aligned} L_1 &= \delta(Z \star V) + B_1 \\ &= \alpha_0 + \sum_{i=1}^{8} \alpha_i \cdot (Z \star V)[i] \\ &\quad + \sum_{\substack{i_1, i_2 = 1 \\ i_1 < i_2}}^{8} \alpha_{i_1, i_2} \cdot (Z \star V)[i_1] \cdot (Z \star V)[i_2] + B_1 \\ L_2 &= \delta(V) + B_2 \\ &= \alpha_0 + \sum_{i=1}^{8} \alpha_i \cdot V[i] \\ &\quad + \sum_{\substack{i_1, i_2 = 1 \\ i_1 < i_2}}^{8} \alpha_{i_1, i_2} \cdot V[i_1] \cdot V[i_2] + B_2 \ , \end{aligned} \tag{8.19}$$

with coefficients $(\alpha_i)_{0 \leqslant i \leqslant 36}$ uniformly picked from $[-1, 1]$. This scenario is used to observe the distinguishers behaviour when the deterministic part of the leakage differs in degree from the model used by the adversary. We restricted ourselves to functions $\delta(\cdot)$ that are quadratic combinations in $\mathbb{R}$ of the bit-coordinates of the shared values.

Leakage Measurements. The details about the leakage used in experiments (For instance the choice of points of interest) have been confined to a dedicated section (see Sect.8-2.3).

Attack Distinguisher.

1. Correlation Power Analysis (CPA). To discriminate the key candidates, those attacks approximate $\rho(\mathscr{C}(L_1, L_2), \mathsf{m}(F_{\hat{k}}(X)))$, where $\mathscr{C}(\cdot)$ is a combining function from $\mathbb{R}^2$ to $\mathbb{R}$ and m is a model function deduced from $\mathscr{C}(\cdot)$ and an hypothesis on $\delta(\cdot)$. A second-order CPA with model m is denoted by $\mathrm{CPA}_{\mathsf{m}}$
2. Linear Regression (LR) is used as described in Sect. 7-2.
3. Mutual Information Analysis (MIA) with Hamming weight model and histogram estimation (the choice of the bin-width is done using the rule proposed in [42]).

Model and Basis Choice. Albeit $Z \star V$ and V jointly depend on Z, each masking scheme induces a different dependency relationship which implies to adapt the attack strategy accordingly. Namely, for each of the attacks above, the choice of the consumption model (in CPA) or the choice of the basis (in LR attacks) require a careful attention.

To perform the second-order CPA, we chose the centered product combining of the leakages and we defined the optimal model function* m as described in [80] under the assumption $\delta(\cdot) = \mathrm{HW}(\cdot)$. This kind of CPA is denoted $\mathrm{CPA}_{\mathrm{Opt}}$ in the sequel.

As argued in Sect. 7-2.2, to perform efficiently linear regression requires a set of well-chosen basis functions. To approximate the function $\varphi : z \mapsto \mathrm{cov}(\delta(z \star V), \delta(V))$, we have analysed different choices of basis†:

lin where the $(\mathsf{g}_i)_i$ are the degree-1 monomials $z \mapsto z^u$ with $\mathrm{HW}(u) \leqslant 1$.

quad where the $(\mathsf{g}_i)_i$ are the monomials $z \mapsto z^u$ with $\mathrm{HW}(u) \leqslant 2$.

cub where the $(\mathsf{g}_i)_i$ are the monomials $z \mapsto z^u$ with $\mathrm{HW}(u) \leqslant 3$.

full where the $(\mathsf{g}_i)_i$ are the monomials $z \mapsto z^u$ with $\mathrm{HW}(u) \leqslant 8$.

deg2 where the $(\mathsf{g}_i)_i$ are the degree-*2* monomials $z \mapsto z^u$ with $\mathrm{HW}(u) = 2$.

Opt where the basis is reduced to the constant function $z \mapsto 1$ and the function g corresponding to the optimal (prediction) function defined in [80]. In Sect. 8-2.1 (*i.e.* Boolean case), the basis *Opt* is

*Notice that the optimal model function m differs from one masking scheme to another and must therefore be computed for each different masking scheme.

†Every basis contains the constant function, $\mathsf{g}_1 : z \mapsto 1$

denoted by HW to emphasis the affine equivalence between the optimal function and the Hamming weight when the optimal function is designed under the assumption $\delta(\cdot) = \mathrm{HW}$ and $\star = \oplus$.

In the sequel, an attack using the linear regression with basis **basis** will be denoted by LR-**basis**, where **basis** is chosen among *lin*, *quad*, *cub*, *deg*2, *full* and *Opt*.

Remark 27. It has been shown in Sect. 7-2.3 that $\mathrm{CPA}_{\mathrm{Opt}}$ is asymptotically equivalent to LR-Opt, nevertheless we have conducted both attacks to confront this theoretical result to experimentations.

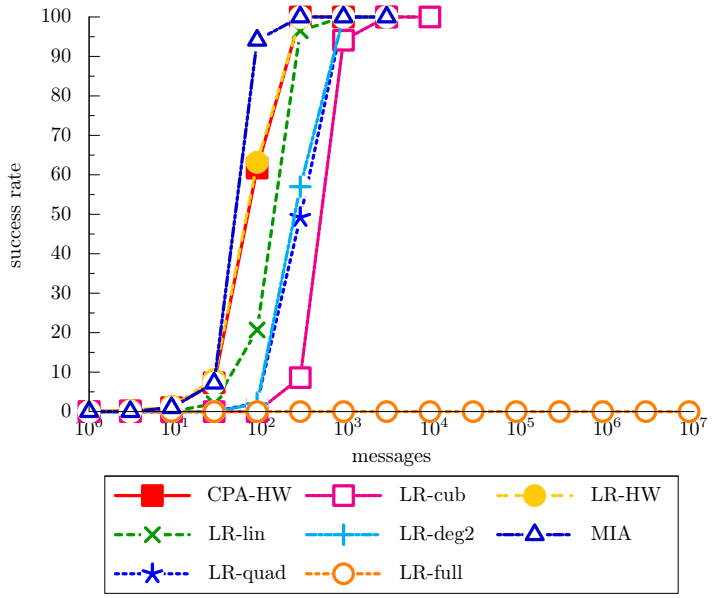
Attack Efficiency. In the following, an attack is said to be *successful* if the good key is output by the attack. An attack is said to be *more efficient than* another if it needs less messages to achieve the same success rate. Success rate is measured over $1,000$ tries.

We report and analyse in next sections our attack simulations results for Scenarios 1, 2 and 3 in case of Boolean (Sect. 8-2.1) and arithmetic masking schemes (Sect. 8-2.2). We inform the reader that we have plotted only attacks which are relevant. In other terms, some attacks never succeeded and thus have not been plotted to ensure readability of figures.

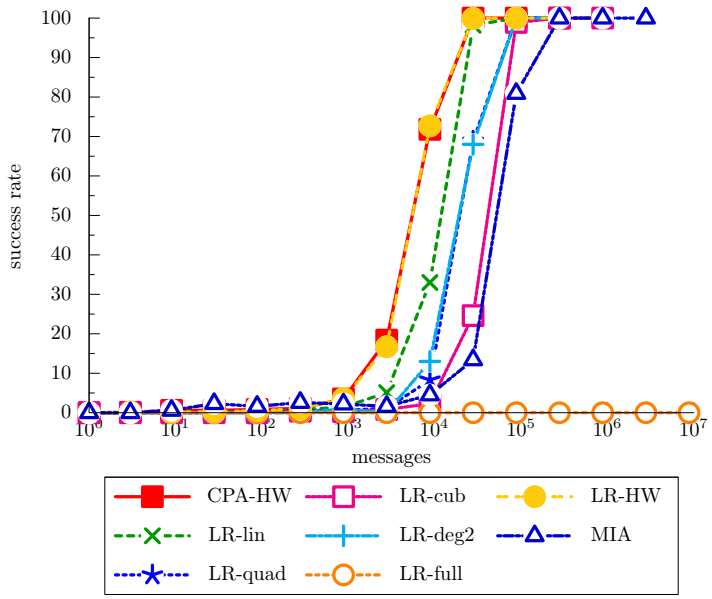
8-2.1 Simulation with Boolean Masking Scheme

In this section we assume that L_1 and L_2 satisfy (8.17) (Scenario 1), or (8.18) (Scenario 2), or (8.19) (Scenario 3). For each attack listed in the previous section, we have plotted in Fig(s). 8-2-1–8-2-3 the success rate as a function of the number of messages. We did this in two different contexts: a non-noisy one (B_1 and B_2 are null) and a noisy one (B_1 and B_2 have mean 0 and standard deviation 4).

In Scenario 1, without noise, MIA is the most efficient attack. When there is noise however, LR-HW performs better than the others. As expected, in both contexts, $\mathrm{CPA}_{\mathrm{HW}}$ and LR-HW share the same rank, while the LR-lin attack is ranked second. This is due to the fact that the hypothesis made over $\delta(\cdot)$ induces a model that exactly corresponds to the



(a) No noise



(b) $\sigma = 4$

Fig. 8-2-1 – Attacks against Boolean masking in Scenario 1

leakage function. Nevertheless, LR-HW and CPA_{HW} stop to be the most efficient attacks in Scenarios 2 and 3. This must be a consequence of the fact that, in those cases the model m is built under the incorrect hypothesis $\delta(\cdot) = \text{HW}(\cdot)$. In Scenario 2, LR-lin is the most efficient attack. The efficiency of the linear regression with basis *lin* is explained by the fact that $y_N(\cdot)$ in (7.3) is linear when $\delta(\cdot)$ does (this is a straightforward extension from the Hamming weight case shown in [80] to any linear function of the bit-coordinates) and it is thus well approximated in the linear basis. In Scenario 3, the results are rather the same than in Scenario 2 since LR-lin is still the most efficient attack. This may appear as a surprising result since we could expect the LR-quad attack to be more efficient. Indeed, in this scenario y_N can be exactly approximated given the basis *quad* but cannot with basis *lin*. So the estimation of y_N returned by the linear regression is better in the quadratic case than in the linear one. Despite this difference, the attack with linear basis discriminates faster. This shows that in some circumstances, it may be sufficient to approximate only the linear part of the leakage and that the computation overhead brought on by a quadratic (or higher) basis, does not worth.

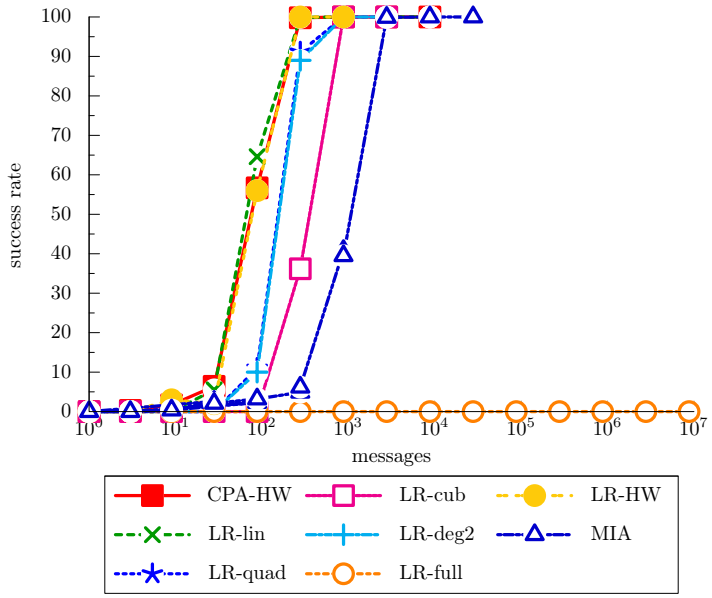
Eventually, it seems that for each attack and in each scenario, the presence of noise makes the curves to be closer from each other. Namely, attacks reaching a 100% success rate seem to become asymptotically equivalent when noise increases. It is explained by the fact that the number of messages needed to annihilate the noise is largely sufficient to have a good approximation with linear regression whatever the size of the basis.

Remark 28. As expected, MIA is always the less efficient attack except in a perfect condition (*i.e.* without noise and with the leakage deterministic part equal to the attack model – here Hamming weight –).

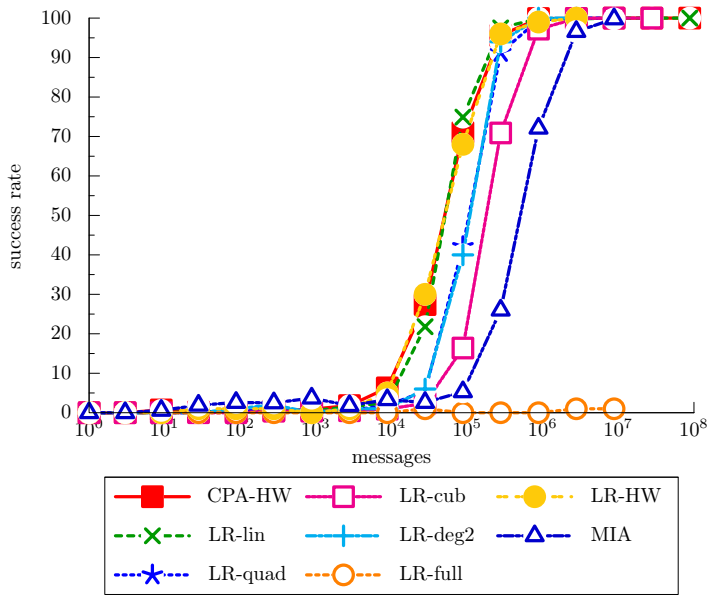
8-2.2 Simulation with Arithmetic Masking Scheme

In this section, L_1 and L_2 satisfy either (8.17) (Scenario 1), or (8.18) (Scenario 2), or (8.19) (Scenario 3). For each attack listed before, we have performed the same attack simulations as in Sect. 8-2.1. The results are plotted in Fig(s). 8-2-4–8-2-6.

In the arithmetic case, all attacks based on the optimal model are the most efficient ones, even in Scenarios 2 and 3. The LR-quad attack is

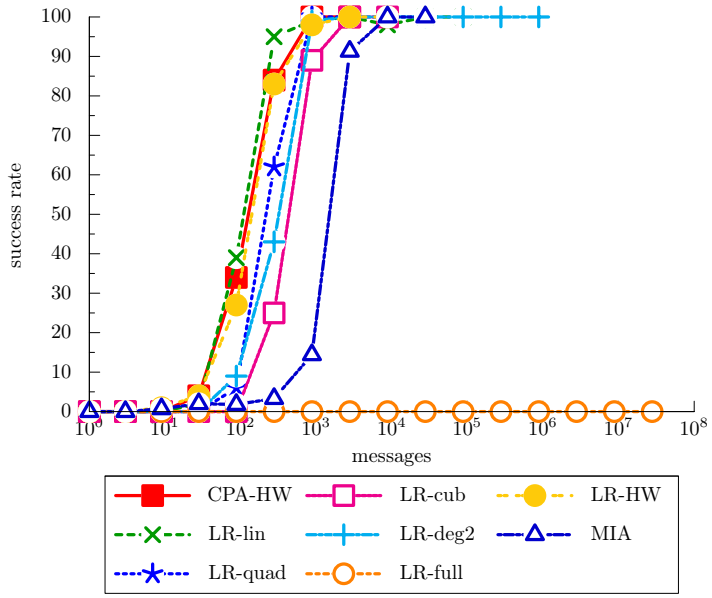


(a) No noise

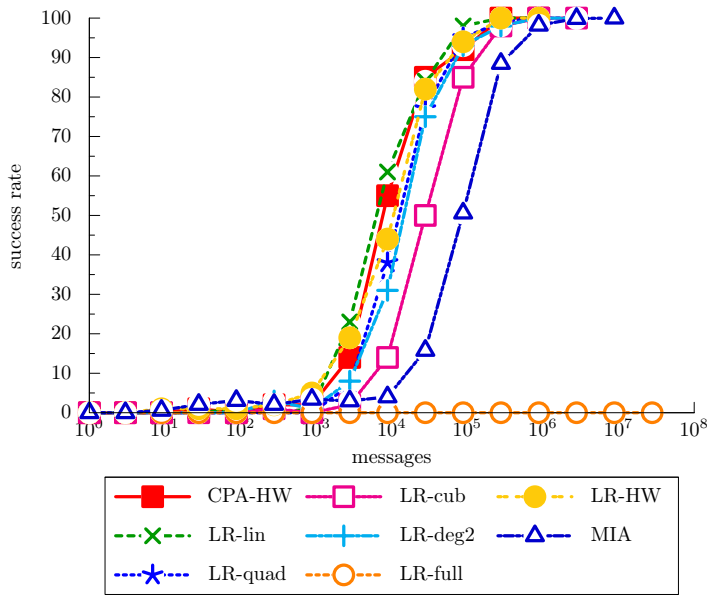


(b) $\sigma = 4$

Fig. 8-2-2 – Attacks against Boolean masking in Scenario 2

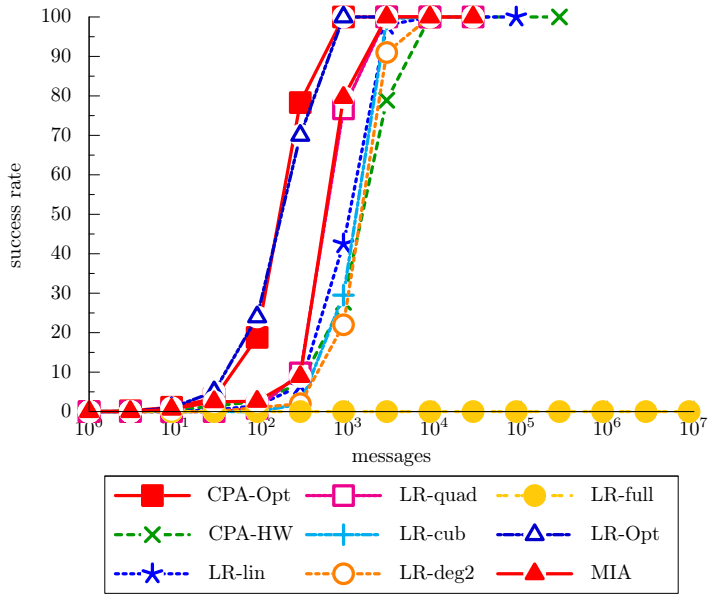


(a) No noise

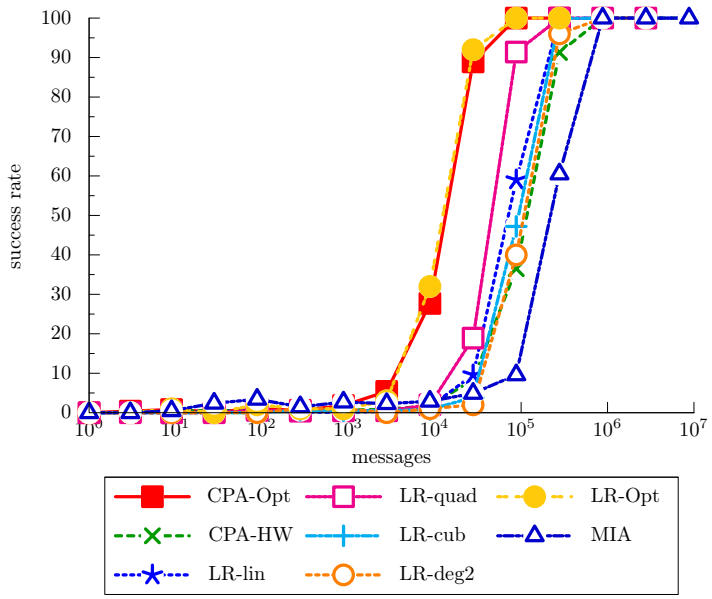


(b) $\sigma = 4$

Fig. 8-2-3 – Attacks against Boolean masking in Scenario 3

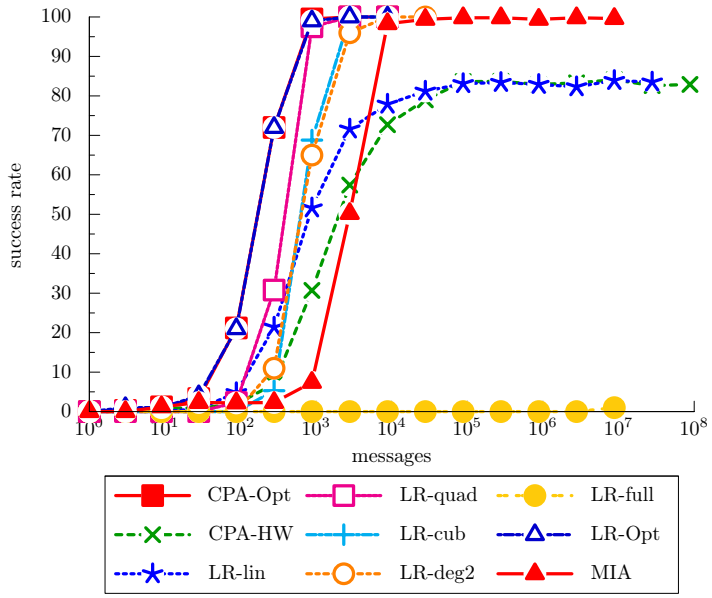


(a) No noise

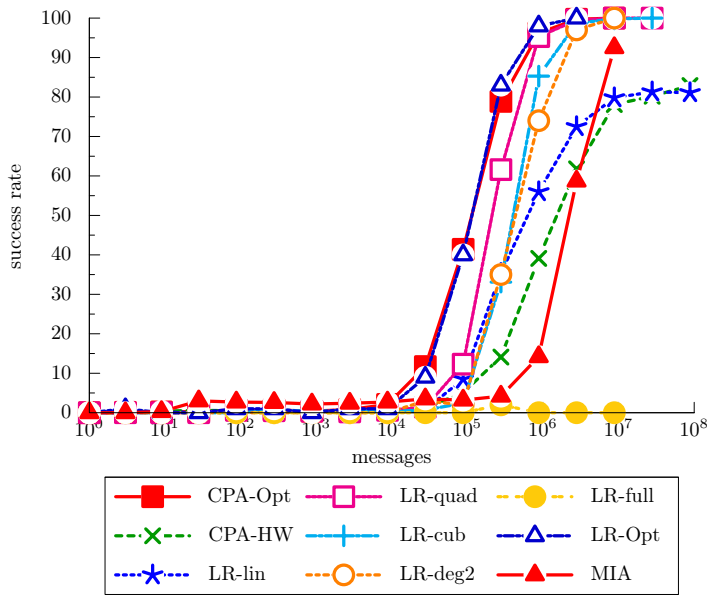


(b) $\sigma = 4$

Fig. 8-2-4 – Attacks against arithmetic masking in Scenario 1

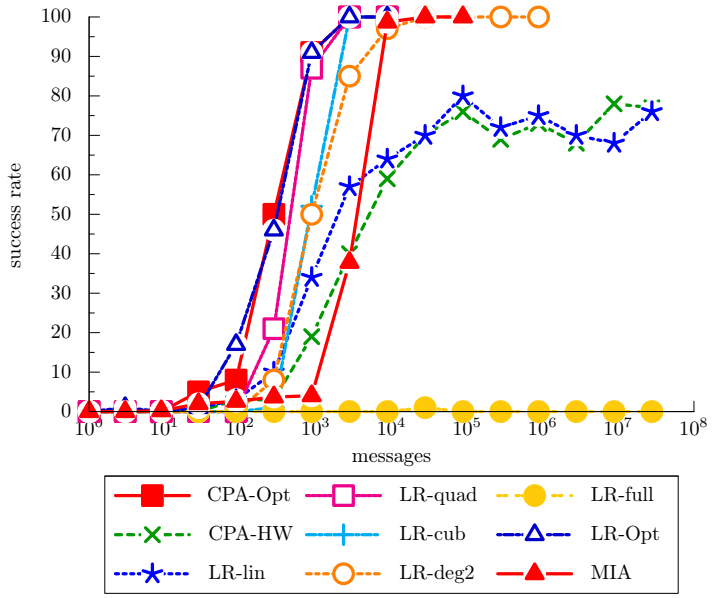


(a) No noise

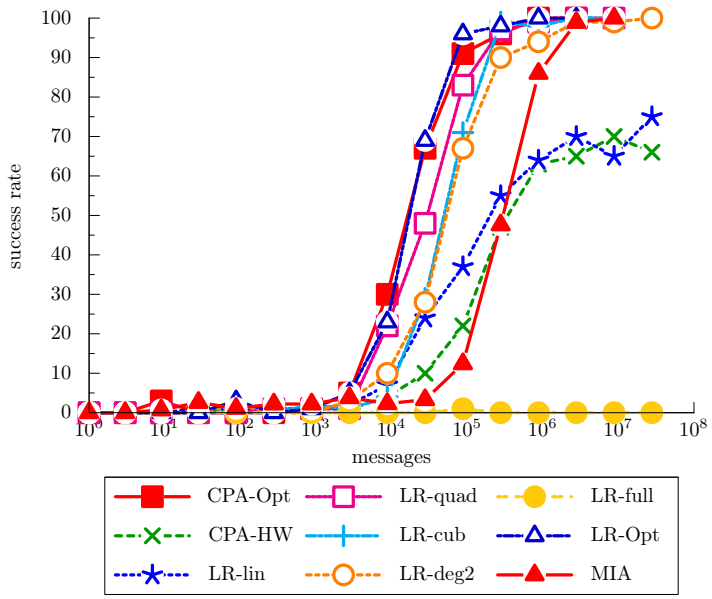


(b) $\sigma = 4$

Fig. 8-2-5 – Attacks against arithmetic masking in Scenario 2



(a) No noise



(b) $\sigma = 4$

Fig. 8-2-6 – Attacks against arithmetic masking in Scenario 3

ranked second for each scenario and its efficiency is close to that of LR-Opt and CPA_{Opt}. In particular, it is always better than CPA_{HW} and LR-lin which actually do not achieve a success rate greater than 85%. This situation can be explained by the fact that the quadratic terms of the function y_N defined in (7.3) have an important influence on the leakage when the masking is arithmetic and not Boolean. To illustrate this, focusing on LR-deg2 attack, it can be checked that its efficiency is close to that of LR-quad (namely the attack performs almost equivalently with and without the linear terms in y_N). The LR-cub attack is ranked third, behind the LR-quad. Therefore, considering the computation overhead induced by the use of a basis with cubic terms, there is no interest to apply the LR-cub attack instead of LR-quad, even if y_N is probably better approximated in *cub* basis than in *quad* basis.

8-2.3 Attacks Experiments in Real Life

In previous sections, we have confronted our theoretical analyses with simulations in realistic scenarios. In the following, we aim at confronting our results against real measurements. Attack parameters like the target, the masking scheme and the distinguisher remain the same as previously defined while the leakage now comes from real power consumption curves.

8-2.3.1 Leakage Measurements

A sample of $400,00$ power consumption leakages have been measured on a 8051 8-bit micro-controller. In each measurement the parts related to the manipulation of $Z \star V$ and V are composed of $100,000$ points. We assumed the curves to be synchronized (a glitch* is used to synchronize at the beginning of the manipulation processing). Since most of the attacks involve different model classes (*e.g.* only HW or linear or quadratic or cubic functions), some of them may be able to succeed when other fail and reciprocally. This observation leads us to not systematically use the same pair of points for all the attacks. Actually, in our attack comparisons, only the pair of points of interest resulting in the maximal distinguishing value has been considered for each attack. Hence,

*By glitch we denoted a brief impulse on the power supply (or clock *etc.*) which have a distinctive pattern on the measurement trace.

before mounting each attack, a pre-processing step has been performed on the curves to determine the two most pertinent *points of interest* (the first point corresponding to $Z \star V$ and the second one corresponding to V). By definition, this pair of points is the one that optimizes the attack efficiency among the $100,000^2$ possible pairs of points. This more or less corresponds to the definition given in [108]. For the CPA, the pair corresponds to the pair of points for which the error resulting from the approximation of the leakage by the attack model is minimal. For the regression-based attacks, the points of interest are those for which the error resulting from the approximation of the leakage in the basis is minimal. During the pre-processing, we have used the fact that we knew the values $Z \star V$ and V manipulated by the device. Even if this does not correspond to a real life adversary, this pre-processing allows us to perform each attack with the optimal choice of points of interest, which is a fair context to compare them together.

Remark 29. The search of the best points of interest is not a prerequisite to the attacks and must therefore not be considered as a profiling step. In fact, in this section we adopt a defensive point of view, meaning that we study the implementation resistance against each attack when it is launched in the most favorable conditions (namely for the best choice of pair). We point out that usually an attacker does not have access to such an information and is consequently less efficient – in algorithmic complexity – even when using the same distinguisher.

8-2.3.2 Experiments Results

For each attack, the distinguishing coefficient has been computed for each key candidate and for a given (increasing) number of power traces up to $460,000$. We recorded the minimal number of messages needed to have the real key ranked first (*i.e.* emerging from others). Results are recorded in Tab. 8-2-1.

Globally, the experiments confirm our simulations results: The attacks are ranked in the same order with the same difference magnitude between them. The number of traces required by the attacks to succeed makes us think that the standard deviation of the noise in leakage is slightly smaller than 4. For Boolean and arithmetic masking, LR attacks and CPA perform quite similarly when they are fed with the same single function (HW or Opt). Interestingly, a basis *lin* is the best choice for the Boolean case, whereas the quadratic terms help to improve at-

Attack \ Masking	Boolean	Arithmetic
CPA_{HW}	933	42,330
CPA_{Opt}		2,039
LR-HW	832	42,320
LR-Opt		2,043
LR-lin	976	6,384
LR-quad	3,907	5,907
LR-cub	15,737	6,620
LR-deg2	4,884	14,705

Tab. 8-2-1 – Experimental results

tack efficiency in the arithmetic case. This is totally in line with the simulations reported in Fig. 8-2-4 and 8-2-5.

8-2.4 Conclusion on the Attack Simulations and Experiments

The theoretical analysis in Sect. 7-2.3 is confirmed by the experimental results. At first, they corroborate the efficiency of the linear regression attacks and show that they are at least as efficient as the CPA and are therefore a real alternative to it. Our simulations point out that LR attacks can even outperform CPA when the device leaks a combination of the manipulated bits that is not well approximated by a simple function (as *e.g.* the Hamming weight). Also, the LR techniques introduced in this paper seem to be particularly suitable against masking schemes with complex algebraic representation over $\mathbb{F}_2$ (like the arithmetic masking). Also, for quite comparable attack success rates (and sometimes even better), the LR techniques are more efficient in terms of computation timings than the classical attacks. This makes them particularly interesting when the leakage noise, and hence the number of required number of traces, is high.

A second outcome of our simulations and experiments is the validation of the importance of the basis choice. Although attacks based on the optimal model in Scenario 1 (for both masking schemes) are always at the first place, this is no longer the case when the optimal model is built from a wrong hypothesis on $\delta(\cdot)$. For instance with Boolean masking,

choosing a linear basis is sufficient to make LR more efficient than LR-Opt, whereas with arithmetic masking a quadratic basis is needed. Finally, as predicted in Sect. 7-2.2, the LR-full attacks (*i.e.* $\mathscr{H} = \mathscr{F}$) always fail.

Remark 30. The presence of noise makes the curves to be closer each one to another. Moreover, whereas the maximal success rate of each attack is unchanged, the higher the noise, the higher the measurements number to achieve the same success rate. In fact the number of messages needed to annihilate the noise is largely sufficient to have a good approximation with linear regression even with a large basis.

8-2.5 A word about Maximum Likelihood Approach

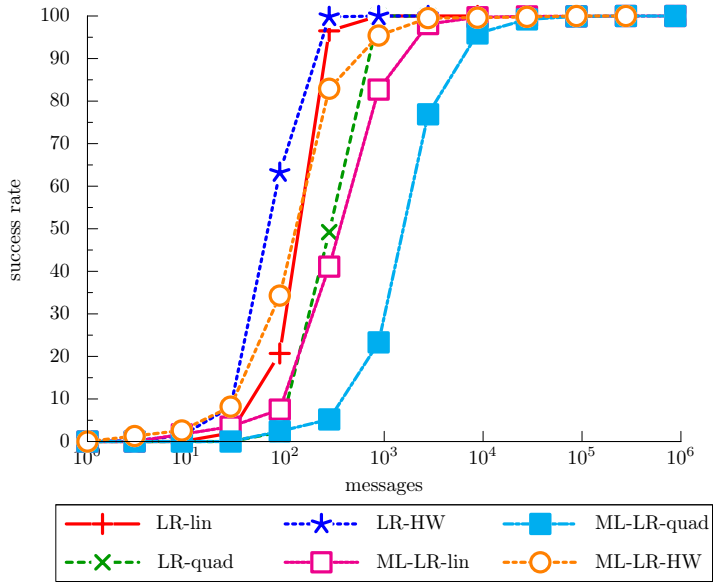
In Sect. 7-2.3.2, we have exhibited the link between our attack and the Maximum Likelihood approach with a *merge* of the mixture components. We propose here to go a step further by using a Maximum Likelihood test as the distinguisher of Step 6 [p. 82] instead of the mean-of-square.

We recall that the Maximum Likelihood test simply consists in computing the product $\prod_i f_{\hat{z}_i}(\ell_1^i, \ell_2^i)$ as already mentioned in Sect. 7-2.3.2. To be able to compute this latter, the adversary must have on hand the pdf f_z for every z. In view of the approximation that is made in (7.15), the only parameter of the pdf that he has to guess is $Y|Z = z$. This latter is already available, as an approximation, at Step 5 of the attack described in Sect. 7-2.1. With this pdf approximation, the adversary replaces the mean-of-square distinguisher used in Step 6 by the Maximum Likelihood test and then outputs the key-candidate which gave rise to the highest discriminating value.

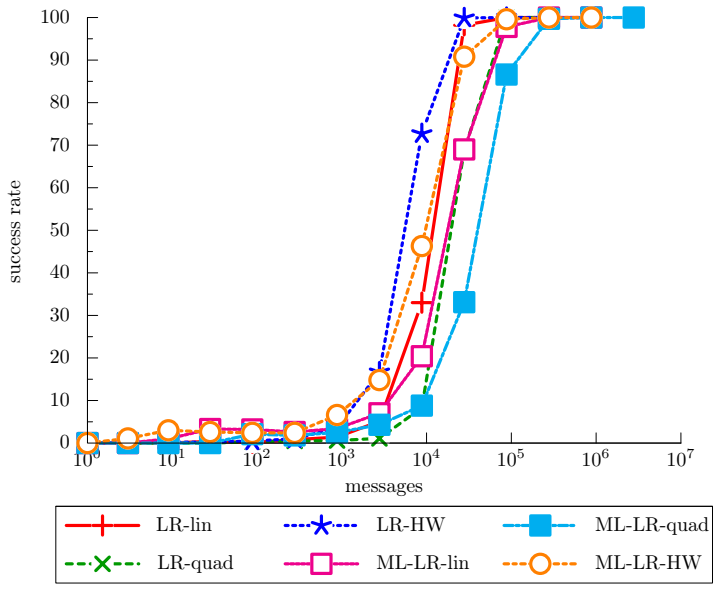
In Sect. 7-2.3.2, we have already shown that this maximum likelihood approach cannot be more efficient than the mean-of-square approach. To confirm and strengthen this fact experimentally, we have conducted some simulations in the Boolean case and scenario 1.

The simulation parameters are the same as in Sect. 8-2 and the results are plotted in Fig. 8-2-7.

As expected, for the same basis, the maximum likelihood approach is never more efficient than the corresponding linear regression approach. More interestingly, the maximum likelihood efficiency is largely lower than the linear regression (by a factor of 3). The reason is that, the



(a) No noise



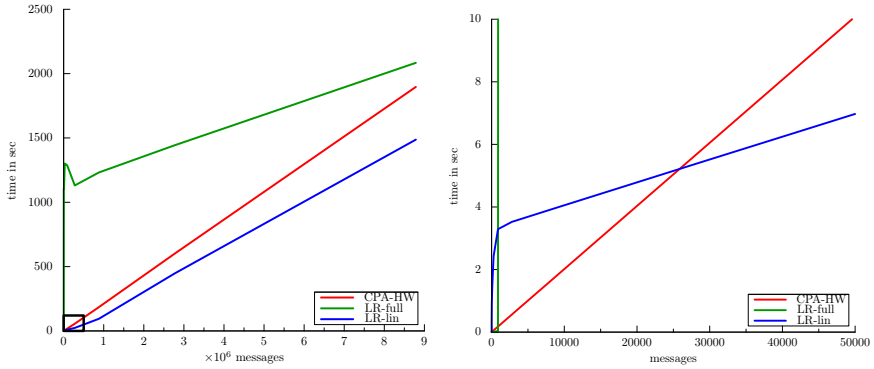
(b) $\sigma = 4$

Fig. 8-2-7 – Comparison between mean-of-square and Maximum Likelihood approach against Boolean masking in Scenario 1

approximation of Y returned by the linear regression is chosen w.r.t. the distance defined in (7.4). In other words, the approximation of Y itself is the result of a discriminating process. Then applying another discriminating test such as the maximum likelihood can only bring more noise.

8-3 Linear Regression Vs CPA: Timings

As demonstrated in [30], the efficiency of an attack decreases exponentially with the masking order. In other term, a successfully attack will need a number of messages N growing exponentially w.r.t. the masking order. This implies that high-order attacks must be able to efficiently deal with a huge number of observations. In particular, the time spent on the processing of the observations may become a bottleneck. Although linear regression processing proposed in Sect. 7-2 is based on matrix operations, the regression matrix has a constant size w.r.t. N (thanks to an initial averaging step – see (7.3)). More precisely, the linear regression complexity can be split into two parts: the matrix operation which i constant w.r.t. N and only depends on the basis size; and the least-square computation (a mean of square) which depends on N. Concerning CPA, its complexity relies on the computation of a mean of product, a product of means and two standard deviations that all depend on N. We can thus expect to have a faster attack when using a linear regression (when N is sufficiently large to neglect the matrix operation). To quantify the timing complexity of linear regression, we did several timing measurements and we compared them with those for CPA attacks. We have first processed linear regression with a linear model as a common use case and with a full basis model as the worst possible case (for $n = 8$), that is with the largest regression matrix (*i.e.* the slowest matrix computation). We remind the reader that in the latter case, the attack always failed (see Sect. 7-2.2). The results are plotted in Fig. 8-3-1a with a zoom on the small numbers of messages in Fig. 8-3-1b. The timings represented in Fig. 8-3-1 are measured over 100 attacks in an univariate setting. Since CPA and linear regression are both univariate and are fed in this paper with the same preprocessed vector of observations (a centered product combination of two leakage vectors), only the core computation differs from one to the other.



(a) **(b)**

Fig. 8-3-1 – Timing comparison for CPA-HW, LR-lin and LR-full attacks.

Results: First and as expected, it can be noticed that the performance of all the attacks are in the same order of magnitude (and thus are computationally viable). Nevertheless, with a linear model, the linear regression becomes noticeably faster than CPA attack (*i.e.* the constant matrix operation cost stays small and can be quickly neglected) for $N > 25,000$ (Fig. 8-3-1b). If we focus on linear regression with the full basis, the cost of the matrix operation is not negligible and thus a large number of message ($N < 10^7$ messages) is needed to counterbalance it. In both cases, when the number of messages is sufficiently large to bypass the timing offset due to the matrix operation, linear regression is faster than CPA as expected.

Conclusion This brief analysis pinpointed the soundness of our attack also in term of computability. That is in all terms the linear regression encompasses and outmatches CPA.

Part III

Countermeasures Analysis

CHAPTER 9

Shuffling and Masking

9-1 Introduction

To thwart DPA attacks, countermeasures try to make leakages as independent as possible of sensitive variables. Nowadays, two main approaches are followed to achieve such a purpose in software: *masking* and *shuffling*. We briefly recall hereafter the two techniques.

The core idea behind masking is to randomly split every sensitive variable Z into $d+1$ shares as explained in Sect. 7-1.1. When carefully implemented (namely when all the shares are processed at different times), d^{th}-order masking perfectly withstands any DPA exploiting less than $d+1$ leakage signals simultaneously. Although attacks exploiting $d+1$ leakages are always theoretically possible, in practical settings their complexity grows exponentially with d [30]. The design of efficient

higher-order masking schemes for block ciphers is therefore of great interest. However, even for small d, dealing with the propagation of the masks through the underlying scheme is an issue. For linear operations, efficient and simple solutions exist that induce an acceptable overhead irrespective of d. Actually, the issue is to protect the non-linear S-boxes computations. In the particular case $d = 1$, a straightforward solution called the *table re-computation* exists (see for instance [20, 68]). Straightforward generalizations of the method to higher-orders d do not provide security *versus* higher-order DPA. Indeed, whatever the number of masks, an attack targeting two different masked input/output is always possible (see for instance [74]). To bypass this flaw, [90] suggests to re-compute a new table before every S-box computation. This solution is very costly in terms of timings and [33] shows the feasibility of third-order attacks, so the scheme is only secure for $d < 3$. An alternative solution for $d = 2$ has been proposed in [84] but the timing overhead is of the same order. Recently, Rivain and Prouff [79] proposed the first high-order masking scheme provably secure for $d > 2$ for the AES cipher and Carlet *et al.* [29] proposed the first generic scheme for high-order masking.

Shuffling consists in spreading the signal containing information about a sensitive variable Z over t different signals $L_1, \ldots, L_t$ leaking at different times. This way, if the spread is uniform, then for every i the probability that L_i corresponds to the manipulation of Z is $\frac{1}{t}$. As a consequence, the signal-to-noise ratio of the instantaneous leakage on Z is reduced by a factor of t (see Sect. 9-2 for details). Applying shuffling is straightforward and does not relate to the nature (linear or non-linear) of the layer to protect. Moreover, shuffling is usually significantly less costly than higher-order masking when applied to non-linear layers.

Since higher-order masking is expensive and since first-order masking can be defeated with quite reasonable efforts [74], a natural idea is to use shuffling together with first-order masking. A few schemes have already been proposed in the literature [46, 103]. In [46], an 8-bit implementation of AES is protected using first-order masking and shuffling. The work in [103] extends this scheme to a 32-bit implementation with the possible use of instructions set extension. Though, [103] proposes some advanced DPA attacks on such schemes whose practicability is demonstrated in [102]. These works show that combining first-order masking with shuffling is definitely not enough to provide a strong security level. A possible improvement is to involve higher-order masking. This raises

two issues. First, a way to combine higher-order masking with shuffling must be defined (especially for S-boxes computations). Secondly, the security of such a scheme should be quantifiable. It would indeed be of particular interest to have a lower bound on the resistance of the overall implementation by choosing *a priori* the appropriate trade-off between masking and shuffling orders. In the rest of the chapter, we address those two issues.

In the next sections, we investigate the security of the combination of masking and shuffling towards DPA. Our analysis is conducted in the Hamming weight leakage model that we formally define hereafter. This model is very common for the analysis of DPA attacks [48, 80, 103] and it has been practically validated several times [69, 74].
Definition 8 (Hamming weight model)**.** Equation 7.1 became

$$L_i = a_i + b_i \cdot \mathrm{H}(Vi) + B_i \ , \tag{9.1}$$

where a_i denotes a constant offset, b_i is a real value, $\mathrm{H}(\cdot)$ denotes the Hamming weight function and B_i denotes a noise with mean 0 and standard deviation σ.

When several leakage signals L_i are jointly considered, we shall make three additional assumptions: (1) the constant b_i is the same for the different L_i (without loss of generality, we consider $b_i = 1$), (2) noises B_i are mutually independent and (3) the noise standard deviation σ is the same for the different B_i.

Higher-order DPA attacks aim at recovering information on a sensitive variable Z by considering several non-simultaneous leakage signals. Let us denote by $\mathbf{L}$ the multivariate random variable corresponding to those signals. The attack starts by converting $\mathbf{L}$ into an univariate random variable by applying it a function g. Then, a *prediction function* f is defined according to some assumptions on the device leakage model. Eventually, every guess $\hat{Z}$ on Z is checked by estimating the correlation coefficient between the combined leakage signal $g(\mathbf{L})$ and the so-called *prediction* $f(\hat{Z})$.

As argued in several works (see for instance [63, 64, 80, 90]), the absolute value of the correlation coefficient $\rho\,[f(Z), g(\mathbf{L})]$ (corresponding to the correct key guess) is a sound estimator of the efficiency of a correlation based DPA characterized by the pair of functions (f, g). Moreover, in [64,95], it is even shown that the number of leakage measurements required for the attack to succeed can be approximated by $c \cdot \rho\,[f(Z), g(\mathbf{L})]^{-2}$

where c is a constant depending on the number of key guesses and the required success rate. In the following, we exhibit in the Hamming weight model (see Def. 8) explicit formulas of this coefficient for advanced DPA attacks where the sensitive variable is either (1) protected by (higher-order) masking, or (2) protected by shuffling or (3) protected with a combination of the two techniques.

9-2 Defeating Shuffling: Integrated DPA

When shuffling is used, the signal containing information about the sensitive variable Z is randomly spread over t different signals L_1, ..., L_t. As a result, the correlation between the prediction and one of these signals is reduced by a factor t compared to the correlation without shuffling. In [32], an *integrated DPA attack* (also called *windowing attack*) is proposed for this issue. The principle is to add the t signals all together to obtain an *integrated signal*. The correlation is then computed between the prediction and the integrated signal. The resulting correlation is reduced by a factor $\sqrt{t}$ instead of t without integration. This is formalized in the next proposition.

Proposition 7. Let $(L_i)_{1\leqslant i\leqslant t}$ be t random variables mutually independent and identically distributed. Let Y denote a signal L_j whose index j is a random variable uniformly distributed over $\{1,\ldots,t\}$. Let X be a random variable that is correlated to Y and that is independent of the remaining L_i. For every measurable function f, the correlation between $f(X)$ and $L_1+\cdots+L_t$ satisfies:

$$\rho[f(X),L_1+\cdots+L_t]=\frac{1}{\sqrt{t}}\rho[f(X),Y]\ . \tag{9.2}$$

Proof. On one hand we have $\mathrm{cov}[f(X),S_1+\cdots+S_t]=\mathrm{cov}[f(X),Y]$ and on the other hand we have $\sigma[S_1+\cdots+S_t]=\sqrt{t}\ \sigma[Y]$. Relation (9.2) straightforwardly follows. ◇

9-3 Defeating Masking: Higher-Order DPA

When d^{th}-order masking is used, any sensitive variable Z is split into $d+1$ shares $Z\oplus\mathbf{V}$, V_1, ..., V_d, where $\mathbf{V}$ denotes the sum $\bigoplus_i V_i$. In the following, we shall denote $Z\oplus\mathbf{V}$ by V_0. The processing of each share V_i

respectively results in a leakage signal L_i. Since the V_i are assumed to be mutually independent, every tuple of d signals or less among the L_i is independent of Z. Thus, to recover information about Z, the joint distribution of all the $d+1$ signals must be considered. Higher-order DPA consists in combining the $d+1$ leakage signals by the mean of a so-called *combining function* $\mathsf{C}(\cdot,\ldots,\cdot)$. This enables the construction of a signal that is correlated to the sensitive variable Z.

In this paper, we therefore consider the normalized product combining (Sect. 7-1.2) generalized to higher orders:

$$\mathsf{C}(L_0,L_1,\ldots,L_d) = \prod_{i=0}^{d} (L_i - \mathbb{E}[L_i]) \ . \tag{9.3}$$

We shall denote by $C_d(Z)$ the – normalized product – combined leakage signal $\mathsf{C}(L_0,L_1,\ldots,L_d)$ where the L_i correspond to the processing of the shares $Z \oplus \mathbf{V}$, V_1, ..., V_d in the Hamming weight model. The following lemma gives the expectation of $C_d(X)$ given $X = x$ for every $x \in \mathbb{F}_2^n$.

Lemma 2. Let $x \in \mathbb{F}_2^n$, then the expectation of $C_d(x)$ satisfies:

$$\mathbb{E}[C_d(x)] = \left(-\frac{1}{2}\right)^d \left(\mathrm{H}(x) - \frac{n}{2}\right) \ . \tag{9.4}$$

In the following, the notation $x[j]$ stands for the j^{th} bit of a value $x \in \mathbb{F}_2^n$. To prove Lemma 2, we first need the following lemma:

Lemma 3. Let $(V_i)_{1 \leqslant i \leqslant d}$ be d random variables uniformly distributed over $\mathbb{F}_2^n$ and mutually independent and let $\mathbf{V} = \bigoplus_i V_i$. For every $x \in \mathbb{F}_2^n$, the expectation of the product $\mathrm{H}(x \oplus \mathbf{V}) \prod_i \mathrm{H}(V_i)$ satisfies:

$$\mathbb{E}\left[\mathrm{H}(x \oplus \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = \left(-\frac{1}{2}\right)^d \left(\mathrm{H}(x) - \frac{n}{2}\right) + \left(\frac{n}{2}\right)^{d+1} \ . \tag{9.5}$$

Proof. We have $\mathrm{H}(x \oplus \mathbf{V}) = \mathrm{H}(x) + \mathrm{H}(\mathbf{V}) - 2\,\mathrm{H}(x \wedge \mathbf{V})$ giving:

$$\begin{aligned}\mathbb{E}\left[\mathrm{H}(x \oplus \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = \mathrm{H}(x)\,\mathbb{E}\left[\prod_{i=1}^{d} \mathrm{H}(V_i)\right] + \mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] \\ - 2\,\mathbb{E}\left[\mathrm{H}(x \wedge \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] \ .\end{aligned} \tag{9.6}$$

Since the V_i are uniformly distributed and mutually independent, we have:

$$\mathbb{E}\left[\prod_{i=1}^{d} \mathrm{H}(V_i)\right] = \left(\frac{n}{2}\right)^d \ , \tag{9.7}$$

and

$$\mathbb{E}\left[\mathrm{H}(x \wedge \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = \frac{\mathrm{H}(x)}{n}\, \mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] \ . \tag{9.8}$$

Relations (9.7) and (9.8) imply that (9.6) can be rewritten as:

$$\mathbb{E}\left[\mathrm{H}(x \oplus \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = \mathrm{H}(x)\left(\frac{n}{2}\right)^d + \left(1 - 2\frac{\mathrm{H}(x)}{n}\right) \mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] \ . \tag{9.9}$$

The uniformity and mutual independence of the V_i further imply:

$$\mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = n\, \mathbb{E}\left[\mathbf{V}[1] \prod_{i=1}^{d} \mathrm{H}(V_i)\right] \ ,$$

which can be rewritten as:

$$\begin{aligned}\mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = n\, \mathbb{E}\left[\mathbf{V}[1] V_1[1] \prod_{i=2}^{d} \mathrm{H}(V_i)\right] \\ + n\, \mathbb{E}\left[\mathbf{V}[1] \left(\sum_{j=2}^{n} V_1[j]\right) \prod_{i=2}^{d} \mathrm{H}(V_i)\right] \ ,\end{aligned}$$

and by induction on d:

$$\begin{aligned}\mathbb{E}\left[\mathrm{H}(\mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] = n\, \mathbb{E}\left[\mathbf{V}[1] \prod_{i=1}^{d} V_i[1]\right] \\ + n \sum_{k=1}^{d} \mathbb{E}\left[\left(\mathbf{V}[1] \prod_{i=1}^{k-1} V_i[1]\right)\left(\sum_{j=2}^{n} V_k[j]\right)\left(\prod_{i=k+1}^{d} \mathrm{H}(V_i)\right)\right] \ .\end{aligned} \tag{9.10}$$

Then, on one hand, we have:

$$\mathbb{E}\left[\mathbf{V}[1] \prod_{i=1}^{d} V_i[1]\right] = 2^{-d}(d \bmod 2) \ , \tag{9.11}$$

and, on the other hand, we have the mutual independence between $\mathbf{V}[1]$,

$(V_i[1])_{1\leqslant i\leqslant k-1}$, $\sum_{j=2}^{n} V_k[j]$ and $(\mathrm{H}(V_i))_{k+1\leqslant i\leqslant d}$ which implies:

$$\begin{aligned}\mathbb{E}\left[\left(\mathbf{V}[1]\prod_{i=1}^{k-1}V_i[1]\right)\left(\sum_{j=2}^{n}V_k[j]\right)\left(\prod_{i=k+1}^{d}\mathrm{H}(V_i)\right)\right]&\\ =\left(\mathbb{E}[\mathbf{V}[1]]\prod_{i=1}^{k-1}\mathbb{E}[V_i[1]]\right)\mathbb{E}\left[\sum_{j=2}^{n}V_k[j]\right]\left(\prod_{i=k+1}^{d}\mathbb{E}[\mathrm{H}(V_i)]\right)&\\ =\left(\frac{1}{2}\right)^k\frac{n-1}{2}\left(\frac{n}{2}\right)^{d-k}\quad.&\quad(9.12)\end{aligned}$$

From (9.11) and (9.12), (9.10) can be rewritten as:

$$\begin{aligned}\mathbb{E}\left[\mathrm{H}(\mathbf{V})\prod_{i=1}^{d}\mathrm{H}(V_i)\right] &= \frac{n(d \bmod 2)}{2^d}+\frac{n(n-1)}{2}\sum_{k=1}^{d}\left(\frac{1}{2}\right)^k\left(\frac{n}{2}\right)^{d-k} &(9.13)\\ &= \frac{n(d \bmod 2)}{2^d}+\frac{n(n-1)}{2^{d+1}}\sum_{k=1}^{d}n^{k-1} &(9.14)\\ &= \frac{n(d \bmod 2)}{2^d}+\frac{n(n^d-1)}{2^{d+1}} &(9.15)\\ &= \frac{n((-1)^d+1)+n(n^d-1)}{2^{d+1}}\quad. &(9.16)\end{aligned}$$

Finally, (9.9) and (9.16) yields (9.5) which conclude the proof. ◇

Proof. (Lemma 2) From the expression of the L_i, we have $\mathbb{E}[L_i]=a_i+\frac{n}{2}$ giving:

$$C_d(X)=\left(\mathrm{H}(x\oplus\mathbf{V})-\frac{n}{2}+B_0\right)\prod_{i=1}^{d}\left(\mathrm{H}(V_i)-\frac{n}{2}+B_i\right)\quad.\qquad(9.17)$$

Since the B_i have zero means, one deduces:

$$\begin{aligned}\mathbb{E}[C_d(x)] &= \mathbb{E}\left[\left(\mathrm{H}(x\oplus\mathbf{V})-\frac{n}{2}\right)\prod_{i=1}^{d}\left(\mathrm{H}(V_i)-\frac{n}{2}\right)\right]\\ &= \mathbb{E}\left[\mathrm{H}(x\oplus\mathbf{V})\prod_{i=1}^{d}\left(\mathrm{H}(V_i)-\frac{n}{2}\right)\right]-\frac{n}{2}\,\mathbb{E}\left[\prod_{i=1}^{d}\left(\mathrm{H}(V_i)-\frac{n}{2}\right)\right]\quad.\end{aligned}$$

The uniformity and the mutual independence between the V_i imply:

$$\begin{aligned}\mathbb{E}[C_d(x)] &= \mathbb{E}\left[\mathrm{H}(x\oplus\mathbf{V})\prod_{i=1}^{d}\left(\mathrm{H}(V_i)-\frac{n}{2}\right)\right]\\ &= \mathbb{E}\left[\mathrm{H}(x\oplus\mathbf{V})\mathrm{H}(V_1)\prod_{i>1}\left(\mathrm{H}(V_i)-\frac{n}{2}\right)\right]\quad,\end{aligned}$$

and by induction on d:

$$\mathbb{E}[C_d(x)] = \mathbb{E}\left[\mathrm{H}(x \oplus \mathbf{V})\mathrm{H}(V_1)\cdots\mathrm{H}(V_{d-1})\left(\mathrm{H}(V_d) - \frac{n}{2}\right)\right] . \tag{9.18}$$

Finally, the uniformity and the mutual independence between the V_i lead to:

$$\mathbb{E}[C_d(x)] = \mathbb{E}\left[\mathrm{H}(x \oplus \mathbf{V}) \prod_{i=1}^{d} \mathrm{H}(V_i)\right] - \left(\frac{n}{2}\right)^{d+1} , \tag{9.19}$$

which together with Lemma 3 imply (9.4). ◇

Lemma 2 shows that the expectation of $C_d(x)$ is an affine function of the Hamming weight of x. According to the analysis in [80], this implies that the Hamming weight of X maximizes the correlation. For the reasons given in [80], this function can therefore be considered as an optimal prediction for $C_d(X)$. Hence, the HO-DPA we focus on here, consists in estimating the correlation between the Hamming weight of the target variable $\mathrm{H}(Z)$ and the combined leakage $C_d(Z)$. The next proposition provides the exact value of this correlation.

Proposition 8. Let X be a random variable uniformly distributed over $\mathbb{F}_2^n$. The correlation between $\mathrm{H}(X)$ and $C_d(X)$ satisfies:

$$\rho[\mathrm{H}(X), C_d(X)] = (-1)^d \frac{\sqrt{n}}{\left(n + 4\sigma^2\right)^{\frac{d+1}{2}}} . \tag{9.20}$$

Proof. For any measurable function f and for any pair of random variables (X, C), the expectation $\mathbb{E}[f(X)C]$ is equal to $\mathbb{E}[f(X)\mathbb{E}[C|X]]$. This implies that the covariance between $\mathrm{H}(X)$ and $C_d(X)$ satisfies:

$$\mathrm{cov}[\mathrm{H}(X), C_d(X)] = \mathrm{cov}[\mathrm{H}(X), \mathbb{E}[C_d(X)|X]] .$$

By Lemma 2, we get:

$$\mathrm{cov}[\mathrm{H}(X), C_d(X)] = \left(-\frac{1}{2}\right)^d \mathrm{var}[\mathrm{H}(X)] ,$$

which leads to:

$$\rho[\mathrm{H}(X), C_d(X)] = \left(-\frac{1}{2}\right)^d \frac{\sigma[\mathrm{H}(X)]}{\sigma[C_d(X)]} = \left(-\frac{1}{2}\right)^d \frac{\sqrt{n}}{2\,\sigma[C_d(X)]} . \tag{9.21}$$

Since X and the V_i are uniformly distributed and mutually independent, then so do $X \oplus \mathbf{V}$ and the V_i. Moreover since the B_i are mutually independent then we get:

$$\begin{aligned}\mathsf{var}[C_d(X)] &= \mathbb{E}\left[\left(\mathrm{H}(X \oplus \mathbf{V}) - \frac{n}{2} + B_0\right)^2\right] \prod_{i=1}^{d} \mathbb{E}\left[\left(\mathrm{H}(V_i) - \frac{n}{2} + B_i\right)^2\right] \\ &= \mathbb{E}\left[\left(\mathrm{H}(V) - \frac{n}{2} + B\right)^2\right]^{d+1} ,\end{aligned}$$

where V is a uniform random variable over $\mathbb{F}_2^n$ and B is a random variable with mean 0 and variance σ^2. Since $\mathbb{E}[\mathrm{H}(V)] = n/2$ and $\mathbb{E}\left[\mathrm{H}(V)^2\right] = (n^2 + n)/4$, one deduces:

$$\mathbb{E}\left[\left(\mathrm{H}(V) - \frac{n}{2} + B\right)^2\right] = \mathbb{E}\left[\left(\mathrm{H}(V) - \frac{n}{2}\right)^2\right] + \mathbb{E}\left[B^2\right] = \frac{n}{4} + \sigma^2 ,$$

which implies:

$$\mathsf{var}[C_d(X)] = \left(\frac{n}{4} + \sigma^2\right)^{d+1} . \tag{9.22}$$

Finally, (9.21) and (9.22) leads to (9.20). ◇

Notation. The correlation coefficient defined in (9.20) shall be referred as $\rho(n, d, \sigma)$.

9-4 Defeating Combined Masking and Shuffling: Combined Higher-Order and Integrated DPA

When masking is combined with shuffling, any sensitive variable Z is split into $d+1$ shares $Z \oplus \mathbf{V}$, V_1, ..., V_d whose manipulations are randomly spread over t different times yielding t different signals L_i. The $(d+1)$-tuple of signals indices corresponding to the shares hence ranges over a subset I of the set of $(d+1)$-combinations from $\{1, \ldots, t\}$. This subset depends on how the shuffling is performed (*e.g.* the shares may be independently shuffled or shuffled all together).

To bypass such a countermeasure, an adversary may combine integrated and higher-order DPA techniques. A pertinent way to perform such a combined attack is to design a so-called *combined-and-integrated* signal

by summing all the possible combinations of $d+1$ signals among L_1, ..., L_t [102, 103]. That is, the combined-and-integrated signal, denoted $IC_{d,I}(Z)$, is defined by:

$$IC_{d,I}(Z) = \sum_{\mathbf{i}\in I} \mathsf{C}(L_{i_0},\ldots,L_{i_d}) \ , \tag{9.23}$$

where $\mathbf{i}$ denotes the vector $(i_0,...,i_d)$.

By construction of I, the family of signals $(L_i)_i$ corresponds to a family of processed data $(D_i)_i$ such that there always exists a single $(d+1)$-tuple $(i'_0,\ldots,i'_d) \in I$ for which we have $(D_{i'_0},D_{i'_1},\ldots,D_{i'_d}) = (Z \oplus \bigoplus_i V_i, V1,\ldots,V_d)$. Let us now view $(i'_0,\ldots,i'_d)$ as a random vector uniformly distributed over I and let us assume that the random variables D_j with $j \neq i'_0,...,i'_d$ are uniformly distributed and mutually independent. Then, we have the following proposition:

Proposition 9. Let X be a random variable uniformly distributed over $\mathbb{F}_2^n$. The correlation between $\mathrm{H}(X)$ and $IC_{d,I}(X)$ satisfies:

$$\rho\left[\mathrm{H}(X), IC_{d,I}(X)\right] = \frac{1}{\sqrt{\#I}}\rho(n,d,\sigma) \ . \tag{9.24}$$

Proof. According to (9.23) the variance of $IC_{d,I}(X)$ satisfies:

$$\mathsf{var}\left[IC_{d,I}(X)\right] = \sum_{(\mathbf{i},\mathbf{j})\in I^2} \mathrm{cov}\left[\mathsf{C}(L_{i_0},\ldots,L_{i_d}), \mathsf{C}(L_{j_0},\ldots,L_{j_d})\right] \ .$$

Since by definition each monomial $\mathsf{C}(L_{j_0},\ldots,L_{j_d})$ is a product of terms with zero expectation, the covariance between two different monomials equal zero. By construction, the $\#I$ monomials $\mathsf{C}(L_{i_0},\ldots,L_{i_d})$ have equal variance and we therefore have $\sigma\left[IC_{d,I}(X)\right] = \sqrt{\#I} \times \sigma\left[\mathsf{C}(L_{i'_0},\ldots,L_{i'_d})\right]$. Moreover, we have only the combination $\mathsf{C}(L_{i'_0},\ldots,L_{i'_d})$ which is statistically dependent on X. Therefore, we deduce that $\mathrm{cov}\left[\mathrm{H}(X), IC_{d,I}(X)\right]$ is equal to $\mathrm{cov}\left[\mathrm{H}(X), \mathsf{C}(L_{i'_0},\ldots,L_{i'_d})\right]$. Since by definition $\mathsf{C}(L_{i'_0},\ldots,L_{i'_d})$ and $C_d(X)$ have an equal distributions, we deduce that the correlation $\rho\left[\mathrm{H}(X), \mathsf{C}(L_{i'_0},\ldots,L_{i'_d})\right]$ equals $\rho[\mathrm{H}(X), C_d(X)] = \rho(n,d,\sigma)$ and Relation (9.24) straightforwardly follows. ◇

CHAPTER 10

A Generic Scheme Combining Higher-Order Masking and Shuffling

10-1 The scheme

In this section, we describe a generic scheme to protect a round φ by combining higher-order masking and operations shuffling. Our scheme involves a d^{th}-order masking for an arbitrarily chosen d. Namely, the state p is split into $d+1$ shares m_0, ..., m_d satisfying:

$$m_0 \oplus \cdots \oplus m_d = p \ . \tag{10.1}$$

In practice, m_1, ..., m_d are random masks and m_0 is the masked state defined according to (10.1). In the sequel, we shall denote by $(m_j)_i$ (resp. $(m_j)_{i(l)}$) the i^{th} n-bit part (resp. the i^{th} l-bit part) of a share m_j. At the beginning of the ciphering the masks are initialized to zero. Then, each time a part of a mask is used during the keyed substitution layer

computation, it is refreshed with a new random value (see below). For the reasons given in Sect. 9-1, our scheme uses two different approaches to protect the keyed substitution layer and the linear layer. These are described hereafter.

10-1.1 Protecting the keyed substitution layer

To protect the keyed substitution layer, we use (for some $d' \leqslant d$) a single d'^{th}-order masked S-box to perform all the S-box computations. As explained in Sect. 9-1, such a method is vulnerable to a second-order DPA attack targeting two masked inputs/outputs. To deal with this issue, we make use of a high level of shuffling in order to render such an attack difficult and to keep an homogeneous security level (see Sect. 10-3).

The input of S is masked with d' masks r_1, ..., $r_{d'}$ and its output is masked with d' masks s_1, ..., $s_{d'}$. Namely, a masked S-box S^* is computed that is defined for every $x \in \{0,1\}^n$ by:

$$S^*(x) = S\left(x \oplus \bigoplus_{j=1}^{d'} r_j\right) \oplus \bigoplus_{j=1}^{d'} s_j \ . \tag{10.2}$$

This masked S-box is then involved to perform all the S-box computations. Namely, when the S-box must be applied to a masked variable $(m_0)_i$, the d masks $(m_j)_i$ of this latter are replaced by the d' masks r_j which enables the application of S^*. The d' masks s_j of the obtained masked output are then switched for d new random masks $(m_j)_i$.

The high level shuffling is ensured by the addition of dummy operations. Namely, the S-box computation is performed t times: N times on a relevant part of the state and $t-N$ times on dummy data. For such a purpose, each share m_j is extended by a dummy part $(m_j)_{N+1}$ that is initialized by a random value at the beginning of the ciphering. The round key k is also extended by such a dummy part k_{N+1}. For each of the t S-box computations, the index i of the parts $(m_j)_i$ to process is read in a table T. This table of size t contains all the indices from 1 to N stored at random positions and its $t-N$ other elements equal $N+1$. Thanks to this table, the S-box computation is performed once on every of the N relevant parts and $t-N$ times on the dummy parts. The following algorithm describes the whole protected keyed substitution layer computation.

Algorithm 1 Protected keyed substitution layer

INPUT: the shares m_0, ..., m_d such that $\bigoplus m_i = p$ and the round key $k = (k_1, \ldots, k_{N+1})$

OUTPUT: the shares m_0, ..., m_d such that $\bigoplus m_i = \gamma(p \oplus k)$

1. **for** $i_T = 1$ **to** t

// Random index pick-up

2. $i \leftarrow T[i_T]$

// Masks conversion : $(m_0)_i \Leftarrow p_i \bigoplus_j r_j$

3. **for** $j = 1$ **to** d' **do** $(m_0)_i \leftarrow ((m_0)_i \oplus r_j) \oplus (m_j)_i$

4. **for** $j = d' + 1$ **to** d **do** $(m_0)_i \leftarrow (m_0)_i \oplus (m_j)_i$

// key addition and S-box computation: $(m_0)_i \Leftarrow S(p_i \oplus k_i) \oplus \bigoplus_j s_j$

5. $(m_0)_i \leftarrow S^*\left((m_0)_i \oplus k_i\right)$

// Masks generation and conversion: $(m_0)_i \Leftarrow S(p_i \oplus k_i) \oplus \bigoplus_j (m_j)_i$

6. **for** $j = 1$ **to** d'

7. $(m_j)_i \leftarrow \mathsf{rand}()$

8. $(m_0)_i \leftarrow ((m_0)_i \oplus (m_j)_i) \oplus s_j$

9. **for** $j = d' + 1$ **to** d

10. $(m_j)_i \leftarrow \mathsf{rand}()$

11. $(m_0)_i \leftarrow (m_0)_i \oplus (m_j)_i$

12. **return** $(m_0, \ldots, m_d)$

Remark 31. In Steps 3 and 8, we used round brackets to underline the order in which the masks are introduced. A new mask is always introduced before removing an old mask. Respecting this order is mandatory for the scheme security.

Masked S-box computation. The look-up table for S^* is computed dynamically at the beginning of the ciphering by performing d' table re-computations such as proposed in [90]. This method has been shown to be insecure for $d' > 2$, or for $d' > 3$ depending on the table re-computation algorithm [33, App. A]. We will therefore consider that one can compute a masked S-box S^* with $d' \leqslant 3$ only. The secure computation of a masked S-box with $d' > 3$ has been resolved recently by Rivain *et al.* in [85] for AES only and is left to further investigations for a generic scheme.

Indices table computation. Several solutions exist in the literature to randomly generate indices permutation over a finite set [49, 75, 77]. Most of them can be slightly transformed to design tables T of size $t \geqslant N$ containing all the indices 1 to N in a random order and whose remaining cells are filled with $N+1$. However, few of those solutions are efficient when implemented in low resources devices. In our case, since t is likely to be much greater than N, we have a straightforward algorithm which tends to be very efficient for $t \gg N$. To generate T, we start by initializing all the cells of T to the value $N+1$. Then, for every $j \leqslant N$, we randomly generate an index $i < t$ until $T[i] = N+1$ and we move j into $T[i]$. The process is detailed hereafter.

Algorithm 2 Generation of T

INPUT: state's length N and shuffling order t

OUTPUT: indices permutation table T

1. **for** $i \leftarrow 0$ **to** $t-1$
2. **do** $T[i] \leftarrow N+1$ // Initialization of T
3. $j \leftarrow 1$
4. **for** $j \leftarrow 1$ **to** N

5. **do** $i \leftarrow rand(t)$ **while** $T[i] = N+1$ // Generate random index $i < t$
6. $T[i] = j$ and $j \leftarrow j+1$
7. **return** T

10-1.2 Protecting the linear layer

The atomic operations λ_i are applied on each part $(m_j)_{i(l)}$ of each share m_j in a random order. For such a purpose a table T' is constructed at the beginning of the ciphering that is randomly filled with all the pairs of indices $(j,i) \in \{0,\ldots,d\} \times \{1,\ldots,L\}$. The linear layer is then implemented such as described by the following algorithm.

Algorithm 3 Protected linear layer

INPUT: the shares m_0, ..., m_d such that $\bigoplus m_i = p$

OUTPUT: the shares m_0, ..., m_d such that $\bigoplus m_i = \lambda(p)$

1. **for** $i_{T'} = 1$ **to** $(d+1) \cdot L$
2. $(j,i) \leftarrow T'[i_{T'}]$ // Random index look-up
3. $(m_j)_{i(l)} \leftarrow \lambda_i\big((m_j)_{i(l)}\big)$ // Linear operation
4. **return** $(m_0,\ldots,m_d)$

Indices table computation. To implement the random generation of a permutation T' on $\{0,\ldots,d\} \times \{1,\ldots,L\}$, we followed the outlines of the method proposed in [34]. However, since this method can only be applied to generate permutations on sets with cardinality a power of 2 (which is not *a priori* the case for T'), we slightly modified it. The new version can be found below.

Generation of T'. In view of the previous complexity, generating a permutation with the same implementation as for T is not pertinent (in this case $t = N$). To generate the permutation T', we follow the outlines of the method proposed in [34]. However, since this method can only be applied to generate permutations on sets with cardinality a power of 2 (which is not a priori the case for T'), we slightly modified it. Let 2^q be the smallest power of 2 which is greater than $(d+1)L$. Our algorithm essentially consists in designing a q-bit random permutation T' from a fixed q-bit permutation π and a family of q random values in $\mathbb{F}_2^q$ (Steps 1 to 6 in Algorithm 4). Then, if $(d+1)L$ is not a power of 2, the table T' is transformed into a permutation over $\{0,\ldots,d\} \times \{1,\ldots,L\}$ by deleting the elements which are strictly greater than $(d+1)L-1$.

Algorithm 4 Generation of T'

INPUT: parameters (d,L) and a n'-bit permutation π with $q = \lceil \log_2((d+1)L) \rceil$

OUTPUT: indices permutation table T'

1. **for** $i \leftarrow 0$ **to** $q-1$
2. **do** $alea_i \leftarrow rand(q)$ // Initialization of aleas
3. **for** $j \leftarrow 0$ **to** $2^q - 1$
4. **do** $T'[j] \leftarrow \pi[j]$
5. **for** $i \leftarrow 0$ **to** $q-1$
6. **do** $T'[j] \leftarrow \pi[T'[j] \oplus alea_i]$ // Process the $i$þindex
7. **if** $q \neq (d+1)L$
8. **then for** $j \leftarrow 0$ **to** $(d+1)L-1$
9. **do** $i \leftarrow j$
10. **while** $T'[i] \geq (d+1)L$
11. **do** $i \leftarrow i+1$
12. $T'[j] \leftarrow T'[i]$
13. **return** T'

With Algorithm 4, it is not possible to generate all the permutations over $\{0,\ldots,d\} \times \{1,\ldots,L\}$. In our context, we assume that this does not introduce any weakness in the scheme.

10-2 Time Complexity

In the following we express the time complexity of each step of our scheme in terms of the parameters (t,d,d',N,L) and of constants a_i that depend on the implementation and the device architecture. Moreover, we provide practical values of these constants (in number of clock cycles) for an AES implementation protected with our scheme and running on a 8051-architecture.

Generation of T.

Complexity Analysis of loop 4-to-6: $f(N,t)$, the expected number of iterations of the loop 4-to-7 in Algorithm 2 satisfies:

$$f(N,t) = t \cdot (H_t - H_{t-N}) \quad , \tag{10.3}$$

where for every r, H_r denotes the r^{th} *Harmonic number* defined by $H_r = \sum_{i=1}^{r} \frac{1}{i}$.

Let us argue about (10.3). For every $j \leqslant N$, the probability that the loop **do-while** ends up after i iterations is $\left(\frac{t-j}{t}\right) \cdot \left(\frac{j}{t}\right)^{i-1}$: at the j^{th} iteration of the **for** loop, the test $T[i] = N+1$ succeeds with probability $p_j = \left(\frac{j}{t}\right)$ and fails with probability $1 - p_j = \left(\frac{t-j}{t}\right)$. One deduces that for every $j \leqslant N$, the expected number of iterations of the loop **do-while** is $\sum_{i\in\mathbb{N}} i \cdot p_j^{i-1} \cdot (1 - p_j)$. We eventually get that the number of iterations $f(N,t)$ satisfies $f(N,t) = \sum_{j=0}^{N-1} \sum_{i\in\mathbb{N}} i \cdot \left({p_j}^{i-1} - {p_j}^{i}\right)$, that is $f(N,t) = \sum_{j=0}^{N-1} \sum_{i\in\mathbb{N}} i \cdot {p_j}^{i-1} - \sum_{j=0}^{N-1} \sum_{i\in\mathbb{N}} (i+1) \cdot {p_j}^{i} + \sum_{j=0}^{N-1} \sum_{i\in\mathbb{N}} {p_j}^{i}$. As the two first sums in the right-hand side of the previous equation are equal, one deduces that $f(N,t)$ equals $\sum_{j=0}^{N-1} \sum_{i\in\mathbb{N}} {p_j}^{i}$

that is $\sum_{j=0}^{N-1} \frac{1}{1-p_j}$. Eventually, as p_j equals $\frac{j}{t}$, we get $f(N,t) = \sum_{j=0}^{N-1} \frac{t}{t-j}$ which is equivalent with (10.3).

As H_r tends towards $\ln(r)+\gamma$, where γ stands for the Euler-Mascheroni constant, we can approximate $H_t - H_{t-N}$ by $\ln(t) - \ln(t-N)$. We eventually get the following relation for $t \gg N$:

$$f(N,t) \approx t \cdot \ln\left(\frac{t}{t-N}\right).$$

Overall Complexity: The complexity $\mathscr{C}_T$ of the generation of T satisfies:

$$\mathscr{C}_T = t \times a_0 + N \times a_1 + f(N,t) \times a_2 \ ,$$

where $f(N,t) = t\sum_{i=0}^{N-1} \frac{1}{t-i}$. As argued above, $f(N,t)$ can be approximated by $t\ln\left(\frac{t}{t-N}\right)$ for $t \gg N$.

Example 3. For our AES implementation, we got $a_0 = 6$, $a_1 = 7$ and $a_2 = 9$.

Generation of T'.

Complexity Analysis of loop 8-to-12: The number of iterations of loop 8-to-12 in Algorithm 4 in the worst case is 2^q.

Overall Complexity: Let q denote $\lceil \log_2((d+1)L) \rceil$. The complexity $\mathscr{C}_{T'}$ satisfies:

$$\mathscr{C}_{T'} = \begin{cases} q \times a_0 + 2^q \times (a_1 + q \times a_2) \text{ if } q = \log_2((d+1)L), \\ q \times a_0 + 2^q \times (a_1 + q \times a_2) + 2^q \times a_3 \text{ otherwise.} \end{cases}$$

Example 4. For our AES implementation, we got $a_0 = 3$, $a_1 = 15$ and $a_2 = 14$, $a_3 = 17$.

Generation the Masked S-box. Its complexity $\mathscr{C}_{MS}$ satisfies:

$$\mathscr{C}_{MS} = d' \times a_0 \ .$$

Example 5. For our AES implementation, we got $a_0 = 4352$.

Protected keyed Substitution Layer. Its complexity $\mathscr{C}_{SL}$ satisfies:

$$\mathscr{C}_{SL} = t \times (a_0 + d \times a_1 + d' \times a_2) \ .$$

Example 6. For our AES implementation, we got $a_0 = 55$, $a_1 = 37$ and $a_2 = 18$.

Protected Linear Layer. Its complexity $\mathscr{C}_{LL}$ satisfies:

$$\mathscr{C}_{LL} = (d+1)L \times a_0 \ .$$

Example 7. For our AES implementation, we got $a_0 = 169$.

10-3 Attack Paths

In this section, we list attacks combining higher-order and integrated DPA that may be attempted against our scheme. Section 9 is then involved to associate each attack with a correlation coefficient that depends on the leakage noise deviation σ, the block cipher parameters (n, N, l', L) and the security parameters (d, d', t). As argued, these coefficients characterize the attacks efficiencies and hence the overall resistance of the scheme.

Remark 32. In this paper, we only consider known plaintext attack *i.e.* we assume the different sensitive variables uniformly distributed. In a chosen plaintext attack, the adversary would be able to fix the value of some sensitive variables which could yield better attack paths. We do not take such attacks into account and let them for further investigations.

Each sensitive variable in the scheme is (1) either masked with d unique masks or (2) masked with d' masks shared with other sensitive variables (during the keyed substitution layer).

(1). In the first case, the $d+1$ shares appear during the keyed substitution layer computation and the linear layer computation. In both cases, their manipulation is shuffled.

(1.1). For the keyed substitution layer (see Algorithm 1), the $d+1$ shares all appear during a single iteration of the loop among t. The attack consists in combining the $d+1$ corresponding signals for each loop iteration and to sum the t obtained combined signals. Proposition 7 implies that this attack can be associated with the following correlation coefficient ρ_1:

$$\rho_1(t,d) = \frac{1}{\sqrt{t}}\rho(n,d,\sigma) \ . \tag{10.4}$$

(1.2). For the linear layer (see Algorithm 3), the $d+1$ shares appear among $(d+1)\cdot L$ possible operations. The attack consists in summing all the combinations of $d+1$ signals among the $(d+1)\cdot L$ corresponding

signals. According to Proposition 9, this attack can be associated with the following correlation coefficient ρ_2:

$$\rho_2(L,d) = \frac{1}{\sqrt{\binom{(d+1)\cdot L}{d+1}}}\rho(l',d,\sigma) \; . \tag{10.5}$$

Remark 33. In the analysis above, we chose to not consider attacks combining shares processed in the linear layers together with shares processed in the keyed substitution layer. Actually, such an attack would yield to a correlation coefficient upper bounded by the maximum of the two correlations in (10.4) and (10.5).

(2). In the second case, the attack targets a d'^{th}-order masked variable occurring during the keyed substitution layer. Two alternatives are possible.

(2.1). The first one is to simultaneously target the masked variable (that appears in one loop iteration among t) and the d' masks that appear at fixed times (*e.g.* in every loop iteration of Algorithm 1 or during the masked S-box computation). The attack hence consists in summing the t possible combined signals obtained by combining the masked variable signal (t possible times) and the d' masks signals (at fixed times). According to Proposition 9, this leads to a correlation coefficient ρ_3 that satisfies:

$$\rho_3(t,d') = \frac{1}{\sqrt{t}}\rho(n,d',\sigma) \; . \tag{10.6}$$

(2.2). The second alternative is to target two different variables both masked with the same sum of d' masks (for instance two masked S-box inputs or outputs). once among $t\cdot(t-1)$ pairs of iterations. These variables are shuffled among t variables. The attack hence consists in summing all the possible combinations of the two signals among the t corresponding signals. According to Proposition 9, this leads to a correlation coefficient ρ_4 that satisfies:

$$\rho_4(t) = \frac{1}{\sqrt{t\cdot(t-1)}}\rho(n,2,\sigma) \; . \tag{10.7}$$

10-4 Parameters Setting

The security parameters (d,d',t) can be chosen to satisfy an arbitrary resistance level characterized by an upper bound ρ^* on the correlation

Tab. 10-5-1 – Timings for the different steps of the scheme for an AES implementation on a 8051-architecture.

T Generation	$\mathscr{C}_T = 112 + t\left(6 + 9\sum_{i=0}^{15} \frac{1}{t-i}\right)$
T' Generation	$\mathscr{C}_{T'} = 3q + 2^q(15 + 14q) \quad [+17 \times 2^q]$
Masked S-box Generation	$\mathscr{C}_{MS} = 4352d'$
Pre-computations	$\mathscr{C}_T + \mathscr{C}_{T'} + \mathscr{C}_{MS}$
Substitution Layer	$\mathscr{C}_{SL} = t(55 + 37d + 18d')$
Linear Layer	$\mathscr{C}_{LL} = 676(d+1)$
Protected Round	$\mathscr{C}_{SL} + \mathscr{C}_{LL} = 676(d+1) + t(55 + 37d + 18d')$
Unprotected Round	432

coefficients corresponding to the different attack paths exhibited in the previous section. That is, the parameters are chosen to satisfy the following inequality:

$$\max(|\rho_1|, |\rho_2|, |\rho_3|, |\rho_4|) \leqslant \rho^* \ . \tag{10.8}$$

Among the 3-tuples (d, d', t) satisfying the relation above, we select one among those that minimize the timing complexity (see Sect. 10-2).
Remark 34. detailed here. Regarding the system dimension, this can be done exhaustively.

10-5 Application to AES

We implemented our scheme for AES on a 8051-architecture. According to Remark 4, the ShiftRows and the MixColumns were merged in a single linear layer applying four times the same operation (but with different state indexings). The block cipher parameters hence satisfy: $n = 8$, $N = 16$, $l = 32$, $l' = 8$ and $L = 4$.
Remark 35. In [46], it is claimed that the manipulations of the different bytes in the MixColumns can be shuffled. However it is not clear how to perform such a shuffling in practice since the processing differs according to the byte index.

Table 10-5-1 summarizes the timings obtained for the different steps of the scheme for our implementation.
Remark 36. The implementation of unprotected rounds has been optimized, in particular by only using variables stored in DATA memory.

Tab. 10-5-2 – Optimal parameters and timings according to SNR and ρ^*.

	ρ^*	t	d	d'	timings
SNR = $+\infty$	10^{-1}	16	1	1	3.66×10^4
	10^{-2}	20	3	3	8.57×10^4
	10^{-3}	1954	4	3	5.08×10^6
	10^{-4}	195313	5	3	5.75×10^8
SNR = 1	10^{-1}	16	1	1	3.66×10^4
	10^{-2}	20	2	2	6.39×10^4
	10^{-3}	123	3	3	3.13×10^5
	10^{-4}	12208	4	3	3.15×10^7
SNR = $\frac{1}{4}$	10^{-1}	16	1	0	2.94×10^4
	10^{-2}	16	1	1	3.66×10^4
	10^{-3}	16	2	2	5.75×10^4
	10^{-4}	19	3	3	8.35×10^4

Because of memory constraints and due to the scalability of the code corresponding to the protected round, many variables have been in stored in XDATA memory which made the implementation more complex. This explains that, even for $d = d' = 0$ and $t = 16$ (*i.e.* when there is no security), the protected round is more time consuming than the unprotected round.

We give hereafter the optimal security parameters (t, d, d') for our AES implementation according to some illustrative values of the device noise deviation σ and of correlation bound ρ^*. We consider three noise deviation values: 0, $\sqrt{2}$ and $4\sqrt{2}$. In the Hamming weight model, these values respectively correspond to a signal-to-noise ratio (SNR) to $+\infty$, 1 and $\frac{1}{4}$. We consider four correlation bounds: 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4}. The security parameters and the corresponding timings for the protected AES implementation are given in Table 10-5-2. Note that all the rounds have been protected.

When SNR = $+\infty$, the bound $d' \leqslant 3$ implies an intensive use of shuffling in the keyed substitution layer. The resulting parameters for correlation bounds 10^{-3} and 10^{-4} imply timings that quickly become prohibitive. A solution to overcome this drawback would be to design secure table recomputation algorithms for $d' \geqslant 3$. Besides, these timings underline the difficulty of securing block ciphers implementations with pure software

countermeasures. When the leakage signals are not very noisy (SNR = 1), timings clearly decrease (by a factor from 10 to 20). This illustrates, once again, the soundness of combining masking with noise addition. This is even clearer when the noise is stronger (SNR = $\frac{1}{4}$), where it can be noticed that the addition of dummy operations is almost not required to achieve the desired security level.

Part IV

Perspectives

THIS thesis deals with side channel attacks against hardware implementations of cryptographic algorithms. Studies led in this document are therefore in place where an adversary has access to noisy observations of intermediate results of a cryptographic computation. In this context, many attacks are dedicated with their countermeasures, but their relevance and their implementation are still unclear.

This thesis initially focuses on the relevance of existing attacks and potential links between them. A formal classification is proposed as well as selection criteria. Based on this study, a generic efficient attack is described and analysed in depth.

In a second step, the implementation of common countermeasures is studied, leading to the creation of an application scheme mixing them to achieve a better efficiency / security trade off.

In conclusion, this thesis presents a unification of the various existing side channel attacks, introduces new attack techniques which are more robust to errors in the modeling steps and proposes a new scheme to counteract these attacks.

Although some points are still open, new points arise and are a matter of interest. Namely, in Part III a generic framework combining common countermeasures is given based on the CPA. This attack was chosen as its distinguisher value is directly linked to the efficiency of the attack (see Sect. 4-6). Nevertheless as argued in Chap. 5 the main drawback of CPA is the choice of a relevant model. Thus an interesting extension of our work could be to make an equivalent study based on MIA instead of CPA. Such a work would bring a theoretical basis to study the efficiency of a countermeasure against an optimal attack.

In Part II, a formal classification is given. Nevertheless it can be increased by the study of new – or more unconventional – attacks. In Chap. 6 the linear regression based attack is introduced and its link with CPA is shown. An interesting question would be to exhibit a link with mutual information based attack for instance starting from *Least Square Mutual Information* [101]. In a more practical view, deeper study of linear regression would be profitable. For example algorithm optimization with respect to the basis choice as initiated in Sect. 6-2.2 would permit to have a more generic and more efficient attack. Another example is

the new promising distinguisher introduced in Sect. 7-3.

In a more general context, it would be interesting to explore high order attacks in practice. For instance the approach based on the EMA algorithm [58] or based on optimization theory [87] are quite exciting. Another turbulent domain which is not treated in this thesis is the domain of profiling attacks. Reducing the gap between profiling and non-profiling attacks is a real challenge.

Eventually, the countermeasure aspect has to be treated too. Currently, to counteract high-order attacks, only one generic scheme exists [29] and this scheme is very costly in practice. Thus reducing the overhead of this framework or find a less costly framework is very important as high-order attacks become efficient in practice.

Part V

Appendix

APPENDIX A

Extra Data From Experimentations in Sect. 8-1 (part II)

In Sect. 8-1, several experimentations have been conducted, resulting in various sets of results. Some of the data sets have been adapted (*e.g.*, fitted, truncated) to become more readable. For informational purpose we plot the whole data set in this section. Figure 1 shows us the raw data use for fitting in Fig. 8-1-1.

Figure 2 shows us the evolution of the success rate according to the number of messages and the noise deviation in the random leakage scenario. Figures 8-1-3 and 8-1-4 are extracted from Fig. 2.

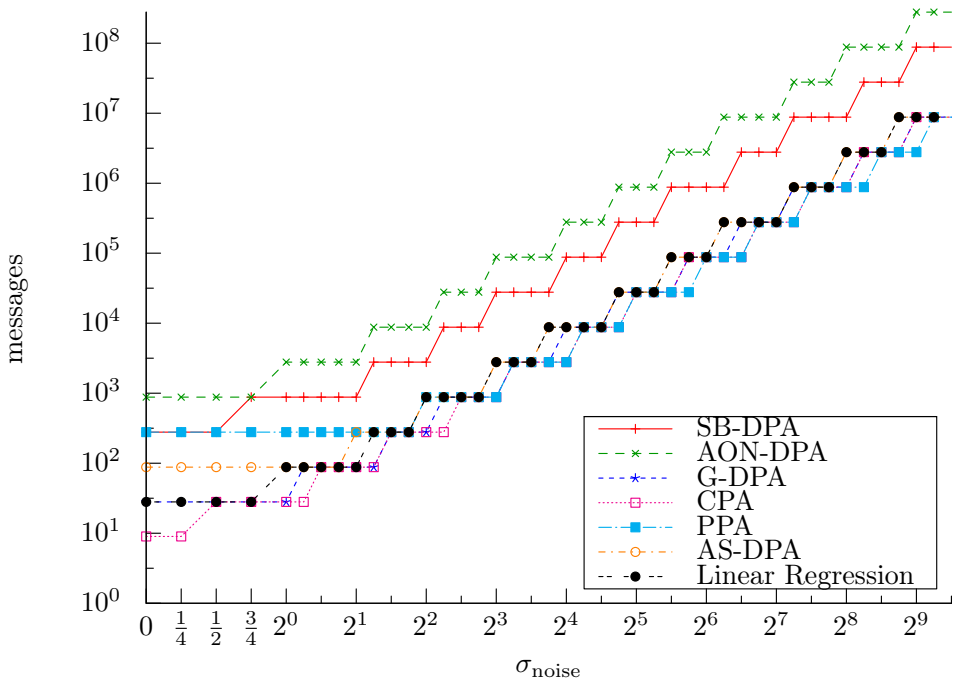


Fig. 1 – Evolution of the number of messages needed to achieve a success rate of 90% for different noise values.

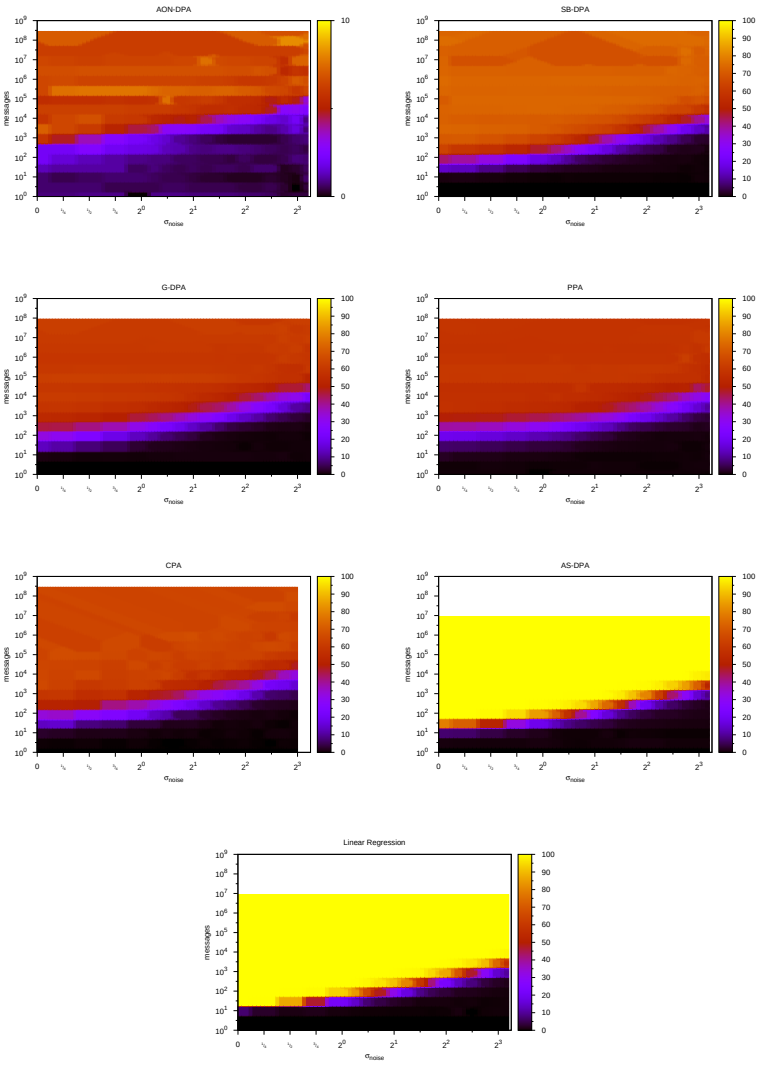


Fig. 2 – evolution of the success rate according to the number of messages and the noise deviation.

APPENDIX $\mathcal{B}$

AES S-box w_i values

Tab. 1 – Values of w_i and $\sum w_i$ for the AES S-box – values must be divided by 256.

a	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	$\sum w_i$
0	256	256	256	256	256	256	256	256	2048
1	-48	-28	0	-68	24	16	48	-72	304
2	-8	8	-16	-68	40	-16	40	4	200
3	0	40	32	8	0	84	48	4	216
4	64	-52	92	52	-44	-12	-12	40	368
5	4	24	-4	-28	-20	-20	-68	-88	256
6	64	-24	24	12	-48	20	40	16	248
7	36	48	-52	4	48	-40	84	-16	328
8	40	-60	20	24	76	84	-24	-96	424
9	-84	-40	-24	-52	12	40	-24	36	312

10	60	0	56	4	-12	24	68	72	296
11	-48	-108	-60	-20	-60	-4	-4	-56	360
12	0	-36	32	-52	-108	32	12	40	312
13	40	-28	-64	60	80	-36	40	-52	400
14	48	-28	100	48	44	64	-20	-24	376
15	12	-12	-4	-56	32	56	-4	-40	216
16	-44	-16	-36	-52	-40	64	-48	-12	312
17	-36	28	-96	8	12	-44	20	-36	280
18	-36	24	-60	40	-32	-36	-12	-56	296
19	-44	-20	-76	-28	80	-64	36	-84	432
20	40	4	56	-68	-12	-24	24	76	304
21	-32	-8	-48	-48	-44	24	36	-32	272
22	20	44	60	140	12	0	16	68	360
23	-36	24	-52	-20	44	-48	-68	-4	296
24	-20	28	44	20	-12	52	-80	16	272
25	-40	12	8	-56	16	-72	68	-56	328
26	-16	8	-32	-16	-100	44	12	100	328
27	-20	36	-8	-104	32	-16	-20	-12	248
28	-56	48	-12	-28	-28	-20	-32	128	352
29	20	48	-104	-76	-68	-80	-40	-52	488
30	100	12	-44	-4	-56	24	-24	80	344
31	-104	-20	-32	4	8	0	-60	20	248
32	60	44	-16	88	56	-12	4	-104	384
33	-76	0	-88	-24	-44	-36	16	-60	344
34	8	24	12	-4	52	60	-60	76	296
35	-52	-48	-16	-28	-24	-64	-28	-68	328
36	20	24	16	44	-4	56	52	40	256
37	-48	76	-28	8	60	84	-12	-76	392
38	32	16	44	-84	52	60	20	44	352
39	-24	52	0	64	0	68	36	52	296
40	64	64	-68	-36	60	28	16	56	392
41	-32	-32	-52	-40	-32	-32	-8	-4	232
42	60	16	32	-36	16	-32	8	16	216
43	-20	-32	36	8	28	32	-24	-20	200

44	12	-60	44	-56	-64	20	-36	84	376
45	40	-28	-16	-12	32	4	4	-48	184
46	32	-20	-28	-84	-36	44	-48	-20	312
47	20	80	-92	32	64	8	-64	-8	368
48	68	20	16	36	28	-40	-16	104	328
49	-12	52	0	-60	-28	-32	-44	36	264
50	92	-28	48	-44	-8	0	76	40	336
51	24	-12	80	16	96	52	40	16	336
52	-4	24	24	40	40	-40	-16	68	256
53	-4	-60	-120	-84	68	-100	-12	-48	496
54	24	-36	44	-8	-16	-60	48	-44	280
55	4	68	8	-40	-60	16	-52	-24	272
56	24	-88	-8	24	-52	76	-28	12	312
57	24	12	-36	-32	-16	-32	-60	36	248
58	-32	-36	-32	40	-72	-4	-80	40	336
59	-72	60	16	-12	-84	-76	-32	-32	384
60	-28	-40	28	-112	-92	-8	-8	-44	360
61	-8	-28	84	24	-44	72	-20	-16	296
62	104	32	36	96	72	-4	36	12	392
63	-64	-40	-76	-124	-8	-20	16	-92	440
64	64	-12	60	-20	-16	-8	-104	28	312
65	12	-48	32	-60	-88	-8	-12	28	288
66	112	-16	-12	8	-32	48	0	76	304
67	-8	44	-24	68	-16	-16	-20	-60	256
68	20	64	-40	52	8	-28	-12	48	272
69	-28	28	-68	-36	-20	-28	44	44	296
70	-24	40	72	4	60	4	16	-36	256
71	20	20	-32	-32	-8	40	-96	-88	336
72	56	16	84	52	-20	20	12	-12	272
73	-20	-52	-32	0	76	-12	4	-12	208
74	-100	-40	-44	-8	-24	40	-36	-36	328
75	8	28	-12	36	-48	12	-28	-28	200
76	8	-12	-20	-16	-16	-96	0	-48	216
77	-32	-56	-32	-24	40	-12	-76	-80	352

78	-36	36	44	4	-32	-12	-12	-48	224
79	0	56	-60	-20	-60	48	24	-28	296
80	-4	-68	-36	40	-36	-16	24	-80	304
81	36	-40	-20	-24	32	-16	-44	-4	216
82	-12	-8	8	64	72	-68	48	80	360
83	8	-12	-40	-76	-12	-28	-20	-36	232
84	-8	-48	-16	8	16	-36	8	44	184
85	20	96	-52	-40	100	52	52	20	432
86	64	8	44	36	56	-16	-16	-24	264
87	-16	-76	-24	-44	-4	-24	-44	-16	248
88	68	20	-20	-20	40	-20	-56	12	256
89	-76	-20	84	-28	-36	28	4	-44	320
90	-24	40	-4	12	-32	-24	76	4	216
91	-12	-8	-24	-20	36	0	0	-4	104
92	-32	104	-4	40	48	24	36	56	344
93	44	40	-52	-20	40	12	24	-64	296
94	-60	-40	24	56	-76	36	32	12	336
95	16	92	-40	-20	0	8	24	72	272
96	32	-16	-60	-60	8	-4	0	-44	224
97	-32	24	-60	24	0	-8	-36	0	184
98	0	16	12	48	88	-40	-20	24	248
99	-56	-24	32	-52	20	48	12	-60	304
100	56	12	8	-24	48	12	76	4	240
101	96	-24	-64	28	-92	48	-68	-4	424
102	64	36	20	12	28	-80	76	-4	320
103	0	24	-24	-40	-64	20	16	-20	208
104	92	68	-16	-4	-16	-16	8	-36	256
105	-80	12	16	-56	-44	-20	-4	-80	312
106	-24	-68	-32	-96	36	12	12	-24	304
107	132	56	-40	0	-8	52	4	-44	336
108	12	44	92	-20	12	-60	4	92	336
109	-32	76	-36	20	4	88	-64	80	400
110	-32	-48	12	-24	-12	0	36	-68	232
111	-20	16	4	28	12	-44	-24	68	216

112	68	28	60	-8	76	0	-16	-8	264
113	-4	-20	-80	20	0	60	52	-20	256
114	-36	-40	-56	-12	-24	-8	104	32	312
115	36	32	44	-44	-48	12	-44	92	352
116	-76	-116	-20	-32	20	-40	12	28	344
117	16	120	64	28	36	88	-24	16	392
118	24	-32	-16	60	4	32	16	24	208
119	48	-28	-40	-60	76	-48	20	16	336
120	-76	-8	24	40	-32	20	72	-16	288
121	-32	8	20	60	-80	36	60	56	352
122	36	0	-72	32	80	-20	12	76	328
123	-20	32	-32	60	108	68	-40	40	400
124	-44	-76	-44	-56	-32	-24	32	44	352
125	-4	8	-60	68	-28	40	72	-24	304
126	-20	-48	-4	72	-12	-44	-48	8	256
127	-28	0	52	-36	-32	4	-16	-40	208
128	-24	-52	-36	-56	4	-16	-16	-100	304
129	52	16	-16	44	80	-32	-32	-40	312
130	-32	-32	-44	-12	-72	-36	-44	-88	360
131	-36	-4	-36	-92	-24	-52	-24	12	280
132	52	-44	12	28	28	-44	-40	24	272
133	-20	24	-12	-32	-92	-48	4	-40	272
134	12	36	20	8	20	-44	-4	-24	168
135	-28	0	64	56	0	24	-52	-16	240
136	8	24	8	-24	44	-44	-16	0	168
137	72	124	0	24	52	28	12	-8	320
138	4	36	-36	48	0	-60	16	-48	248
139	24	44	0	-100	-28	8	24	28	256
140	-8	-28	16	-24	36	-12	16	-4	144
141	4	-20	-24	0	-108	-52	-32	-24	264
142	-4	32	-16	64	-32	-36	56	-48	288
143	-12	24	12	8	-64	32	28	-44	224
144	24	24	0	4	112	-76	12	44	296
145	36	-12	8	92	-12	-32	-16	104	312

146	112	-4	84	32	-8	-32	-36	60	368
147	0	-80	72	12	-40	32	0	-28	264
148	-84	-8	72	-56	-68	-56	-20	-36	400
149	-4	-8	-12	52	-40	-20	-12	100	248
150	-24	8	-20	56	-20	24	80	8	240
151	0	0	-68	32	28	76	-48	52	304
152	8	-48	88	80	48	4	20	16	312
153	-120	-44	0	32	-60	0	-40	-96	392
154	12	-36	-32	0	-8	-28	-16	-44	176
155	-20	-48	-72	44	-40	-36	24	76	360
156	28	-8	60	32	104	0	96	32	360
157	-48	-96	-48	-44	-4	64	12	-4	320
158	-72	0	20	-8	52	-24	44	-52	272
159	-8	68	32	88	8	8	40	28	280
160	36	-36	80	20	-40	-8	-32	-92	344
161	60	0	12	32	28	72	-12	0	216
162	36	72	16	40	12	-44	36	64	320
163	0	-44	36	-56	16	4	-4	16	176
164	-12	20	36	-52	8	20	-40	-4	192
165	24	4	4	-4	24	-32	52	0	144
166	-56	-36	28	-4	12	44	-72	28	280
167	-84	-60	-12	-80	20	-56	-60	-52	424
168	-4	-56	16	68	-36	0	-4	24	208
169	-76	-60	-60	-24	-56	-40	36	-32	384
170	-68	-84	80	-12	-32	-52	-40	-24	392
171	20	-20	24	-12	44	40	-44	12	216
172	-32	-8	52	12	-4	-4	8	24	144
173	-12	-36	-32	-56	-52	24	64	-60	336
174	36	56	12	8	52	48	60	72	344
175	64	-24	-24	-12	40	40	-72	12	288
176	28	24	-28	-4	-12	68	-64	28	256
177	-72	8	-44	8	12	16	0	-56	216
178	-44	-28	4	32	-20	32	4	-68	232
179	32	-16	-76	-8	-52	-40	64	-24	312

180	-28	-4	32	0	4	-4	32	32	136
181	-8	-4	12	4	-88	12	-32	24	184
182	0	16	24	-8	-20	-20	40	0	128
183	0	-40	-28	20	48	28	24	-28	216
184	48	112	0	48	-8	72	36	52	376
185	-32	40	-8	-16	-4	0	16	4	120
186	4	20	100	-16	60	-48	-16	8	272
187	-24	-68	-60	-32	-72	16	44	-52	368
188	20	36	60	68	8	-4	4	-40	240
189	-132	28	-64	-16	-24	-28	-44	-40	376
190	-32	8	4	16	-4	-24	20	60	168
191	-12	52	4	20	-4	4	72	32	200
192	-64	0	-40	28	8	32	-20	40	232
193	-40	8	48	-28	-16	56	-20	-48	264
194	-72	-44	-56	24	56	4	-32	-32	320
195	-52	96	-32	20	52	4	-60	52	368
196	4	-28	-16	0	96	-20	0	-28	192
197	-32	-48	20	28	-16	92	44	-8	288
198	4	-44	40	60	-40	48	64	28	328
199	-68	0	52	-32	32	-4	-32	-44	264
200	-36	32	12	-4	-20	-28	-28	-56	216
201	68	48	4	20	32	60	4	76	312
202	48	64	8	40	64	4	68	72	368
203	-24	-32	8	-52	-64	32	-12	24	248
204	-44	12	44	24	92	-52	0	-4	272
205	72	16	-28	-24	-40	12	8	56	256
206	56	-32	-36	12	-12	-64	4	24	240
207	-12	-4	36	4	-8	36	-36	32	168
208	-72	52	156	20	-40	28	24	0	392
209	-20	-32	40	-40	-36	68	-68	-72	376
210	20	-36	-20	-32	16	-48	8	-20	200
211	-12	36	36	16	-80	64	80	92	416
212	92	60	-36	-12	76	16	-4	56	352
213	12	-8	24	20	8	56	36	28	192

214	-72	4	-16	-96	-44	12	-16	-44	304
215	0	44	28	40	-92	8	44	-8	264
216	56	16	-32	24	4	-68	8	-80	288
217	-12	-48	-8	24	-68	-44	64	-4	272
218	0	-36	-4	76	28	-36	36	-8	224
219	4	-20	0	-8	-32	-52	-72	-92	280
220	28	4	-32	-12	-8	0	-28	-40	152
221	-72	-4	88	8	-84	-44	-48	-20	368
222	32	-76	-12	-48	56	16	32	8	280
223	-36	-36	-24	-8	-44	36	-52	-52	288
224	-64	-52	24	48	-12	8	4	36	248
225	12	20	-16	0	-44	-28	88	40	248
226	64	-28	0	28	-12	0	8	12	152
227	-32	-20	84	-76	24	4	48	16	304
228	-28	28	-28	56	56	120	-84	80	480
229	-48	-36	-16	-44	56	-16	32	-32	280
230	68	-44	24	4	36	-64	4	12	256
231	32	-28	44	28	60	68	0	28	288
232	-20	-48	32	64	-76	-20	-84	16	360
233	48	-28	-20	68	8	-52	-28	-44	296
234	8	64	-4	136	-12	4	44	32	304
235	12	16	28	4	8	-52	32	32	184
236	116	-8	-12	12	76	-52	28	0	304
237	-24	-32	32	20	12	68	100	16	304
238	-16	4	-12	96	-44	-16	8	-4	200
239	44	-28	32	0	-80	-52	-12	-120	368
240	-16	60	-28	8	-24	-64	-32	48	280
241	-60	-4	-40	-20	-4	40	-8	0	176
242	-104	-8	60	16	8	56	-40	12	304
243	-4	-8	-28	-68	36	-60	-40	28	272
244	40	44	0	20	24	0	-52	-4	184
245	-104	-92	-56	-16	-72	4	-44	-92	480
246	-56	-40	16	-44	-28	0	8	-40	232
247	48	4	28	40	4	24	8	4	160

248	48	16	88	-64	52	-60	12	12	352
249	44	-28	32	-84	-28	-52	-24	-44	336
250	68	40	-24	-16	60	-32	-48	56	344
251	-76	32	-56	-112	-56	28	20	-4	384
252	44	-24	20	56	32	-48	-96	0	320
253	-12	-24	-28	0	-68	-88	-32	60	312
254	100	28	16	28	-16	-24	-28	8	248
255	0	-28	-36	-36	40	-52	-76	-28	296

APPENDIX C

Full Basis Linear Regression

As pointed out in Sect. 7-3, the linear regression outputs a function in its normal algebraic form and we suggest that the normal algebraic form can permit to discriminate the key. In the following we have performed a few experiments to validate our suggestion. These experiments need more deeper analyses that why they are provide as-is in this appendix.

C-1 Simulation

First of all we are interested in the distribution of the coefficient of the algebraic normal form. A straightforward way to analyse this distribution is to take a look on the cumulative distribution function (which is in fact equivalent to analyse the shape of the sorted list of coefficient). Here we will draw the coefficients sorted (without normalization). We also studied the deviation from the mean, that is we compute the av-

erage function g over the whole candidates (*i.e.* each coefficient of the average function g is the mean of the correspinding coefficient of every function).

Attacks description. The 8-bit output of the AES sbox is targeted. Leakages are simulated according to three scenarios which depend of the deterministic part of the leakage δ:

1. δ is the Hamming weight function

2. δ is a linear function in the bit of the sensitive variable, with coefficients picked uniformly into $[-1,1]$

3. δ is a quadratic function in the bit of the sensitive variable, with coefficients picked uniformly into $[-1,1]$

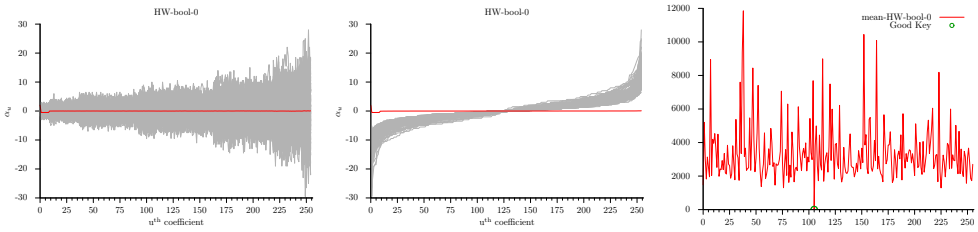
For each scenario, only one attack has been performed with a set of $100,000,000$. That implies that for scenario 2 and 3, only one δ is generate, which is different for each session.

The attack is a linear regression with the full basis over the centered product combined leakages. In the following we have recorded the value of each coefficient for each key guess (leftmost figure, the good key is drawn in red). We have also drawn the coefficients sorted by value (center figure, the good key is drawn in red). Then we compute the Euclidean distance between the coefficient vector of each key guess with the average function (righmost figure).

Remark 37. The attacks have been performed in a noiseless context. The same attacks have also been performed with noise and the results are approximately the same and thus are not reported here.

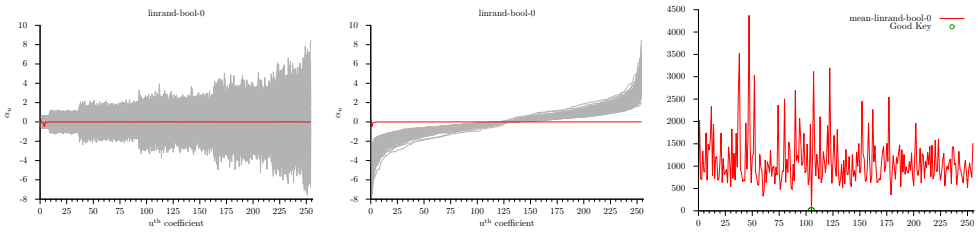
C-1.1 Boolean Masking

C-1.1.1 Scenario 1



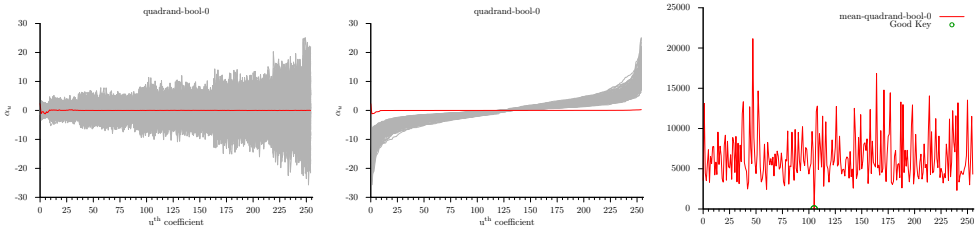
Due to the small degree of the combined function, only a small fraction of the coefficient for the good key have a nonzero value. Where the key hypothesis is wrong, each coefficient plays a role due to the complexity of $F_k \circ F_{\hat{k}}^{-1}$. Moreover we observe that the mean of the coefficients tends toward zero. That is why in the rightmost figure, the good hypothesis have the lower distance from the average function.

C-1.1.2 Scenario 2



The combined fonction has still a small algebraic degree, thus the same observation as in the first scenario can be made.

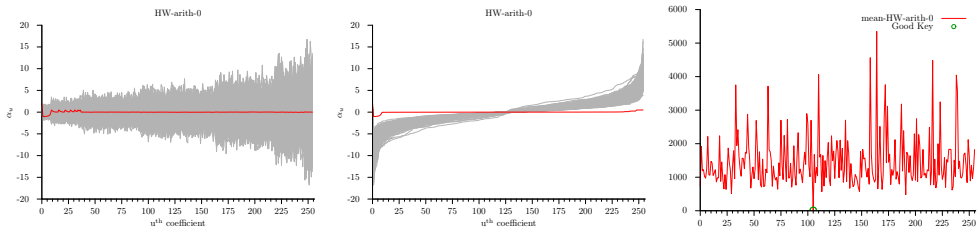
C-1.1.3 Scenario 3



Same observations as for scenario 1 and 2.

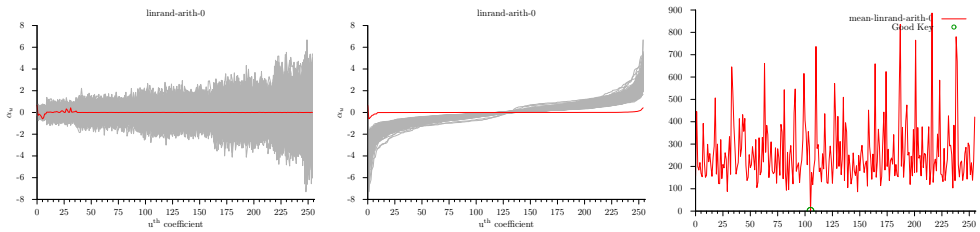
C-1.2 Arithmetic Masking

C-1.2.1 Scenario 1



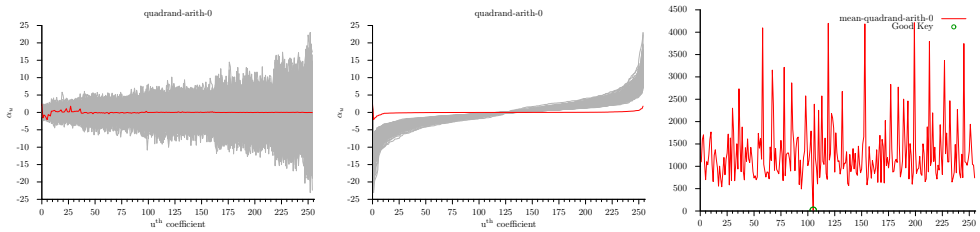
We observe that the combined function has still a small algebraic degree and so same observations as with boolean masking can be done.

C-1.2.2 Scenario 2



The same observations as for Boolean case can be done.

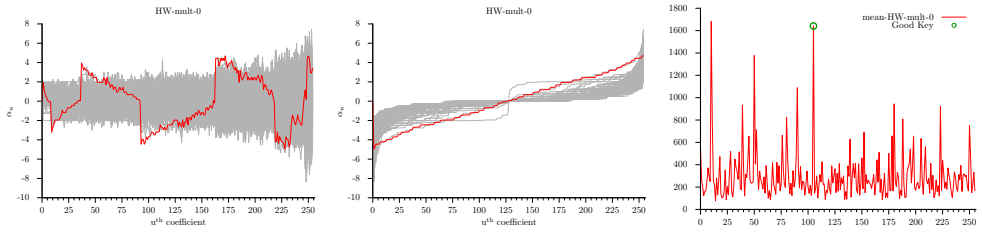
C-1.2.3 Scenario 3



The same observations as for Boolean case can be done.

C-1.3 Multiplicative Masking

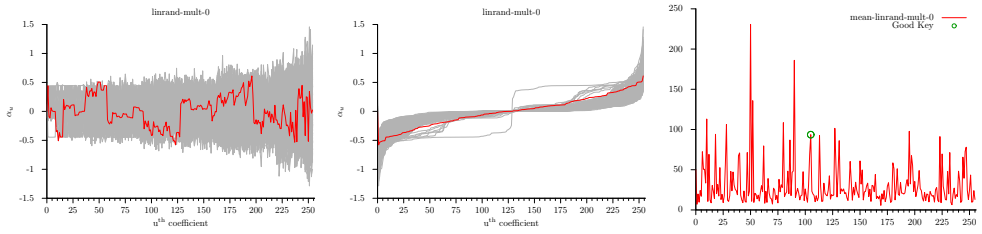
C-1.3.1 Scenario 1



With multiplicative masking, even if δ is the Hamming weight function, the centered product combination seems to have a more complex algebraic form. The curve for the good hypothesis does not seem to emerge from the other curves. When sorted, we can see that the coefficients for the good key seems to be uniformly distributed whereas for a wrong hypothesis it is not uniformly distributed. In this case, testing the uniformity of the distribution of the coefficient may be a good approach. Nevertheless, some other keys seems to have a very similar behavior (*i.e.* uniformity will not discriminate the good key but the good key will be among a few ones). Thus the distance to the average function does not permit to discriminated the good key.

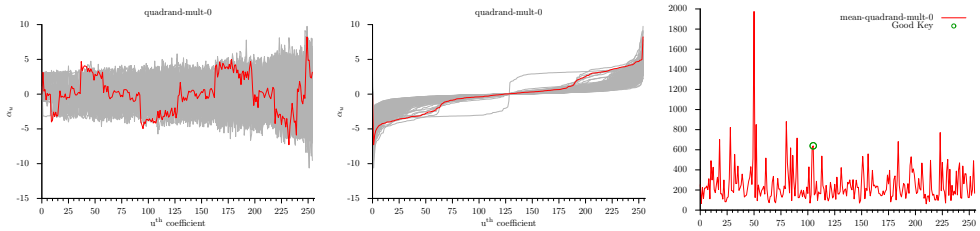
Remark 38. Notice that we are aware of the fact that this masking is flawed at first-order (due to the masking of zero, see [44]). Nevertheless it allow us to validate our approach.

C-1.3.2 Scenario 2



When δ is not the Hamming weight, the curve for the good key seems to be a little *noisier* nevertheless when sorted, the coefficient seems to have a very similar behavior as in HW case.

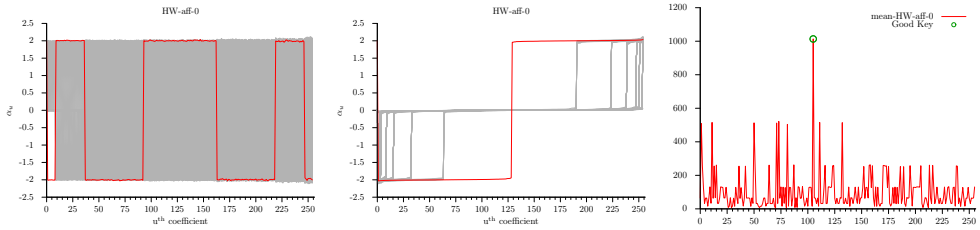
C-1.3.3 Scenario 3



Same observations as in the linear random case.

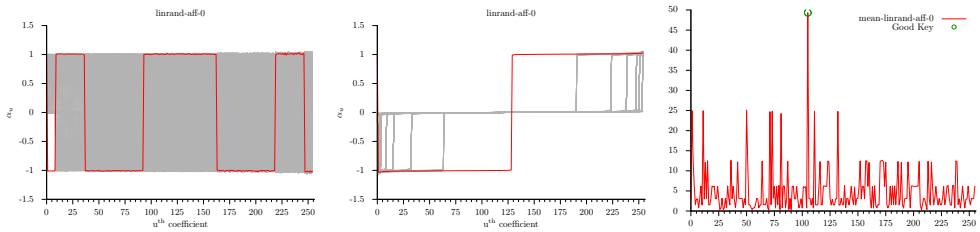
C-1.4 Affine Masking

C-1.4.1 Scenario 1



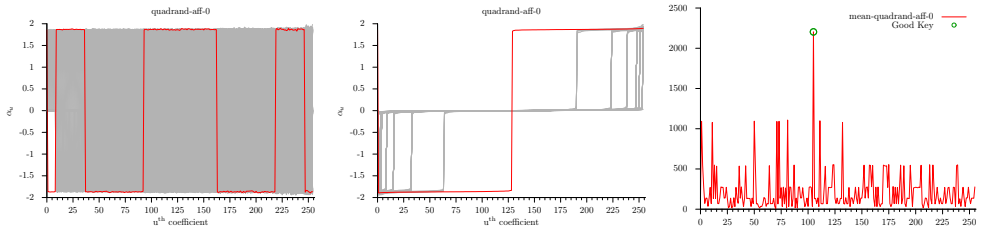
When δ is the Hamming weight, we know that the combined function is a dirac function (cf. [40]). Although the curves have a characteristic shape it can be dissociated from others. Nevertheless when the coefficient are sorted, we can observe that the good curves is the one with the smallest number of zero coefficient. This implies that at the opposite of the Boolean and arithmetic case, the distance to the average function is maximize.

C-1.4.2 Scenario 2



Same observation as in the Hamming weight case.

C-1.4.3 Scenario 3



ibidem.

C-2 Conclusion

First we observe that the masking scheme has an huge impact on the combined function and thus the discrimination process must be adapt in accordance with the scheme. Moreover, while δ is kept to a simple algebraic function, it does not really impact the complexity of the combined function. Nevertheless, although δ is a simple function, the combined function may not (cf. affine case).

The relevant point seems to be the shape – complexity – of the combined function which rely on the masking scheme. When it has a low algebraic degree (such as with Boolean and arithmetic masking), each higher degree coefficient will be zero and is thus sufficient to discriminate the good key. When it is has not a low algebraic degree function, we do not have such a generic remark but ad-hoc one. For instance, for a dirac function (*e.g.* affine masking), the good curve has only non-zero coefficients whereas others curves seems to have zero coefficients. The zeroiness of the coefficient is in those case a good discriminating method.

The case of multiplicative masking is more tricky as no specific behavior seems to outcomes. Perhaps another combination function can bring more information.

Finally, we have a powerful attack which outpass the drawback of the basis building in linear regression but which still failed in multiplicative masking. Moreover, we have tested it only for small algebraic degree δ function. We have also validate the retro-engineering approach of this attack. It open the door to a real interesting way of lead such attack.

APPENDIX D

Résumé en Français

D-1 Remerciements

FINALEMENT cette thèse arrive à sa fin. Il s'agit d'une étape importante à franchir et je tiens à remercier toutes les personnes ayant contribué à son succès que ce soit par leur apport ou leur soutien aussi bien sur le plan professionnel que personnel durant toute cette intéressante mais éprouvante période.

En premier lieu, Emmanuel Prouff, qui m'a donné l'opportunité de mener cette thèse au sein d'Oberthur Technologies. Il a su avoir la patience et le dévouement nécessaire pour me transmettre sa passion pour la recherche. Ses nombreuses remarques et relectures, ses conseils, ses idées, sa disponibilité ont permis de mener cette thèse au point où elle est aujourd'hui.

Ensuite je tiens à remercier mes deux directeurs de thèse, François-

Xavier Standaert et Claude Carlet pour m'avoir fait l'honneur de m'encadrer et me faire profiter de leur inestimable expérience.
Je suis heureux que Werner Schindler et Elisabeth Oswald soient mes rapporteurs et aient accepté de consacrer du temps à la relecture minutieuse de mon manuscript.
Je tiens à remercier François Koeune de faire partir de mon jury d'accompagnement et mon jury final de thèse et d'avoir su être garant de l'esprit belge au sein d'une certaine crêperie bretonne.
Merci encore à Louis Goubin et Ingrid Verbauwhede pour leur présence dans mon jury et leurs pertinentes remarques.

Durant toute cette période de préparation de ma thèse j'ai été amené à croiser de nombreuses personnes qui ont toutes su m'apporter quelque chose. A défaut d'être surement exhaustif je tiens tout d'abord à remercier les personnes croisées au sein d'Oberthur Technologies.
Paul Dischamp a qui je suis reconnaissant d'avoir permis cette thèse au sein de l'entreprise et ainsi m'inculquer l' »esprit entreprise ». Emmanuelle Dottax qui à eu la lourde tâche de me détourner de ma thèse *clin d'oeil*
Matthieu Rivain pour m'avoir laissé sa place d'éminent doctorant.
Je suis malheureusement obliger de citer et remercier Gilles Piret qui a réussi à me belgifier avec l'aide sournoise d'un dénommé JP. Que de nombreuses heures passées à refaire la jounée et le monde.
Je n'oublie pas Yannick Sierra (et ses « filles »), Guillaume Dabosville (et sa grosse . . . pointe de dérision), Laurie Genelle (et ses petits . . . tics de rangement), Robert Naciri le vénérable *Tea Master* , Franck Rondepierre (et ses bogues du compilateur), les nouveaux plus vraiment nouveaux Luk Bettale, Sonia Belaïd (déja ancienne) et Rina Zeitoun (pas encore vraiment là). Je n'oublie pas Thomas Roche avec qui j'ai toujours eu d'intéressantes discussions (les autres aussi je vous rassure) ainsi que Coraline Streiff.

Un peu plus ensoleillés, les Bordelais ne sont pas en reste avec Christophe Giraud (laser smashed card), Soline Renner avec qui j'ai découvert l'île d'Oléron, Philippe Andouard et nos 24 heures d'avion, sans oublier Hugues Thiébeauld, Nicolas Morin, Alberto Battistello, Guillaume Barbu et tous les autres.

Je tiens particulièrement à citer la petite équipe *spectrale* des Critères Communs, Clement Capel, Fabien Deboyser et Sarra Mestiri, que de longues discussions pour meubler entre deux pauses, qu'aurais-je fais

sans vous ?

Au cours de cette thèse j'ai aussi été amené à rencontrer d'autres personnes au cours de séminaires, trop nombreuses pour toutes les citées, je vous remercie pour les discussions partagées.
Je tiens aussi à citer Ange Martinelli et Nabil Hamzi, ainsi que tout ceux croisé avant, et qui sont toujours là !

Enfin je finirai par le cercle familiale qui m'a supporter durant cette longue période. Merci à mes parents, à mon frère, et surtout merci à toi Camille.

mon correcteur d'orthographe est en panne ;-)

D-2 Introduction

Cette thèse dans le domaine de la cryptologie s'intéresse aux attaques par canaux auxiliaires. Traditionnellement, les preuves de sécurité en cryptologie se placent dans un modèle dit en *boite noire*, qui suppose que l'adversaire connaît l'algorithme utilisé et n'a accès qu'à un oracle paramétré par un secret lui fournissant les résultats (chiffrement ou déchiffrement) de ses requêtes. Dans ce modèle, il est possible de montrer que, pour certains algorithmes, un adversaire appliquant une stratégie optimale ne peut retrouver le secret de l'oracle plus rapidement qu'une recherche exhaustive. Cependant, le modèle en boite noire ne permet pas de prouver la sécurité d'un système en pratique.

En pratique, en effet, l'adversaire peut avoir physiquement un accès à l'oracle (c'est notamment le cas pour les cartes à puces, largement utilisées dans le monde bancaire et de la téléphonie mobile comme outils de sécurité). Dans ce contexte, il peut observer (voire même perturber) les calculs faits par l'oracle et mesurer l'impact de ce calcul sur son environnement (par exemple la consommation du courant, le temps de calcul, *etc.*). Ces observations sont généralement liées aux valeurs des résultats intermédiaires manipulées par l'oracle et donnent ainsi de l'information supplémentaire à l'adversaire, lui permettant de retrouver plus efficacement le secret. Ce modèle où un adversaire a accès à des valeurs intermédiaires est appelé modèle en *boite grise*.

L'utilisation d'informations physiques afin de *casser* un crypto-système ainsi que l'étude des contre-mesures associées font partie du domaine de l'*analyse par canaux auxiliaires* dans lequel s'inscrit cette thèse.

Dans un premier temps, une attention particulière est portée aux attaques par canaux auxiliaires existantes et un cadre d'étude général est proposé pour permettre leur étude comparée.

D-3 Étude comparée des attaques par canaux auxiliaires

D-3.1 Un cadre général

Les attaques par canaux auxiliaires peuvent être classifiées selon trois paramètres.
Le premier paramètre est la puissance de l'attaquant qui permet de définir deux classes. Dans la première classe, l'attaquant est supposé être très puissant et contrôle une copie identique du dispositif attaqué (attaque par apprentissage). Sous cette hypothèse, il peut insérer n'importe quel secret dans la copie, faire les observations correspondantes et se constituer un dictionnaire. Dans la seconde classe (attaque simple), l'attaquant est supposé avoir un accès limité au dispositif attaqué et n'a plus la capacité de faire un dictionnaire, même partiel.
Un second paramètre est le type d'opération ciblée. L'attaque peut, en effet, par exemple, cibler le flot d'opérations, ce qui consiste essentiellement à analyser directement les observations afin d'y lire la valeur manipulée. D'autres types d'attaques peuvent consister à cibler la manipulation d'une donnée. Dans ce cas, une valeur intermédiaire dépendante du secret et d'une valeur connue est ciblée. Dans ce cas l'attaquant peut avoir besoin de plusieurs observations avec différentes valeurs connues afin d'appliquer un traitement statistique pour retrouver le secret.
Le troisième et dernier paramètre est l'arité de l'attaque. Une attaque est dite univariée si elle cible un seul instant de fuite et multivariée si elle cible plusieurs instants de fuite.

Une attaque par canaux auxiliaires est principalement composée de 6 étapes : (1) mesure des observations physiques (fuites) dépendantes de l'entrée (connue) et du secret (non connu), (2) choix d'une modélisation des fuites, (3) choix d'une hypothèse sur une partie du secret et modélisation de la possible fuite, (4) choix d'un outil statistique pour valider l'hypothèse, (5) comparaison, à l'aide de l'outil statistique choisi, de la fuite modélisée pour chaque hypothèse avec la fuite réellement mesurée, (6) choix de l'hypothèse qui produit le modèle le plus proche des mesures. Dans cette description générale, il peut être observé que les étapes (2) et (4) relèvent d'un choix empirique par l'attaquant. L'un des buts de cette thèse est de donner un cadre formel permettant de guider l'attaquant dans cette étape. En particulier, une classification précise

des attaques existantes est établie, mettant en avant l'importance du choix de la modélisation (étape (2)). De plus, un lien entre différentes attaques de la littérature est établi et une nouvelle attaque générique plus performante que celles existantes est exhibée.

Dans un premier temps nous nous sommes intéressés aux attaques par canaux auxiliaire simples et univariées ciblant une unique valeur intermédiaire. Dans ce cas, la fuite, notée L est modélisée de la manière suivante :

$$L = \delta(Z) + B \ , \tag{D.1}$$

où δ correspond à la partie déterministe de la fuite, où B est un bruit gaussien et où Z correspond à la donnée manipulée. Nous supposons que cette donnée résulte de l'application d'une fonction F connue à une variable connue X et un secret k. La variable $Z = F_k(X)$ est dite *sensible* car elle dépend et d'un secret et d'une valeur connue.

D-3.2 Principales attaques et classification

De nombreuses attaques basées sur une analyse statistique des données existent dans la littérature, notamment la *Differential Power Analysis (DPA)* introduite par Kocher *et al.* [52]. Le principe de cette dernière est de partitionner les fuites en fonction d'une hypothèse sur un bit de la valeur intermédiaire ciblée et de soustraire la moyenne des deux partitions ainsi crées. Cette attaque a été étendue à plusieurs bits par Messerges dans [67] de deux façons : la *all-or-nothing DPA*, qui partitionne suivant 2 valeurs (au lieu de 2 bits) et la *generalized DPA* qui partitionne les fuites suivant 2 sous-ensembles de valeurs.

Après le travail de Messerges, d'autres extensions ont été proposées. Notamment, la *Partition Power Analysis (PPA)* par Le *et al.* dans [56] qui, au lieu de partitionner les fuites en 2 ensembles, partitionne les fuites suivant chaque valeur de la variable intermédiaire ciblée puis fait une somme pondérée de la moyenne de chacune des partitions. Une autre extension des travaux de Messerges a été proposée sous le terme *Variance Power Analysis (VPA)* par Standaert *et al.* dans [97] et Magrhebi *et al.* dans [60]. Son principe est assez proche de celui de la PPA : une somme pondérée est calculée, non plus entre des moyennes, mais entre des variances, chacune correspondant à une partition. Enfin, en 2004, la *Correlation Power Analysis (CPA)* a été proposée par Brier *et al.* dans [26],

où un coefficient de corrélation linéaire entre les fuites et les hypothèses est calculée.
Tout comme les travaux originaux de Messerges, les extensions que nous avons rappelées, sont basées sur un critère de dépendance linéaire entre la fuite mesurée et une hypothèse. Un autre type d'attaques basé sur l'information mutuelle a été aussi proposé par Gierlich *et al.* [42]. Le principe est de tenter de mesurer l'information mutuelle entre la fuite et une hypothèse sur la fuite. Cette attaque permet théoriquement de mesurer toute dépendance même non linéaire. Cependant, les différentes expériences rapportées dans la littérature montrent, qu'en pratique, ces attaques sont moins efficaces que celles basées sur le coefficient de Pearson.

Toutes les attaques citées précédemment ont été appliquées dans de nombreux papiers (*e.g.* [35, 44, 67].) et parfois comparées empiriquement entre elles [55, 65, 97]. Néanmoins aucun de ces travaux ne permet de tirer une conclusion définitive sur leurs similarités et différences. Cette thèse a pour but de combler ce manque, d'une part en établissant une réduction des différentes attaques à la CPA, et d'autre part en établissant une classification formelle des attaques qui met en avant l'importance de la modélisation de la fuite. Cela à conduit à l'exhibition d'une nouvelle attaque générique plus performante que celles existantes.

D-4 Une nouvelle attaque générique

D-4.1 Description

Dans une seconde partie de cette thèse, nous avons analysé en détails une nouvelle attaque qui a pour particularité principale d'élargir le choix de la modélisation à l'étape (2). D'un point de vue technique, l'attaque, basée sur le principe de la régression linéaire, permet de tester tout les modèles qui appartiennent à une certaine catégorie obtenue en faisant des hypothèses réalistes sur le fonctionnement du matériel pendant les calculs. Plus concrètement, une hypothèse est faite sur le degré algébrique de la partie déterministe des fuites et l'attaque cherche le modèle (vu comme une fonction) qui se rapproche le plus de cette partie déterministe.

Pour mener la nouvelle, attaque proposée, l'attaquant doit choisir une

base de fonctions puis appliquer une régression linéaire qui va calculer la fonction qui appartient à l'espace engendré par la base (*i.e.* qui est une combinaison linéaire des éléments de la base) qui est la plus proche (au sens de la distance euclidienne) des observations. Ce calcul repose sur un calcul des moindres carrés. En effet, la régression recherche les coefficients de la combinaison linéaire des éléments de la base qui minimise la distance avec les observations. Cette distance étant une forme quadratique avec une matrice Hessienne définie positive, la minimiser revient à résoudre le système qui annule les dérivés partielles en chaque coefficients, ce qui se fait facilement si le système est déterminé (ce qui est généralement le cas).

La principale difficulté de cette nouvelle attaque réside dans le choix de la base. L'analyse de cette difficulté fait l'objet de la section suivante.

D-4.2 Choix de la base

Comme remarqué précédemment, le choix de la base pour la régression linéaire est un choix important. En effet, idéalement la base doit garantir que la fonction calculée sous la bonne hypothèse de clé k est « plus proche » des observations que les fonctions calculées pour des mauvaises hypothèses de clé $\hat{k}$. Si nous notons $\mathscr{H}$ l'ensemble des fonctions générées par la base choisie, l'étape de régression minimise la distance entre la fonction $\delta \circ F_k \circ F_{\hat{k}}^{-1}$ et l'espace $\mathscr{H}$. Ainsi lorsque la bonne hypothèse est faite, l'attaque approxime directement la fonction δ alors que sinon elle approxime une fonction de la forme $\delta \circ F_k \circ F_{\hat{k}}^{-1}$, qui suivant la nature de F, est plus ou moins « éloignée »de δ. Idéalement, il faut donc choisir une base (et donc un ensemble engendré $\mathscr{H}$) telle que δ appartient à $\mathscr{H}$ mais pas $\delta \circ F_k \circ F_{\hat{k}}^{-1}$ quand $\hat{k} \neq k$. Si nous notons $\mathscr{J}$ l'ensemble des fonctions $\{F_k \circ F_{\hat{k}}\ ;\ k \neq \hat{k}\}$, une bonne stratégie consiste en le choix d'une base qui contient δ et pour laquelle la distance entre $\mathscr{H}$ et $\mathscr{H} \circ \mathscr{J}$ est maximale. Cela est résumé dans la figure D-4-1.
Une conséquence importante de cette stratégie est que choisir la base la plus large (et à l'extrême choisir la base qui engendre l'espace complet $\mathscr{F}$ des fonctions) est inefficace. En effet dans ce cas, aussi bien δ que $\delta \circ F_k \circ F_{\hat{k}}^{-1}$ pour $\hat{k} \neq k$ risque d'appartenir à $\mathscr{H}$. En pratique, on observe que δ est de faible degré algébrique (typiquement une combinaison linéaire des bits de la variable manipulée) et que $F_k \circ F_{\hat{k}}^{-1}$ pour $k \neq \hat{k}$ est une fonction de haut degré algébrique (c'est typiquement le cas lorsque F est

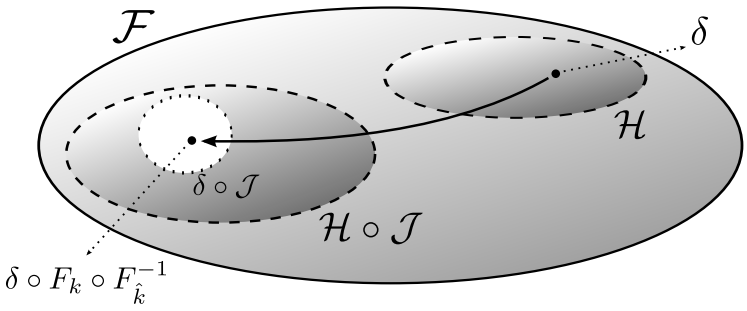


FIGURE D-4-1 – Relations entre les différents espaces.

une *boîte-S* d'un chiffrement par blocs). Ainsi choisir une base qui génère les fonctions de petits degrés algébriques est une bonne stratégie.

Remarque : Bien qu'utiliser une base générant tout l'espace soit inefficace lorsqu'on utilise la distance euclidienne, il est possible d'appliquer une étape supplémentaire aux approximations fournies par la régression dans le but de « reconnaître » celle correspondant à δ de celles correspondant à $\delta \circ F_k \circ F_{\hat{k}}^{-1}$ pour $k \neq \hat{k}$. Par exemple dans le cas d'une fonction δ de petit degré algébrique et d'une fonction F de type boîte-S, la fonction δ aura donc tous ses coefficients de degré élevé nuls contrairement aux autres fonctions.

De nombreuses expériences ont été faites afin de valider le lien entre les différentes attaques connues et la pertinence de cette nouvelle attaque et ainsi que des différentes stratégies pour le choix de la base.

D-4.3 Optimisations

Deux optimisations algorithmiques de l'attaque par régression linéaire sont aussi proposées et analysées dans cette thèse.

La première optimisation concerne un pré-traitement des fuites suivant la valeur connue qui leur est associée. En effet, la régression linéaire étant basée sur la résolution d'un système linéaire, elle utilise des opérations matricielles qui ont une complexité quadratique en le nombres d'observations. De ce fait, lorsque le nombre d'observations est grand, la complexité de l'attaque devient critique. Une solution pour remédier à ce problème est de préalablement regrouper (*i.e.* moyenner) les fuites suivant la valeur de la donnée connue. Nous avons montré que cela ne modifie pas l'efficacité de l'attaque. En revanche sa complexité devient indé-

pendante du nombre d'observations ce qui permet de considérablement réduire les temps de calcul lorsque l'attaque nécessite de nombreuses observations.

Remarque : Cette optimisation n'est pas dédiée à la régression linéaire mais peux être appliquée à toute attaque basée sur un calcul de moments conditionnels (par exemple la CPA).

La seconde optimisation proposée dans cette thèse consiste en l'utilisation d'un algorithme itératif afin de calculer la fonction qui minimise la distance euclidienne avec les observations. Le principe repose sur le fait que si chaque élément de la base est linéairement indépendant, alors il est possible de faire la régression linéaire élément par élément. Ainsi cet algorithme recherche l'élément de la base le plus corrélé avec les observations, fait une régression linéaire suivant cet élément, puis retranche le résultat des observations puis recommence (pour plus de détails techniques,voir [38]). Cet algorithme introduisant les éléments de la base du plus corrélé au moins corrélé, il permet de réduire la taille de la base (et donc l'une des dimensions de la matrice dans la régression) en ne gardant que les éléments pertinents pour l'attaque. Cela permet en outre, quand le nombre d'observations est trop faible, de ne pas introduire tous les éléments de la base. De plus, cela permet de choisir à chaque fois les éléments les plus pertinents de la base. Dans ce cas de figure, une bonne stratégie est de donner une base très large et de laisser l'algorithme choisir un petit nombre d'éléments dans cette base. Cela apporte ainsi une plus grande flexibilité sur le choix de la base de régression linéaire et une attaque plus performante.

Ces deux optimisations proposées dans cette thèse peuvent bien évidemment être combinées afin d'atteindre une attaque performante et rapide.

D-5 Attaques d'ordre supérieur

Pour contrer les attaques univariées, des contre-mesures efficaces ont été mises en place (cf. section suivante sur les contre-mesures associées). Dans ce contexte les attaques doivent souvent cibler deux instants afin d'avoir une chance de retrouver de l'information sur le secret. Dans une troisième partie, nous nous sommes donc intéressés au coté multivarié

des attaques par régression linéaire afin de les appliquer à des implantations protégées.

Après avoir étudié les attaques par canaux auxiliaires simples ciblant une valeur intermédiaire dans le monde univarié, nous nous sommes intéressés au monde multivarié. En particulier, nous nous sommes concentrés sur les attaques bivariées (*a.k.a.* du second-ordre). En général, les différents instants sont combinés entre eux par produit (centré) afin d'appliquer des attaque univariées parmi celles rappelées dans la section D-3.2. En effet, seules quelques attaques comme la MIA sont intrinsèquement multivariées.
Dans un premier temps, nous avons montré que l'attaque par régression linéaire présentée dans la section D-4 pouvait aussi s'appliquer dans le cas d'observations bivariées. Pour se faire, la régression linéaire étant une attaque univariée, nous avons montré qu'une fonction de combinaison devait être appliquée. Nous avons choisi le produit centré et nous avons étudié la pertinence théorique de ce choix. Nous avons ensuite montré d'une part, qu'elle englobait les précédentes attaques basées sur la CPA, et d'autre part, qu'il existait un lien entre celle-ci et une attaque par maximum de vraisemblance (attaque optimale en théorie).

Remarque : Dans le cas bivarié, et la fonction de combinaison et la contre-mesure employée ont un impact sur la fonction approximée par la régression linéaire. Par conséquent, utiliser une base complète et discriminer directement avec la forme algébrique (comme remarqué précédemment) nécéssite de prendre en compte ces deux facteurs supplémentaires. Plus intéressant, la forme algébrique peut aussi servir à faire de la rétro-ingénierie et permettre de retrouver la contre-mesure appliquée si elle n'était pas connue.

De nombreuses expériences ont été menées afin de valider cette approche.

D-6 Contre-mesures associées

Dans une quatrième partie de cette thèse, nous nous sommes intéressés aux deux principales contre-mesures existantes, à savoir la manipulation temporelle aléatoire des données et le partage aléatoire des données, ainsi qu'aux techniques pour combiner ensemble ces deux contre-mesures.

D-6.1 Description

La manipulation temporelle aléatoire des données consiste à rendre aléatoire sur une certaine durée t l'instant auquel une certaine donnée sensible est manipulée. Bien que cette technique ne rende pas inefficace une attaque par canaux auxiliaires, elle en diminue l'efficacité (environ t fois plus de mesures seront nécessaires afin d'obtenir un même taux de succès qu'en l'absence de contre-mesures). Pour contourner une telle contre-mesure, un attaquant doit cibler les t instants en même temps (par exemple en les sommant, le résultat contenant obligatoirement la valeur de la donnée sensible). Cette contre-mesure est assez simple à mettre en place mais nécessite de donner une grande valeur à t pour être efficace.

La seconde grande famille de contre-mesures, le partage aléatoire de données, quant à lui consiste à rendre tout uplet de d instants de mesure (d étant un paramètre de la contre-mesure) indépendant de la donnée sensible. Pour se faire, un nombre d de valeurs aléatoires sont générés à chaque exécution de l'algorithme. Ces valeurs aléatoires (appelées généralement masques) sont ensuite combinées avec la donnée sensible. De cette façon la donnée manipulée correspondante à la donnée sensible combinée aux aléas ne dépend plus du secret. Une attaque par canaux auxiliaires ne ciblant que d instants de la fuite (ou moins) est alors inefficace.

Cette contre-mesure bien que très efficace (le nombre de message nécessaire pour obtenir un même taux de succès qu'en l'absence de contre-mesures augmente exponentiellement avec d) est très coûteuse à implémenter lorsque les opérations effectuées sur la donnée sensible ne sont pas linéaires.

D-6.2 Combinaisons

Des travaux ont montré qu'une stratégie efficace de sécurisation et d'implantation consistait à combiner le partage de données avec la manipulation temporelle aléatoire. Dans ce cas, la donnée sensible est partagé en $d+1$ parts et la manipulation de chacune des parties est rendue aléatoire sur t instants.

Pour contourner une telle combinaison de contre-mesures, une combinai-

son des attaques elles-mêmes doit être opérée. La meilleure application de cette stratégie connue consiste à ce que l'attaquant fasse une somme de toutes les $d+1$ combinaisons possibles parmi les t fuites. Bien qu'en théorie, une telle combinaison soit toujours possible, de nombreux obstacles pratiques jalonnent l'implantation d'une telle contre-mesure, notamment la quantification de la sécurité apportée. Certains schémas ont déjà été proposé [46, 103] mais se limitent à un partage d'ordre 1.

Dans cette thèse nous avons proposé des nouveaux schémas combinant la manipulation aléatoire des données et le partage de données dans le but de bénéficier des avantages des deux types de contre-mesures tout en limitant les défauts. Nous avons par ailleurs proposé un nouveau cadre permettant de quantifier la sécurité apportée par de telles techniques en fonction des paramètres t et d.

D-7 Conclusion

En conclusion, les apports principaux de cette thèse sont : une unification des différentes attaques par canaux auxiliaires existantes, une introduction de nouvelles techniques d'attaques plus robustes aux erreurs lors des étapes de modélisation et un nouveau schéma de protection contre toutes ces attaques ainsi qu'un schéma d'évaluation associé..

Publications

[DDP12] Guillaume Dabosville, Julien Doget, and Emmanuel Prouff.
A New Second Order Side Channel Attack Based on Linear Regression.
IEEE Trans. Comput., 2012.
To appear.
A prior version is available on IACR eprint at `http://eprint.iacr.org/2011/505.pdf`.

[DPRS11] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert.
Univariate Side Channel Attacks and Leakage Modeling.
Journal of Cryptographic Engineering, 1(2):123–144, 2011.
A short version was presented at COSADE 2011 and is available at `http://cosade2011.cased.de/files/2011/cosade2011_talk1_paper.pdf`.
A prior version is available on IACR eprint at `http://eprint.iacr.org/2011/302.pdf`.

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget.
Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers.
In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

An extended version is available on IACR eprint at `http://eprint.iacr.org/2009/420.pdf`.

Bibliography

[1] FIPS PUB 197. *Advanced Encryption Standard*. National Institute of Standards and Technology, November 2001.

[2] FIPS PUB 46. *The Data Encryption Standard*. National Bureau of Standards, January 1977.

[3] FIPS PUB 46-3. *Data Encryption Standard (DES)*. National Institute of Standards and Technology, October 1999.

[4] ISO/IEC 7810. *Identification Cards – Physical Characteristics*, third edition, November 2003.

[5] ISO/IEC 7816-1. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 1: Physical Characteristics*, 2003.

[6] ISO/IEC 7816-10. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 10: Electronic Signals and Answer to Reset for Synchronous Cards*, 1999.

[7] ISO/IEC 7816-11. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 11: Personal Verification through Biometric Methods*, 2004.

[8] ISO/IEC 7816-12. *Identification Cards – Integrated Circuit Cards – Part 12: Cards with Contacts: USB Electrical Interface and Operating Procedures*, 2005.

[9] ISO/IEC 7816-13. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 13: Commands for Application Management in Multi-Application Environment*, 2007.

[10] ISO/IEC 7816-15. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 15: Cryptographic Information Application*, 2008.

[11] ISO/IEC 7816-2. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 2: Dimensions and Location of the Contacts*, 2007.

[12] ISO/IEC 7816-3. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 3: Electronic Signals and Transmission Protocols*, 2006.

[13] ISO/IEC 7816-4. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 4: Organization, Security and Commands for Interchange*, 2005.

[14] ISO/IEC 7816-5. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 5: Registration of Application Providers*, 2004.

[15] ISO/IEC 7816-6. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 6: Interindustry Data Elements for I]nterchange*, 2004.

[16] ISO/IEC 7816-7. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 7: Interindustry Commands for Structured Card Query Language (SCQL)*, 1999.

[17] ISO/IEC 7816-8. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 8: Commands for Security Operations*, 2004.

[18] ISO/IEC 7816-9. *Information Technology – Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 9: Commands for Card Management*, 2004.

[19] D. Agrawal, J.R. Rao, and P. Rohatgi. Multi-channel Attacks. In C.D. Walter, Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 2–16. Springer, 2003.

[20] Mehdi-Laurent Akkar and C. Giraud. An Implementation of DES and AES, Secure against Some Attacks. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.

[21] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems – CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.

[22] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F.-X. Standaert, and N. Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *to appear in the Journal of Cryptology*, 24(2):269–291, April 2011.

[23] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying Hash Functions for Message Authentication. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996.

[24] R. Bévan and E. Knudsen. Ways to Enhance Power Analysis. In P.J. Lee and C.H. Lim, editors, *Information Security and Cryptology – ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2002.

[25] John Black and Phillip Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2000.

[26] É. Brier, C. Clavier, and F. Olivier. Optimal Statistical Power Analysis. Cryptology ePrint Archive, Report 2003/152, 2003.

[27] É. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume

3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[28] Claude Carlet. Boolean functions for cryptography and error correcting codes. *Boolean Methods and Models*, page 257, 2010.

[29] Claude Carlet, Louis Goubin, Emmanuel Prouff, Michael Quisquater, and Matthieu Rivain. Higher-order masking schemes for s-boxes. In Anne Canteaut, editor, *FSE*, Lecture Notes in Computer Science. Springer, 2012.

[30] S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In M.J. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[31] S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–29. Springer, 2002.

[32] C. Clavier, J.-S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2000.

[33] J.-S. Coron, E. Prouff, and M. Rivain. Side Channel Cryptanalysis of a Higher Order Masking Scheme. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2007.

[34] Jean-Sébastien Coron. A New DPA Countermeasure Based on Permutation Tables. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *Security and Cryptography for Networks, 6th International Conference, SCN 2008*, volume 5229 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2008.

[35] Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, and Matthieu Rivain. Attack and Improvement of a Secure S-Box Calculation Based on the Fourier Transform. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2008.

[36] J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer, 2002.

[37] Chunjie Duan, Victor H. Cordero Calle, and Sunil P. Khatri. Efficient On-Chip Crosstalk Avoidance CODEC Design. *IEEE Trans. VLSI Syst.*, 17(4):551–560, 2009.

[38] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[39] B. Everitt and A. Skrondal. *The Cambridge Dictionary of Statistics*. Cambridge University Press, New York, NY, USA, 2002.

[40] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine Masking against Higher-Order Side Channel Analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 262–280. Springer, 2010.

[41] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic Analysis: Concrete Results. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[42] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.

[43] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems – CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.

[44] J. Golić and C. Tymen. Multiplicative Masking and Power Analysis of AES. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 198–212. Springer, 2002.

[45] Sylvain Guilley, Olivier Meynard, Laurent Sauvage, and Jean-Luc Danger. An Empirical Study of the EIS Assumption in Side-

Channel Attacks against Hardware Implementations. In Werner Schindler and Sorin Huss, editors, *COSADE*, pages 10–14, 2010.

[46] P. Herbst, E. Oswald, and S. Mangard. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In J. Zhou, M. Yung, and F. Bao, editors, *Applied Cryptography and Network Security – ANCS 2006*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 2006.

[47] Annelie Heuser, Michael Kasper, Werner Schindler, and Marc Stöttinger. How a Symmetry Metric Assists Side-Channel Evaluation - A Novel Model Verification Method for Power Analysis. In *DSD*, pages 674–681. IEEE, 2011.

[48] M. Joye, P. Paillier, and B. Schoenmakers. On Second-order Differential Power Analysis. In J.R. Rao and B. Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2005.

[49] D.E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, third edition, 1988.

[50] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[51] P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. Technical report, Cryptography Research Inc., 1998.

[52] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M.J. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[53] Sandeep Kumar, Christof Paar, Jan Pelzl, Gerd Pfeiffer, and Manfred Schimmler. Breaking Ciphers with COPACOBANA – A Cost-Optimized Parallel Code Breaker. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 101–118. Springer, 2006.

[54] Thanh-Ha Le and Mael Berthier. Mutual information analysis under the view of higher-order statistics. In Isao Echizen, Noboru Kunihiro, and Ryôichi Sasaki, editors, *IWSEC*, volume 6434 of *Lecture Notes in Computer Science*, pages 285–300. Springer, 2010.

[55] Thanh-Ha Le, Cécile Canovas, and Jessy Clédière. An overview of side channel analysis attacks. In Masayuki Abe and Virgil D. Gligor, editors, *ASIACCS*, pages 33–43. ACM, 2008.

[56] Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume. A Proposition for Correlation Power Analysis Enhancement. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems – CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 174–186. Springer, 2006.

[57] Kerstin Lemke-Rust. *Models and Algorithms for Physical Cryptanalysis*. PhD thesis, Ruhr-Universität-Bochum, Germany, Jan 2007.

[58] Kerstin Lemke-Rust and Christof Paar. Gaussian Mixture Models for Higher-Order Side Channel Analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 14–27. Springer, 2007.

[59] Jiye Liu, Yongbin Zhou, Shuguo Yang, and Dengguo Feng. Generic side-channel distinguisher based on kolmogorov-smirnov test: Explicit construction and practical evaluation. *IACR Cryptology ePrint Archive*, 2011:694, 2011.

[60] Houssem Maghrebi, Jean-Luc Danger, Florent Flament, Sylvain Guilley, and Laurent Sauvage. Evaluation of countermeasure implementations based on Boolean masking to thwart side-channel attacks. In *Proc. 3rd Int Signals, Circuits and Systems (SCS) Conf*, pages 1–6, 2009.

[61] Houssem Maghrebi, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. Entropy-based Power Attack. In Jim Plusquellic and Ken Mai, editors, *HOST*, pages 1–6. IEEE Computer Society, 2010.

[62] S. Mangard, E. Oswald, and F.-X. Standaert. One for All - All for One: Unifying Standard DPA Attacks. *IET Information Security*, 2011.

[63] Stefan Mangard. Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness. In T. Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[64] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks – Revealing the Secrets of Smartcards*. Springer, 2007.

[65] Stefan Mangard, Elisabeth Oswald, and Francois-Xavier Standeart. One for All - All for One: Unifying Standard DPA Attacks. Cryptology ePrint Archive, Report 2009/449, 2009. http://eprint.iacr.org/.

[66] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[67] T.S. Messerges. *Power Analysis Attacks and Countermeasures for Cryptographic Algorithms*. PhD thesis, University of Illinois, 2000.

[68] T.S. Messerges. Securing the AES Finalists against Power Analysis Attacks. In B. Schneier, editor, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 2000.

[69] T.S. Messerges. Using Second-order Power Analysis to Attack DPA Resistant Software. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.

[70] T.S. Messerges, E.A. Dabbish, and R.H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *the USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 151–161, 1999.

[71] Francesc Moll, Miquel Roca, and Eugeni Isern. Analysis of dissipation energy of switching digital CMOS gates with coupled outputs. *Microelectronics Journal*, 34(9):833–842, 2003.

[72] NSA. TEMPEST: A Signal Problem, The story of the discovery of various compromising radiations from communications and Comsec equipment. *Cryptologic Spectrum*, 2(3), 1972.

[73] Elisabeth Oswald and Stefan Mangard. Template Attacks on Masking—Resistance is Futile. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 243–256. Springer, 2007.

[74] Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical Second-order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. In D. Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2006.

[75] J. Patarin. How to Construct Pseudorandom and Super Pseudorandom Permutation from one Single Pseudorandom Function. In R.A. Rueppel, editor, *Advances in Cryptology – EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1992.

[76] J.K. Patel and C.B. Read. *Handbook of the Normal Distribution*. Statistics, textbooks and monographs. Marcel Dekker, 1996.

[77] J. Pieprzyk. How to Construct Pseudorandom Permutations from Single Pseudorandom Functions Advances. In I.B. Damgård, editor, *Advances in Cryptology – EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 140–150. Springer, 1990.

[78] Emmanuel Prouff. DPA attacks and S-Boxes. In H. Handschuh and H. Gilbert, editors, *Fast Software Encryption – FSE 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 424–442. Springer, 2005.

[79] Emmanuel Prouff and Matthieu Rivain. Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis (Extended Version). To appear in the Int. Journal of Applied Cryptography (IJACT), 2010.

[80] Emmanuel Prouff, Matthieu Rivain, and Régis Bévan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Comput.*, 58(6):799–811, 2009.

[81] Jean-Jacques Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA): Measures and Countermeasures for Smart Cards. In I. Attali and T. Jensen, editors, *Smart Card Programming and Security – E-smart 2001*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[82] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In Kenneth G. Paterson, editor, *EUROCRYPT*, vol-

ume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.

[83] Matthieu Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In Roberto Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography – SAC 2008*, Lecture Notes in Computer Science. Springer, 2008.

[84] Matthieu Rivain, Emmanuelle Dottax, and Emmanuel Prouff. Block Ciphers Implementations Provably Secure Against Second Order Side Channel Analysis. In T. Baignères and S. Vaudenay, editors, *Fast Software Encryption – FSE 2008*, Lecture Notes in Computer Science, pages 127–143. Springer, 2008.

[85] Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.

[86] Andrew R. Runnalls. Kullback-Leibler Approach to Gaussian Mixture Reduction. *IEEE Transactions of Aerospace and Electronic Systems*, 43(3):989–999, July 2007.

[87] Werner Schindler. Advanced stochastic methods in side channel analysis on block ciphers in the presence of masking. *Journal of Mathematical Cryptology*, 2:291–310, 2008.

[88] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In J.R. Rao and B. Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*. Springer, 2005.

[89] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, Inc, 2nd edition, 1996.

[90] Kai Schramm and Christof Paar. Higher Order Masking of the AES. In D. Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2006.

[91] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*. Wiley-Interscience, September 1992.

[92] C. E. Shannon. Communication theory of secrecy systems. *Bell System Tech. J.*, 28:656–715, 1949.

[93] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[94] S. Singh. *Histoire des codes secrets – De l'Égypte des pharaons à l'ordinateur quantique*. LC Lattès, 1999.

[95] F.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater. An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays. *IEEE*, 94(2):383–394, 2006.

[96] François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.

[97] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology – ICISC 2008*, volume 5461 of *Lecture Notes in Computer Science*, pages 253–267. Springer, 2008.

[98] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[99] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[100] D.R. Stinson. *Cryptography – Theory and Practice*. CRC Press, 1995.

[101] Taiji Suzuki, Masashi Sugiyama, Takafumi Kanamori, and Jun Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(S-1), 2009.

[102] Stefan Tillich and Christoph Herbst. Attacking State-of-the-Art Software Countermeasures-A Case Study for AES. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 228–243. Springer, 2008.

[103] Stefan Tillich, Christoph Herbst, and Stefan Mangard. Protecting AES Software Implementations on 32-Bit Processors Against Power Analysis. In Jonathan Katz and Moti Yung, editors, *ACNS*, volume 4521 of *Lecture Notes in Computer Science*, pages 141–157. Springer, 2007.

[104] Berwin A. Turlach. Bandwidth Selection in Kernel Density Estimation: A Review. In *CORE and Institut de Statistique*, pages 23–493, 1993.

[105] Wim van Eck. Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computer & Security*, 4:269–286, 1985.

[106] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual Information Analysis: How, When and Why? In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 429–443. Springer, 2009.

[107] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Adaptive Chosen-Message Side-Channel Attacks. In Jianying Zhou and Moti Yung, editors, *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 186–199, 2010.

[108] J. Waddle and D. Wagner. Toward Efficient Second-order Power Analysis. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2004.

[109] Janett Walters-Williams and Yan Li. Estimation of mutual information: A survey. In Peng Wen, Yuefeng Li, Lech Polkowski, Yiyu Yao, Shusaku Tsumoto, and Guoyin Wang, editors, *RSKT*, volume 5589 of *Lecture Notes in Computer Science*, pages 389–396. Springer, 2009.

[110] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51:59–64, 1997.

[111] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics, 2005.

[112] Carolyn Whitnall and Elisabeth Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2011.

[113] Carolyn Whitnall, Elisabeth Oswald, and Luke Mather. An Exploration of the Kolmogorov-Smirnov Test as Competitor to Mutual Information Analysis. In Vincent Rijmen and Emmanuel Prouff, editors, *CARDIS*, Lecture Notes in Computer Science, pages 316–334. Springer, 2011.

[114] P. Wright. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer*. William Heinemann Australia, 1987.

RÉSUMÉ COURT :

Cette thèse s'intéresse aux attaques par canaux auxiliaires contre les implantations matérielles d'algorithmes cryptographiques. Les études conduites dans ce document se placent donc dans le cadre où un adversaire a accès à des observations bruitées des résultats intermédiaires d'un calcul cryptographique. Dans ce contexte, de nombreuses attaques existent avec leurs contremesures dédiées, mais leur pertinence et leur mise en pratique restent encore floues.

Cette thèse s'intéresse dans un premier temps à la pertinence des attaques existantes et aux possibles liens qui les unissent. Une classification formelle est proposée ainsi que des critères de choix. Sur la base de cette étude, une attaque générique performante est décrite et analysée en profondeur.

Dans un second temps, la mise en pratique des contremesures actuelles est étudiée, donnant lieu à la création d'un schéma d'application les mélangeant pour atteindre de meilleurs compromis efficacité/sécurité.

MOTS-CLÉS :

• Systèmes enfouis (informatique) • Cryptographie • Analyse de régression • Analyse stochastique • Corrélation

TITLE:

Side-Channel Analysis and Countermeasures

BRIEF SUMMARY:

This thesis deals with side channel attacks against hardware implementations of cryptographic algorithms. Studies conducted in this document are therefore in place where an adversary has access to noisy observations of intermediate results of a cryptographic computation. In this context, many attacks are dedicated with their countermeasures, but their relevance and their implementation are still unclear.

This thesis initially focuses on the relevance of existing attacks and potential links between them. A formal classification is proposed as well as selection criteria. Based on this study, a generic efficient attack is described and analysed in depth.

In a second step, the implementation of common countermeasures is studied, leading to the creation of an application scheme mixing them to achieve a better efficiency / security trade off.

KEYWORDS:

• Embedded systems (computer science) • Cryptography • Regression analysis • Stochastic analysis • Correlation (statistic)