Towards a better formalisation of the side-channel threat

Eloi De Cherisey

université
PARIS-SACLAY

TELECOM ParisTech

NNT: 2018SACLT0016

Vers une meilleure formalisation des attaques par canaux cachés

Thèse de doctorat de l'Université Paris-Saclay
préparée à Télécom ParisTech
(Institut Mines-Télécom)

École doctorale n° 580
Spécialité de doctorat: Information et communication

Thèse présentée et soutenue a Télécom ParisTech, le 18 décembre 2018, par

Éloi de Chérisey

Composition du jury :

Anne Canteaut DR — INRIA	Présidente du jury
Damien Vergnaud Professeur — Paris VI	Rapporteur
Emmanuel Prouff HDR — Chef de laboratoire à l'ANSSI	Rapporteur
Philippe Maurine HDR — LIRMM	Examinateur
Jean-Luc Danger Professeur — Télécom ParisTech	Examinateur
Pablo Piantanida Professeur à Centrale-Supélec — L2S	Invité
Sylvain Guilley HDR — Professeur invité à Télécom ParisTech	Directeur de thèse
Olivier Rioul HDR — Professeur à TélécomParisTech	Directeur de thèse

Thèse de doctorat

Abstract

In the field of the security of the embedded systems, it is necessary to know and understand the possible physical attacks that could break the security of cryptographic components. Since the current algorithms such as Advanced Encryption Standard (AES) are very resilient against differential and linear cryptanalysis, other methods are used to recover the secrets of these components. Indeed, the secret key used to encrypt data leaks during the computation of the algorithm, and it is possible to measure this leakage and exploit it. This technique to recover the secret key is called side-channel analysis.

The main target of this Ph. D. manuscript is to increase and consolidate the knowledge on the side-channel threat. To do so, we apply some information theoretic results to side-channel analysis. The main objective is show how a side-channel leaking model can be seen as a communication channel.

We first show that the security of a chip is dependant to the signal-to-noise ratio (SNR) of the leakage. This result is very useful since it is a generic result independent from the attack. When a designer builds a chip, he might not be able to know in advance how his embedded system will be attacked, maybe several years later. The tools that we provide in this manuscript will help designers to estimated the level of liability of their chips.

Résumé

Dans le cadre de la sécurité des systèmes embarqués, il est nécessaire de connaître les attaques logicielles et physiques pouvant briser la sécurité de composants cryptographiques garantissant l'intégrité, la fiabilité et la confidentialité des données. Étant donné que les algorithmes utilisés aujourd'hui comme *Advanced Encryption Standard* (AES) sont considérés comme résistants contre la cryptanalyse linéaire et différentielle, d'autres méthodes plus insidieuses sont utilisés pour récupérer les secrets de ces composants. En effet, la clé secrète utilisée pour le chiffrement de données peut fuiter pendant l'algorithme. Il est ainsi possible de mesurer cette fuite et de l'exploiter. Cette technique est appelée *attaque par canal auxiliaire.*

Le principal objectif de ce manuscrit de thèse est de consolider les connaissances théoriques sur ce type de menace. Pour cela, nous appliquons des résultats de théorie de l'information à l'étude par canal auxiliaire. Nous montrons ainsi comment il est possible de comparer un modèle de fuite à un modèle de transmission de l'information.

Dans un premier temps, nous montrons que la sécurité d'un composant est fortement dépendante du rapport signal à bruit de la fuite. Ce résultat a un impact fort car il ne dépend pas de l'attaque choisie. Lorsqu'un designer équipe son produit, il ne connaît pas encore la manière dont son système embarqué pourra être attaqué plusieurs années plus tard. Les outils mathématiques proposés dans ce manuscrit pourront aider les concepteurs à estimer le niveau de fiabilité de leurs puces électroniques.

Remerciements

En tout premier lieu, je voudrais remercier ma femme, Gabrielle pour le soutien indéfectible dont elle a fait part pendant ces trois années longues et fastidieuses. Elle a su être attentive et patiente surtout dans les moments pendant lesquels il me semblait perdre pied.

Merci à mes directeurs de thèse, M. Sylvain Guilley et M. Olivier Rioul qui ont su m'accompagner pendant ces trois années. Merci à Sylvain pour sa créativité et son enthousiasme. Merci à Olivier pour sa rigueur et ses relectures très fines. Merci à eux deux de m'avoir fait gagner en maturité intellectuelle.

Je voudrais aussi remercier particulièrement Pablo Piantanida pour tout le temps qu'il m'a consacré dans mes recherches liées à la théorie de l'information. Il s'est très vite intéressé au domaine des attaques par canaux cachés et son aide m'a été très précieuse.

Merci également à mes deux rapporteurs de thèse MM. Emmanuel Prouff et Damien Vergnaud qui ont très gentiment accepté de lire et commenter ce présent manuscrit.

D'un point de vue plus scientifique, je voudrais remercier tous ceux qui ont créé cette belle discipline des attaques par canaux cachés, en particulier Daniel Bernstein, François-Xavier Standaert et Benedikt Gierlich dont les différents papiers m'ont initié et donné du fil à retordre. Merci à Annelie Heuser de m'avoir précédé dans cette tâche de formaliser mathématiquement les menaces. Enfin, Merci à Claude Shannon d'avoir créé cette si belle théorie de l'information qui, bien que vieille de plus de 60 ans, trouve toujours de nouvelles applications.

Merci à Chap de Midi Chapi d'avoir agrémenté mes pauses déjeuner, surtout dans les moments où mon cerveau n'était plus disponible pour faire des équations.

Merci à l'école doctorale, en particuliers Mme Florence Besnard et M. Alain Sibille pour leur gentillesse et leur patience.

Contents

List of Figures

List of Tables

Part I

Introduction and Contributions

Chapter 1

Introduction

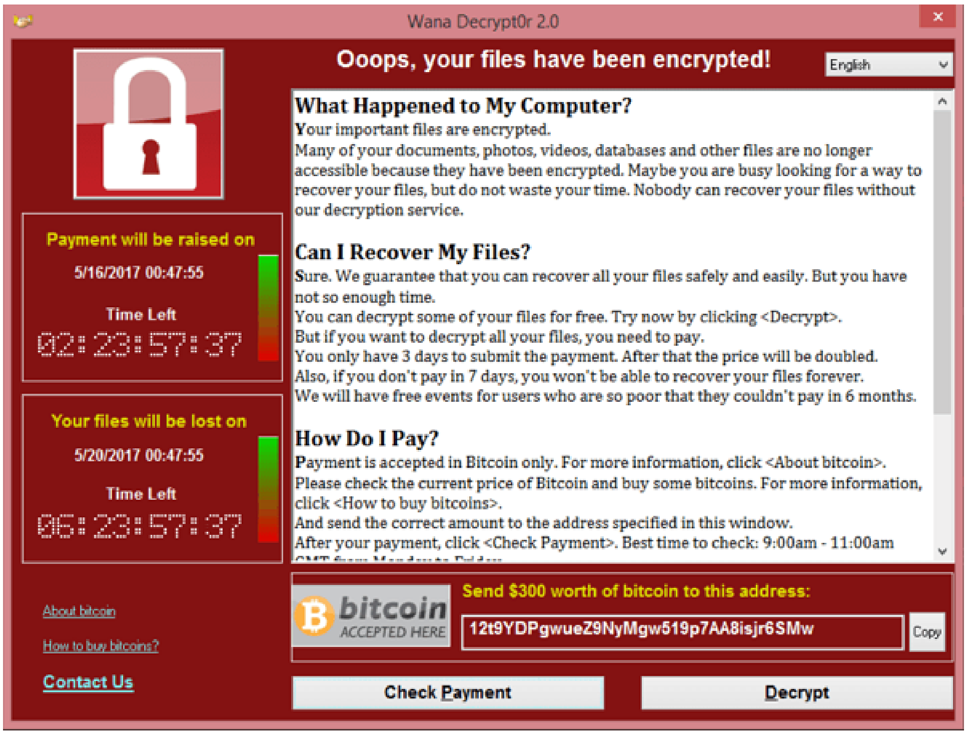
Contents

1.1 On the Importance of Cyber Security

In electronic devices such as embedded systems, personal computers, GPUs, etc, the need of security has grown wildly over the last 20 years. Indeed, millions of threats may compromise the security and therefore the private life of users, companies or state agencies, and one breach of security may alter the privacy of millions of users. For example, when ransomware *WannaCry* appeared in May 2017, about 200,000 computers were infected over 150 countries. The financial impact of this worldwide attack has been evaluated by cyber-risks modeling firm at $4 billion USD. However, experts of this domain noticed that such an attack could possibly go even worse since more sensible systems could have been affected, such as nuclear plants. This is why state agencies and big companies invest a lot of money to secure such sensible sites.

However, it is difficult to predict how a target will be attacked as the security breaches are very various and are often discovered by attackers. The attackers can have different goals such as:

- take the control of the target device;
- make the target device inoperant;
- catch the private data of the target.



What Happened to My Computer?

Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Send $300 worth of bitcoin to this address:

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

Figure 1.1: Screenshot of an infected system by WannaCry

In order to protect the systems, the designer should protect their software as well as their hardware systems. The protection of the data can be made thanks to *encryption* algorithms..

1.2 The Rise of Cryptography

As mentioned earlier, the protection of sensitive data is therefore crucial. This is why the notion of cryptography appeared.

1.2.1 A Bit of History

The notion of cryptography has appeared more than two thousands years ago. The oldest known ciphered document dates from the 16th century B.C. in Iraq. This is a clay tablet, in which a potter wrote his secret recipe. To make it secret, he removed the consonants, modifying the spelling of the words [41].

The Greeks where also using their own methods of encoding. One of them is the *scytale*. The scytale is a stick with a strip of parchment wound around it (cf Figure 1.2). The encoder and the decoder must own a stick having the same geometric characteristics to encrypt and

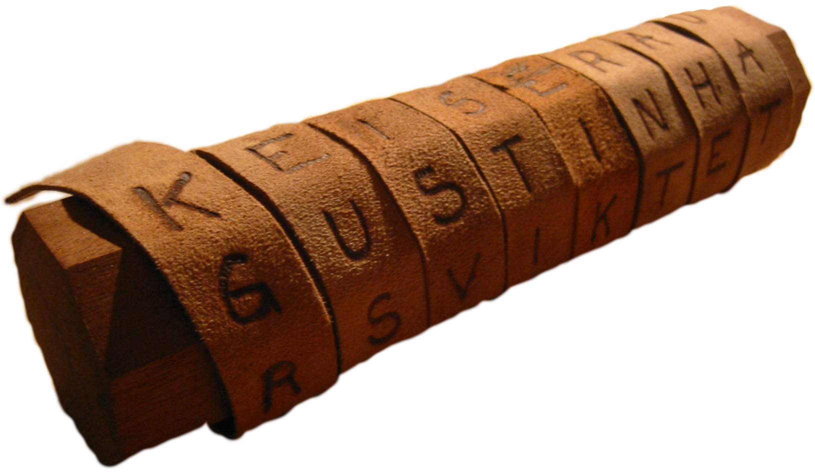


Figure 1.2: A scytale © CC BY-SA 3.0

decrypt the message. Overall, the encryption that is performed is a transpositions of letters. Today, it would be considered as a very weak code, but we must not forget that in these times, a lot of people where not even able to read and write.

During the first century B.C., Caesar ciphers appeared. They were the first letter substitution codes that appeared. In a message, every letter was changed into the n-th letter after. This method was used by the Roman army. Once again, this code is weak as one only has to check 26 possibilities to recover the message.

In 1586, French diplomat Blaise de Vigenère, published a book in which he exposed his own method of ciphering. This was called the Vigenère cipher. The method is still simple to encode and decode a message but the strength of the code is much higher than a Caesar cipher. Indeed, the code is based on a password and every letter of the message can be changed into different letter, depending on its place. The Vigenère cipher was broken in 1863 by Friedrich Kasiki.

During the Second World War, cryptography and cryptanalysis played a crucial role as Alan Turing managed to break the German encoding algorithm Enigma, supposed to be fully secured by the Wehrmacht. The impact of this was so important that today's historians agree that the war would have been at least two years longer if Enigma had not been broken [47].

1.2.2 Cryptography Today

Nowadays, codes are more sophisticated, but the idea is still the same: protect secrets from malicious threats.

For a greater security, Auguste Kerckhoffs showed that the security and the secret of a crypto-system should only be based on the secret of the encoding key [42]. This means that anyone can know the process leading to encoded data, but in this process, the encoder uses a secret key (therefore only known by him).

Eventually, a “good” coding algorithm is supposed to ensure that the best possible way to recover the secret message is to try every possibility of the key (exhaustive search). For 256 bits long keys, this would take more than 10 billions years with the best current computing power.

We can divide the codes into two big families:

- The symmetric codes where the secret key is the same for both encoding and decoding. This means that the sender and the receiver of the message must know the secret key before.

- The asymmetric codes where there is a key to encode the message (often called a *public* key) and a secret key to decode the message (often called the *private* key).

The advantage of symmetric codes is that they are often very easy to compute and to design on hardware. The speed of coding is very high (more than 10 Mb/s). The main drawback is the key exchange. How to share a secret key in a safe manner with someone?

On the contrary, asymmetric codes are much slower, but the sharing of the key is absolutely not a problem because the public key is used to encode the message and only the owner of the secret key will be able to decode the message. The only difficulty is to certify the owners of the public keys.

On the adversary’s side, the goal is to recover the secret key. Obviously, the longer the key is, the more difficult it will be for the attacker to get it. The most naive way to recover a key is the *brute force*, i.e. to check every possible key. This process is very fast when the size of the key is low (today’s machines can treat more than a billion operations per second). However, when keys are longer (for example 128 or 256 bits), it would take years to examine all the possibilities, even with the best processing units of the world. Other possibilities may exist to decrease to number of possibilities. They are called *differential* and *linear cryptanalysis*.

- In the differential cryptanalysis, the attacker is able to choose the input text. He exploits the differences at the input and sees how these differences behave at the output. A “good” algorithm transforms small differences into big ones.

Code	Year and type	Size of the key	Broken?
DES	1977 - symmetric	56 bits	Yes
RSA	1983 - asymmetric	2048 bits	No[1]
RC4	1987 - symmetric	40 to 256 bits	Yes
AES (see 1.2.3)	1999 - symmetric	128, 192 or 256 bits	No[2]

Table 1.1: List of the main encryption standards

- In linear cryptanalysis, the attacker approximates the algorithm as a linear function, and by carefully choosing the plaintext, the secret key can be recovered if the algorithm presents some linearities.

A small list of the existing encryption methods is given Table 1.1. This list is not exhaustive since there are many ways to encrypt data but is shows the main standards that are used or have been used.

1.2.3 AES

Currently, the most used algorithm is called AES [26] (for *Advanced Encryption Standards*). AES was invented by Joan Daemen and Vincent Rijmen in 1997. In 2001, the National Institute of Standards and Technologies (NIST) chooses AES as the main encoding standard. Since then, AES has acquired a very good reputation of being a very secure algorithm. It has been built so that differential and linear cryptanalysis have no effect on its security. The only known attack on a full AES algorithm has been published in 2011 by a team working for Microsoft [10]. However, this attack recovers the key in 2^{126} operations, while exhaustive search takes 2^{128} operations. This attack is therefore four times faster than the exhaustive search, but it is still a very long. This is why AES is still considered as safe.

AES is a block encoding algorithm. This means that it cuts the message to be encoded into blocks of 16 bytes. Then each block is encoded separately with the same secret key. The length of the secret key is either 16, 24 or 32 bytes. From the secret key, the algorithm first generates subkeys, as many as the number of rounds of the algorithm. For the first round, the secret key

[1] The security of RSA is based on the difficulty to reduce very big numbers into a product of prime numbers. The algorithm is simple and can be used for any key size. Nowadays, it is considered as safe to choose keys that are at least 2048 bits long. However, it has been proved that RSA will be very vulnerable to quantum cryptanalysis [80].

[2] According to the NSA, AES can be considered as secure. It is however advised to encrypt very sensitive data with keys that are at least 192 bits long.

meets the plaintext block through an exclusive or function. Then, for each round, the block passes through linear and non-linear functions such as:

1. SubBytes, a non-linear one-to-one function;
2. ShifRows, a cyclical shift of the rows of the block;
3. MixColumns, an invertible linear transformation.

Then the subkey corresponding to the number of the rounds is added.

1.3 Side-Channel Analysis

As we mentioned earlier, without more information than the message to be encoded and its encoded version, there is no better way to recover the key than the exhaustive search. To break the security of a device without performing an exhaustive search, one has therefore to use other type of information than only the plaintext and the ciphertext. These are the *side* information. They can be of any type:

- the computation time of the algorithm;
- the electro-magnetic radiations of the device during the algorithm;
- the power consumption of the device;
- the insertion of faults during the computation of the algorithm.

These methods can be classified into two types of attacks: the invasive and non-invasive attacks. Invasive attacks may alterate the targeted device while non-invasive attacks are more passive. They only listen the behaviour of the system and try to establish a model of the leakage.

According to François-Xavier Standaert [82], we can model a side-channel attack by the framework designed Figure 3.1.

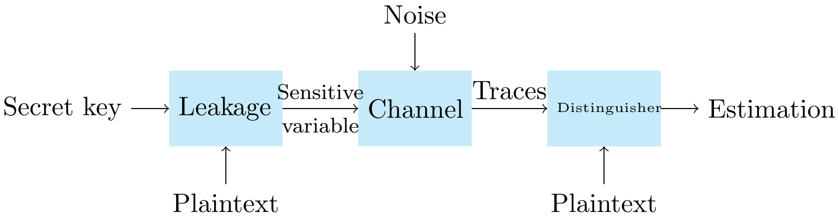


Figure 1.3: Representation of a side-channel attack

During the encryption algorithm, the plaintext that is to be encrypted meets the secret key via an *exclusive or* function (or *xor*). As the secret key is used in the algorithm, a possible leakage may happen since it is possible to measure an image of the secret key.

For example, in AES, the secret key is used during the first round of the algorithm, and in some devices, it is possible to recover the secret key when the substitution box of the secret key *xor* the plaintext is stored in the register of the target device [51].

1.3.1 Vulnerabilities

The main advantage of side-channel analysis compared to classical cryptanalysis, is that it is possible to recover each of the byte of the secret key separately. For attacker, this is a great improvement because, they no more supposed to consider at least 2^{128} possibilities but 16 times $2^8 = 256$ possibilities to recover the secret key. Therefore, in side-channel, the vulnerabilities come no-more from the algorithm itself, but from the way that this algorithm is implemented in a hardware chip. This means that, form the designer point of view, it is crucial to perfectly know the hardware architecture of the chip and to be very aware of any possible power leakage that may occur.

Nowadays, systems are build with sensors that are able to detect intrusive attacks and algorithms are developed to bring counter-measures to the leakages.

1.3.2 Attack

In practice, an attack happens in the following way as described by François-Xavier Standaert in [82]. In this framework, we notice that most of the attack follow the same pattern.

- As mentioned in Section 1.2, the encryption algorithm is known by the attacker. In most of the cases it will be AES (sometimes RSA or DES).

- The first phase of an attack is called the *profiling* phase. The attacker builds a modelization of the leakage using a copy of the target device. This modelization can be under a leakage model function or via histograms called *template*. Of course, as this profiling phase is made on a copy of the target device, the secret key of this copy is known. A drawback of this technique is that there may exist a bias between the model obtained with the copy device and the target device.

- Then, the exploitation phase is based on applying the model obtained during the profiling phase to the target device.

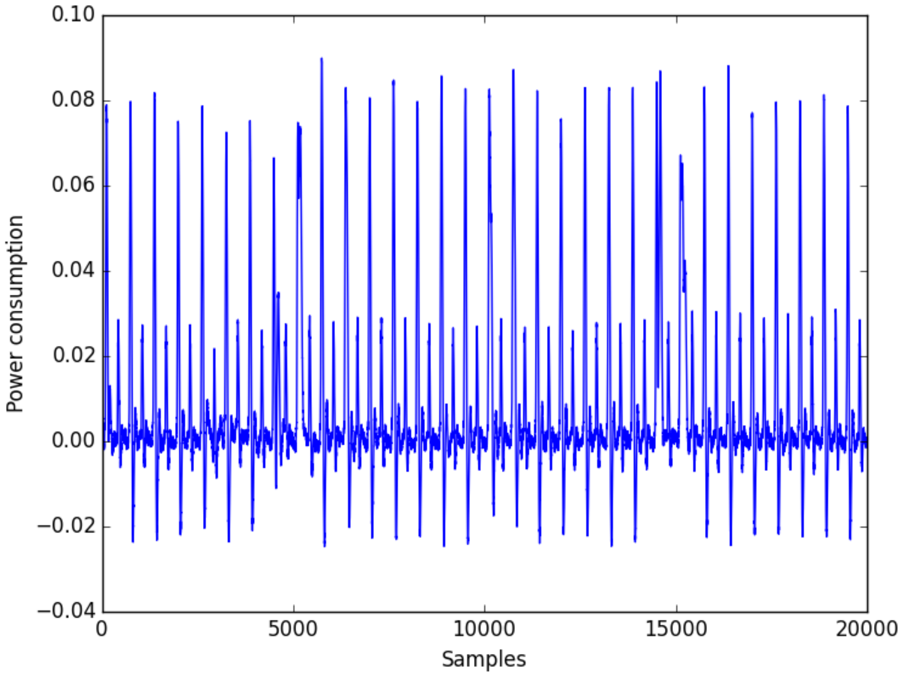


Figure 1.4: One power consumption trace of DES algorithm

Example: the DPA Contest In 2008, the DPA contest [84] was launched by Télécom Paristech. The challenge was to recover a secret key used in an encrypting algorithm DES with a very little number of traces. The participants were provided very big sets of data such as:

- For each query, the 64 bits plaintext to be encrypted;
- The corresponding 64 bits ciphertext;
- The power consumption of the whole encryption process.

A trace has the shape given in Figure 1.4. With one figure like this, it is not possible to recover the secret key. However, if we average them according to existing leakage models such as the hamming weight leakage [51], we can notice some points of interest. For example, the SNR of the leakage of one DES substitution box is given in Figure 4.8. In this figure, we notice that given a leakage model, the SNR is relevant in three points. We call these samples *points of interest*. By selecting only theses points of interest and computing mathematical functions called *distinguishers*, it is possible to recover the secret key. For the first edition of the DPA contest,

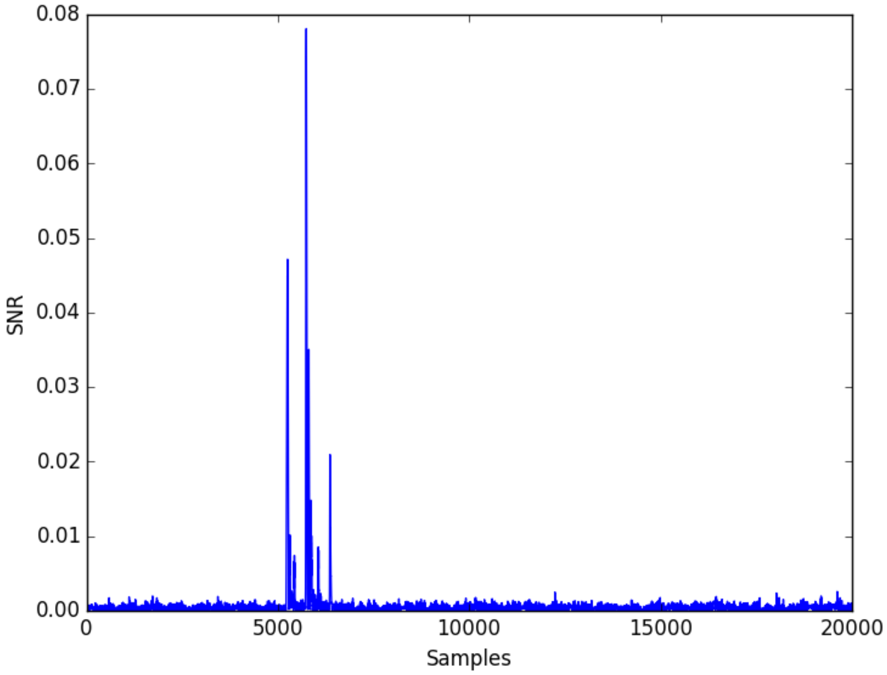


Figure 1.5: The SNR of this leakage according to the Hamming weight model.

the best attacking method was provided by Christophe Clavier [21]. In his method, the full key recovery takes only 43 traces in average.

1.4 A Look on Information Theory

1.4.1 Background

The communication of information has benefited from many improvements at the physical level. However, it is also profitable to optimize the data rate by studying how information shall be preprocessed before being sent. This need has given rise to *Information Theory*: the science of data transmission. It was created in 1948 by Claude E. Shannon from the Bell labs in his famous article *A Mathematical Theory of Communication* [78]. In this article, for the first time, the basis of digital communication were drawn. The idea was: how to send some information from a sender to a receiver through a noisy channel? To do so, Shannon proposed a framework for a communication channel, from the message to be sent, to the received message. Figure 1.6 shows this communication system designed in 1948.

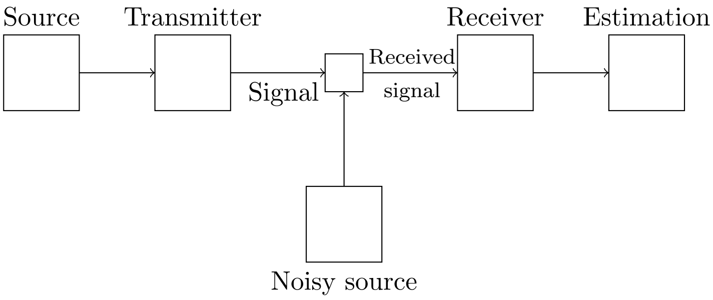


Figure 1.6: A communication system by Shannon

For the first time, a precise model was proposed to describe distant communications. But Shannon did not only describe the model. With probabilistic considerations, he proved that it is possible to send messages with an arbitrary small error at the decoding phase, as long as the coding rate (i.e. the amount of data per sample) is lower than a given limit. Moreover, this limit has an analytic expression and is called the *Mutual Information* between the signal and the received signal. As the transmitter has access to the way that the data is sent, it is possible to modify the Mutual Information, and therefore reach the highest possible limit, called the *Channel Capacity*. When the noisy source is additive and the noise is Gaussian, the capacity of the channel takes the well-known expression $\mathcal{C} = \frac{1}{2}\log(1 + \mathsf{SNR})$, where SNR is the signal to noise ratio, i.e. the power or the input signal divided by the power power of the noise. This formula is known as the Shannon capacity.

However, Shannon did only predict that there exists ways to transmit data that reach this bound. He did not tell how to find one. For mobile communications, the first time that a team managed to implement a transmission scheme reaching the Shannon capacity, was in 1996 when Berrou and Glavieux invented the Turbo-codes [7].

1.4.2 Link With Side-Channel Analysis

The attractive point of information theory is that its fields of applications are wide. Indeed, in the case of side-channel analysis, we can consider that the secret key is an information, and that the leakage is a transmission.

In 2014, Annelie Heuser proposed a diagram where both notions of side-channel analysis and information theory are represented [39]. We have copied this figure in Figure 1.8. The notations are the following:

- K^* is the random variable standing for the leaking secret key.
- $\mathbf{T}$ is the random vector standing for the plaintext vector.
- $\mathbf{Y}$ is the random vector standing for the sensitive variable vector.
- $\mathbf{N}$ is the random vector standing for the additive noise (most often supposed as Gaussian).
- $\mathbf{X}$ is the random vector standing for the measured traces.
- $\widehat{K}$ is the random variable standing for the estimated key. If the attack is efficient, the estimated key is equal to the secret key.
- The functions f and φ are respectively the algorithmic function (for example a SubBytes function and the leakage model function. The leakage model function can be supposed to be known or estimated. The best case for the attacker is of course when the leakage function is known.
- $\mathcal{D}$ is the distinguishing rule. It is a mathematical function taking as inputs the traces and the plaintext, and returning an estimation of the secret key.

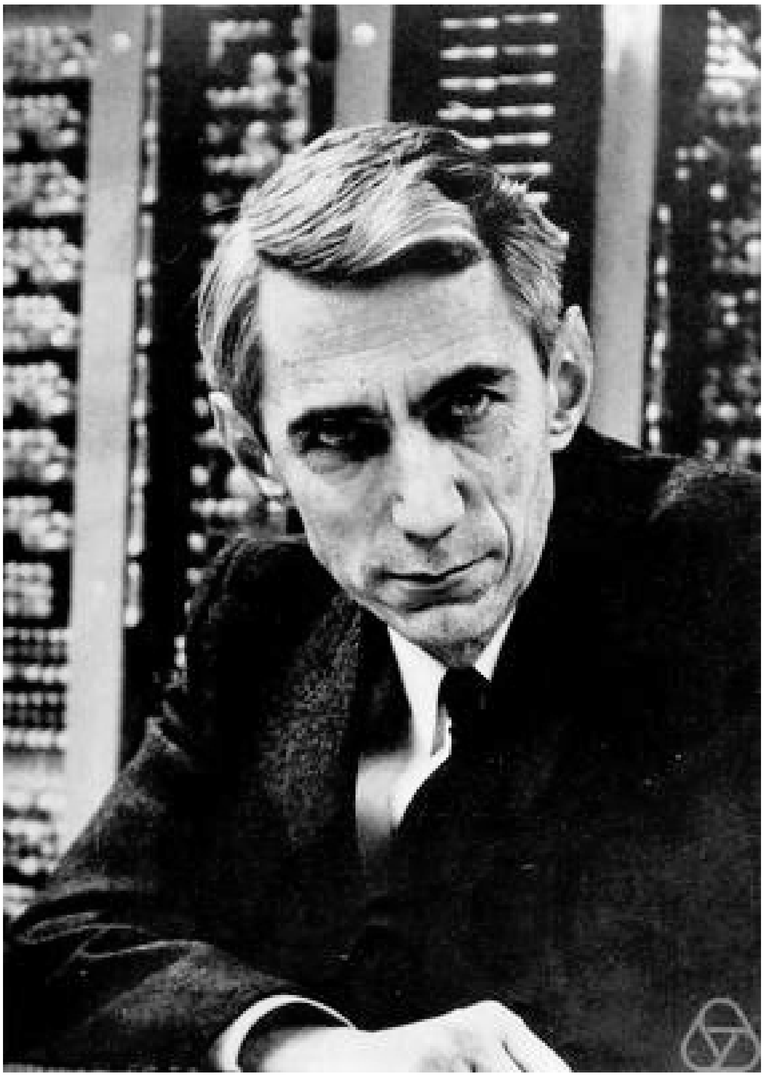


Figure 1.7: Claude Shannon ©MFO

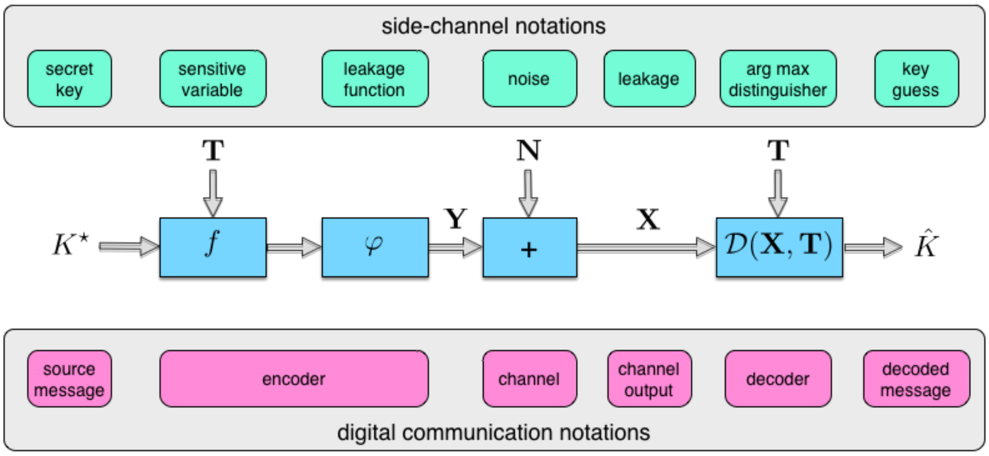


Figure 1.8: Framework linking communication channels with leakage model.

1.5 Organization of the Manuscript

The manuscript is organized as follows. In Chapter 2, I describe the main contributions of my thesis. Part II deals with a generic upper bound to any type of attack thanks to information theoretic results. In Part III, I consolidate the knowledge of some distinguishers. More specifically, Chapter 5 deals with monobit leakages while Chapter 6 shows that Mutual Information Analysis can be optimal in some scenarios. In Part IV, we discuss about practical issues that may happen in real world devices, in particular for timing attacks.

1.6 Notations

All over this manuscript, we will use the following notations for the mathematical derivations. The sets will be written with calligraphic letters, and elements of such sets in small caps. If possible we will use the same letter. for example $x \in \mathcal{X}$. Random variables will be written in capital letter. For example, X is a random variable taking values in $\mathcal{X}$. Probabilities are written with the symbol $\mathbb{P}$. Therefore, the probability that X is x is noted $\mathbb{P}(X = x)$. If it is clear that the random variable is X, we will only write $\mathbb{P}(x)$. We use bold letters to write vectors. For example $\mathbf{x}$ is a vector whose all the element are in $\mathcal{X}$. And $\mathbf{X}$ is a random vector taking values in $\mathcal{X}$. If the set $\mathcal{X}$ is continuous, the probability distribution function is written as $p(x)$.

We also recall some information theoretic definitions that are used in this manuscript. The entropy of a random variable X is defined as:

$$H(X) = -\sum_{x \in \mathcal{X}} \mathbb{P}(x) \log \mathbb{P}(x).$$

The conditional entropy of X knowing Y is defined as:

$$\begin{aligned} H(X \mid Y) &= \sum_{y \in \mathcal{Y}} \mathbb{P}(y) H(X \mid Y = y) \\ &= \sum_{y \in \mathcal{Y}} \mathbb{P}(y) \sum_{x \in \mathcal{X}} \mathbb{P}(x \mid y) \log \mathbb{P}(x \mid y). \end{aligned}$$

The mutual information between two random variables X and Y is defined as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \\ &= \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \mathbb{P}(x,y) \log \frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)} \end{aligned}$$

For these definitions, we have not precised the base of the logarithm since it is mathematically possible to choose any base. However, in communication theory, base 2 is very often used and information is consequently expressed in bits. This is why we will use this base for all the logarithms.

Chapter 2

Contributions

Contents

2.1 Formal Security Proofs and Studies

During my thesis, my main concern has been: how to unify and understand the link between the different metrics used to evaluate the sensitive information leakage of cryptographic chips? Indeed, various different metrics have been proposed by various authors in respond to specific issues. For example, we can cite:

- The Mutual Information (MI) between measurements and the sensitive variables;
- The Perceived Information computed with the *estimated* distribution of the leakage, based on template attacks;
- The Success Rate or the Guessing Entropy of an attack exploiting a leakage.

2.1.1 A Generic Bound for Any Leakage With Only SNR

For example, the notion of MI is a widely spread concept in Side-Channel Analysis [30], and everyone agrees that the larger the MI, the better the attack, or from the defender's side, the weaker the chip. However, the link between MI and the success rate of an attack, has never been established formally yet for generic attacks. Some links have already been made [29, 53] but they have been made for particular types of attacks, especially to show how the masking order impacts the MI [67] and are therefore true in a specific context. Moreover, I have noticed that no paper really deals with a tight prediction of the security of a chip. My question was: how can a designer predict the success rate of an attack with a very restricted number of assumptions. Indeed, a designer does not know how a device will be attacked. New methods may appear several years after the conception of the chip. This is why I have started looking for a framework that can unify any type of attack under simple concepts and notations.

This is the beginning of my reflection: if we use some information theoretic metrics, then the answer must be in this field.

In order to better formalize a side-channel attack, I based myself on the framework given by François-Xavier Standaert in [82]. This framework presents a simple albeit comprehensive look on what a side-channel attack is.

In 2014, Annelie Heuser has published an article [39] where she demonstrates that the optimal distinguisher for an attack is the Maximum Likelihood decoder. To obtain this result, she establishes a first formal link between side-channel analysis and information theory. The leakage model is there seen as a communication channel with a message to recover (i.e. the secret key). The method to derive the optimal distinguisher has been to express the expression for the key decoding rule which maximizes the success rate.

As MI is an information theoretic metric, it becomes here natural to study its impact with information theoretic models and theorems. I have therefore worked on this aspect. The main difficulty for me has been to deal with the philosophical differences between information theory and side-channel. Indeed, the purpose of each is clearly different:

- The goal of an information theorist is to send a message with the highest possible rate and with an arbitrary small error rate. To do so, the communication engineer can use correcting codes and choose how the message is encoded.

- In side-channel analysis, the message is the secret key. But the attacker do not have access to it (which is obvious). This means that the input distributions are imposed by the model and that there is not coding possibility to improve the success rate.

However, Shannon's coding theorem and converse coding theorem are still useful to determine bounds. Indeed, the Shannon's converse coding theorem shows that the probability of success of a transmission is upper-bounded by the mutual information of the channel [79]. I have used this major theorem (proved 60 years ago!) and applied it to my side-channel model.

I have therefore applied the theorem to side-channel. The main problem that I have encountered is a formal calculation of the Mutual Information. Indeed, the leakages *are not* independent and identically distributed (i.i.d.). This means that the Mutual Information between the traces and the sensitive variables cannot be easily estimated via Shannon's formula

$\mathcal{C} = \frac{1}{2}\log_2(1 + \mathsf{SNR})$. This one remains only a loose upper-bound of the real MI. Therefore, I had to resort to other methods to estimated the MI as tight as possible.

I therefore proposed two new methods to estimate the MI of large vectors:

1. A mathematical upper-bound that converges to the right value when the number of traces q tends to infinity;

2. A parametric estimation based on empirical results.

The mathematical approach is true for any leakage model and any type of noise. However, the bound is not tight for small values of q. In this case, a good tradeoff is to merge our approach with Shannon's bound. Our parametric estimation is due to empirical observations with Additional White Gaussian Noise (AWGN). Indeed, I have noticed that the MI between two vectors behaves like an error function. With such estimation, the only knowledge of the Signal-to-Noise Ratio (SNR) is enough to approximate the MI. This leads to a computation of the Success Rate (SR) of an attack that can be computed only with the calculation of the SNR and with the assumption of an additive white Gaussian noise. We do not suppose anything on the leakage model.

I have then compared the bound obtained by these estimations of the MI, with the best possible distinguisher i.e. the Maximum Likelihood distinguisher (as demonstrated by Heuser et al. in [39]). The results show that the tighter the estimation of the MI, the tighter the bound is.

2.1.2 Extension to Template Attacks

My work on the best possible success rate for a given number of traces relies on the calculation of the Mutual Information between the sensitive variable and the measured traces. In many cases, the leakage model is not perfectly known and it has to be estimated. This is the case for example, when the attacker has a copy of the targeted device and learns the leakage model from this copy. These are called *template attacks.* Here the goal is different because we want to know how fast an attacker can break a secret key after a learning phase.

This means that the attacker does not know the real leakage model, but an estimation that may even be biased. François-Xavier Standaert proposed the notion of *Perceived Information* (PI) [30] to replace a MI that cannot be computed because of the lack of knowledge.

A first formal study about this PI shows that it is obviously lower than the MI and can even be smaller than zero. But we wish a link between PI and SR. I have shown that, when PI is strictly positive, the attack will succeed with a sufficient enough number of traces. This result is based on the demonstration of the Shannon theorem and a proof based on the mismatched decoders by Merhav [57].

2.1.3 MIA is Universal ML

On the distinguishing point of view, I have noticed that when the attacker has to profile with *on-the-fly* data, the Maximum Likelihood distinguisher is perfectly equivalent to a Mutual Information Analysis (MIA).

We have shown with theoretical case-study that MIA is very relevant when the leakage model is not perfectly known. Indeed, we have built an experiment where Correlation Power Analysis (CPA) crashes while MIA works well to recover a secret key.

2.1.4 A Unified Vision of Monobit Leakages

Several papers noticed that some distinguishers related to monobit leakages can be linked with Fei et al.'s confusion coefficient [32]. For example, Heuser et al. derived the Kolmogorov-Smirnov Analysis (KSA) as a function of the confusion coefficient and the SNR [38]. Moreover, in [51], Mangard et al. made the link between Correlation Power Analysis (CPA) and the confusion coefficient.

After reading these articles, we had the feeling that the link between any distinguisher of a monobit leakage, was deeper. Therefore, we first made the link between Mutual Information Analysis (MIA) and the confusion coefficient. We noticed that it also depends on the standard deviation of the noise. We extracted an analytic function linking MIA the confusion coefficient, and the standard deviation of the noise.

Eventually, we have noticed that for monobit leakages, the confusion coefficient can also be seen as the transition probability of a Binary Symmetric Channel (BSC). This has allowed us to prove that any sound distinguisher in monobit leakages is a function of two parameters: the confusion coefficient and the standard deviation of the noise.

2.2 Adapting Theoretical Tools for Practical Issues

Another task of my thesis was to adapt some of the theoretical tools for practical issues. I based my studies on an ARM processor edited by ST Microelectronics: the STM32 Discovery Board [59].

On this architecture, we compute timing attacks on the AES algorithm. Here, in this practical example, the leakage model is not easy to find and to deal with. A template attack is therefore needed to learn the model. However a learning phase may encounter some problems such as:

- A bias between the learned model and the real leakage;
- A poor learning phase.

2.2.1 Avoid the Empty Bin Issue

One of the issues that we have met is the *empty bin issues.* The empty bin issue appears when there is a difference between the distribution of the learning phase and the distribution of the attack. We may face a strange case: when we compute the maximum likelihood distinguisher based on the distribution of the templates, we can meet some data such that the probability is null. This appears even for the correct key guess and therefore, the ML distinguisher crashes.

We have therefore imagined several solutions to avoid this empty bin issue but still keeping the notion of maximum likelihood that is supposed to be the best distinguishing rule according to Annelie Heuser's paper [39].

The solutions that we have imagined are easy to compute and are sound. When the profiling is correct, the best distinguisher is a ML with a small penalty if an empty bin occurs. On the contrary, when the profiling is poor, the best possible distinguisher is to compute an MIA based on the learned model. Indeed, MIA is known to be more robust when the leakage model is not well characterized.

2.2.2 Extract a Model for a Timing Attack

In addition to the empty bin issue, I have worked on the STM32 Discovery board in order to extract a leakage model. In a black box view, we have as inputs the plaintext that is to be encoded and the number of clock cycles to compute AES as the output of this black box.

With this architecture, it is possible to enable or not the data cache (DC) or the instruction cache (IC). We have noticed that, when the DC is enabled, the computation of AES is not time constant, meaning that there is a leakage. A part of this leakage is still difficult to understand but we have managed to find out that the number of cache hits during the computation of the algorithm has a great impact.

Part II

A Mathematical Bound on Success Rate

Introduction

As a general rule, the most successful man in life
is a man who has the best information.
— Benjamin Disraeli.

Side-channel analysis is renown as an effective “eavesdropping” attack technique to extract sensitive secrets from cryptographic chips. In recent literature, many exploits have been put forward. Starting from the seminal timing attack of Kocher [44], various biases of different kinds have been exhibited. Vertical attacks such as power analysis [45] have been shown to be highly efficient. However, from a designer’s viewpoint, the exact details of the various attacks are irrelevant. Instead, defenders aim at estimating a security risk in general, e.g., the chance that a major security breach occurs. It is thus highly desired to protect designs against all kinds of SCA attacks in a provable way. When implementing a secure design, the natural question which arises is the quantification of its security, with respect to its architecture and its operational environment. In [30], the authors present several metrics that can help the designers to secure cryptographic chips. Shannon’s mutual information (MI) between measured traces and guessed models has been considered, but is often thought of as theoretical (too far from practical evaluations) and impracticable (too computationally inefficient). In [83], the authors explain the relative importance of MI and probability of success, but in a separate way. Our aim is to join the two concepts and to show how the knowledge of MI allows to derive an upper bound on the success rate.

We wish to estimate the success rate with very few assumptions, based on simple and easy-to-compute tools, such as the signal to noise ratio (SNR). The calculation of the SNR can be made without the knowledge of the leakage model as the SNR is the ratio between the power of the useful signal and the power of the noise. The power of the noise is easily measured as is is the measurement noise, and as the power of the useful signal is the difference between the power of the measured signal and the power of the noise, the SNR is obtained.

Related Work As our main goal is to find an estimation of the success rate of an attack that can be as accurate as possible. Using Information theoretic tools, [39] extracted the best possible distinguishing rule. However, this does not give any clue to estimate the success rate of an attack. In practice, the success rate is estimated by repeating a sufficient number of simulations. Moreover, this is dependent of the knowledge of the leakage model. In practice, it is difficult to know exactly this model. Indeed, the estimation may be biased, the learning phase of the model, may be too short, the model, may be too complicated, etc. This is why, we wish to use general information theoretic tools in order to be as generic as possible, and to give bounds that are true whatever the attacker may do or may know.

In [50], a link between the success rate and the number of traces to succeed in a correlation power analysis [11] has been studied, and an analytical formula has been derived. However, this results is untrustworthy in practice because of the assumption that incorrect key guesses lead to independent distinguishers, which is not true. Subsequent work on this topic therefore consider the joint distribution of all values of the distinguisher (correct key and all remaining incorrect key guesses).

In [36, 48, 74], the authors propose an estimation of the success rate of specific distinguishers. Namely, Rivain [74] studies the distribution of two examples of distinguishers (correlation and template) in the presence of normal noise. Lomné et al. [48] extend this work for masked implementations, while however still focusing on correlation and template attacks. Guilley et al. [36] extend the approach from additive to some non-additive distinguishers (such as the mutual information analysis), but through the approximation that the number of traces tends to the infinity. To summarize, all three papers [36, 48, 74] have in common that the knowledge of the leakage model, or at least an estimation via a learning phase with templates, is needed to predict the success rate. In addition, this estimation, in the three cases, is based on the central limit theorem, meaning that it is relevant for a large number of traces and only for additive distinguishers. We wish a bound valid for any distinguisher, for any number of traces (even small).

A bound on the Mutual Information is proposed in [67]. The MI involved is based on one trace, supposing that every leakage is independent from each other. We show in this part that this is not the case in practice. In this paper, the bound is valid for MI with only *one measurement*. We will see in this part of the manuscript that calculating MI with the probability functions of *all* the traces is crucial.

In [29, Theorem 2], the authors proposed a link between success rate and the number of measurements. This bound is based on the the link between MI and random probing. Therefore, it is valid only for leakages with very low SNR and the bound is very loose For instance (see Figure 4.6), with $\mathsf{SNR} > 10^{-4}$, the bound of Duc et al. [29] is trivial (the success rate is smaller than one), and for $\mathsf{SNR} = 10^{-5}$, it predicts a number of traces 4, which is much smaller than our result of 1.3×10^6 (where the best attack using ML predicts 1.5×10^6, which is in the order of magnitude of our prediction). In fact, the main contribution of the bound of Duc et al. [29] is to show that the masking order of an attack has an exponential impact on the success rate, but not to yield an accurate link between number of traces and success rate.

In the field of information theory, Arimoto [2] proved a lower bound of the error rate (hence an upper-bound of the success rate) in terms of a so-called Gallager coefficient. However, not only requires intensive computations, but also the model assumes a freely chosen input distribution. In our case, that input distribution is set by the leakage model and therefore, cannot be freely chosen. Arimoto's main result (Equation 24 of [2]) remains true because it represents the best possible case for an attacker for all possible input distributions; but the resulting bound is very loose in our side-channel context. Equation 9 of [2] could be used instead but depends on a parameter β. With our notations (presented in section 3.1), Arimoto's Equation 9 becomes:

$$\forall \beta > 0, \qquad \mathrm{P}_e \leq 1 - 2^{n(\beta-1)} \sum_{\mathbf{t} \in \mathcal{T}^q} \mathbb{P}(\mathbf{t}) \sum_{\mathbf{x} \in \mathcal{X}^q} \left[\sum_{k=0}^{2^n-1} \mathbb{P}(k) \mathbb{P}(\mathbf{x} \mid k, \mathbf{t})^{1/\beta} \right]^{\beta}.$$

The minimization of the r.h.s is practical untraceable for $q > 1$. Indeed, it consists in sums over $|\mathcal{X}|^q$ elements; the complexity is even worse when the output is continuous.

Overall, we can sum up the related work with the following table 2.1. The table classifies the state-of-the-art according various criteria, such as the way the results are derived and whether or not the mutual information is involved in the estimation of the success rate. The last two columns show whether a closed form bound exists and whether it is generic in the attack method. Our method provided an analytic expression for the lower bound (Theorem 3.1) and is agnostic in the attack method.

Contributions In this chapter, we derive bounds on the success rate of any attack, irrespective to the exact attack. Thus we can consider our bounds as *universal*. To do so, we address this problem using rigorous information theoretic tools. This is why we revisit the use of MI as a conservative security metric. Our main contribution is to give a clear relationship between MI and probability of success. More precisely, we seek a lower bound on the number of available

Related work	**Link with information theory**	**Usage of MI**	**Closed form bound on SR**	**Generic**
[39]	Yes	No	No	No
[74] [48] [36]	No	No	Yes (but asymptotic)	No
[67]	No	Yes	No	Yes
[29]	No	Yes	Yes (but very loose)	Yes
[2]	Yes	No	Computationally too difficult	Yes
This part	Yes	Yes	Yes (Theorem 3.1)	Yes

Table 2.1: Summary of the related work

traces where a given success level can be reached, based only on theoretical assumptions on the channel. The actual value of MI is important to estimate and such an estimation is not immediate because random vectors of very high dimensions are involved in its expression. Therefore, we propose several ways to simply estimate the MI by mathematically proved upper bounds and by numerical estimations. Our results are applied to the most common type of noise, namely the additive white Gaussian noise. We show that, in the case of additive Gaussian noise, the only calculation of the SNR is sufficient enough predict accurately the security of a device. Last, the main result on success rate is translated in terms of guessing entropy, another informative criterion in side-channel analysis.

Organization This part is organized as follows. In Chapter 3, we provide the mathematical computaions to prove this bound and we apply them in the case of additive white Gaussian noise. Section 3.1 describes the side-channel and shows how a leakage can be modeled with a Markov chain. Section 3.2 provides our main result and three different ways to exploit it. An application to leakages with additive Gaussian noise is carried out in Section 3.3, where we show at the end that the SNR is enough to predict the security of a device. The link to the guessing entropy is done in Section 3.4. In Chapter 4, we show how we can tighten the bound with numerical estimations of the mutual information. We give a general conclusion of both chapters in Section 4.4. Technical computations involved in proofs are in Appendix.

Notations Throughout this paper we use the following notations. Calligraphic letters (e.g. $\mathcal{X}$) denote sets. Uppercase letters (e.g. X) denote random variables taking their values in the corresponding set (e.g. $\mathcal{X}$). Lowercase letters (e.g. x) denote realizations of this random variable. Vectors are written in bold characters. By default, the length of a vector is $q \in \mathbb{N}$. Thus, a random vector is denoted with a bold capital letter (e.g. $\mathbf{X} = (X_1, X_2, \ldots, X_q)$) and a vector of realizations on this random vector is denoted with a small bold letter (e.g. $\mathbf{x} = (x_1, x_2, \ldots, x_q)$). Given the random variable X taking its values in $\mathcal{X}$ and $x \in \mathcal{X}$, the probability that X equals x is noted $\mathbb{P}(X = x)$ or simply $\mathbb{P}(x)$.

We also define some information theoretic tools. The entropy of a random vector $\mathbf{X}$ of length q is defined by:

$$H(\mathbf{X}) = -\sum_{\mathbf{x} \in \mathcal{X}^q} \mathbb{P}(\mathbf{x}) \log_2 \mathbb{P}(\mathbf{x}).$$

The conditional entropy of a random vector $\mathbf{X}$ knowing vector $\mathbf{Y}$ is defined by:

$$\begin{aligned} H(\mathbf{X} \mid \mathbf{Y}) &= -\sum_{\mathbf{y} \in \mathcal{Y}^q} \mathbb{P}(\mathbf{y}) H(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}) \\ &= -\sum_{\mathbf{y} \in \mathcal{Y}^q} \mathbb{P}(\mathbf{y}) \sum_{\mathbf{x} \in \mathcal{X}^q} \mathbb{P}(\mathbf{x} \mid \mathbf{y}) \log_2 \mathbb{P}(\mathbf{x} \mid \mathbf{y}). \end{aligned}$$

The Mutual Information between two random vectors $\mathbf{X}$ and $\mathbf{Y}$ is defined as $I(\mathbf{X};\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{Y})$. The conditional Mutual Information $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{T}$ are random vectors is defined as $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T}) = H(\mathbf{X} \mid \mathbf{T}) - H(\mathbf{X} \mid \mathbf{Y}, \mathbf{T})$. Last, the Kullback-Leibler divergence between two distributions $\mathbb{P}$ and $\mathbb{Q}$ over the same set $\mathcal{X}$ is defined as:

$$\mathrm{D}(\mathbb{P} \| \mathbb{Q}) = \sum_{x \in \mathcal{X}} \mathbb{P}(x) \log_2 \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}.$$

Chapter 3

A Mathematical Bound of the Success Rate with Mutual Information

3. A MATHEMATICAL BOUND OF THE SUCCESS RATE WITH MUTUAL INFORMATION

This chapter presents a mathematical theory of leakage models. Some open issues are discussed in Appendix A.

Contents

3.1 Side-Channel Seen as a Communication Channel

The link between side-channel analysis and information theory has been proposed by [39] to derive the optimal distinguisher. In this section, we review how the side-channel can be seen as a communication channel. The secret key byte that the attacker wants to recover is denoted as k^* and is n bits long (typically $n = 8$). We assume that the attacker inputs q text bytes $\mathbf{t} = (t_1, t_2, \ldots, t_q)$ and receives that many traces in a vector $\mathbf{x} = (x_1, x_2, \ldots, x_q)$, with the following *leakage model*:

$$x_i = f(t_i \oplus k^*) + n_i \qquad (i = 1, 2, \ldots, q) \tag{3.1}$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_q)$ is an additive noise independent of $\mathbf{x}$ and $f(.)$ is some leakage function. We assume that f is deterministic but not necessary known to the attacker. This assumption will make our calculations generic and therefore true for any type of attack. This is the worst possible case for the security designers. Define the *sensitive variable* $\mathbf{y}(k) = \mathbf{y_t}(k)$ as

$$\mathbf{y_t}(k) = f(\mathbf{t} \oplus k) = (f(t_1 \oplus k), \ldots, f(t_q \oplus k)) \tag{3.2}$$

so that the leakage can be written in compact form as

$$\mathbf{x} = \mathbf{y_t}(k^*) + \mathbf{n}.$$

Such vectors $\mathbf{t}, \mathbf{y}$ and $\mathbf{x}$ are realizations of random vectors noted $\mathbf{T}, \mathbf{Y}$ and $\mathbf{X}$. In the case of one particular sample, t, y and x are realizations of random variables T, Y and X. We assume that the channel is *memoryless*, which means that each trace x_i depends on the input $\mathbf{y}$ only from y_i. In particular x_i and y_j are independent for all if $i \neq j$. We also make the natural assumption that the secret key is independent from all text bytes: the secret key random variable K is independent from $\mathbf{T}$. In other words, the text bytes do not give any information about the secret key (at least in a design which adheres to Kerckhoffs's principle).

Following [39] we make the following hypotheses:

- K is uniformly distributed over $\mathcal{K} = \{0, \ldots, 2^n - 1\}$. K is a scalar (there is one key-byte to break), and is therefore not written in bold font.
- T is uniformly distributed over $\mathcal{T} = \{0, \ldots, 2^n - 1\}$. Moreover, we suppose that vector $\mathbf{T}$ is *balanced*, meaning that the number of occurrences of each symbol in the vector is the same.
- As seen above, the random variable Y is such that $Y = f(T \oplus K)$, with f a known deterministic function.
- As q textbytes are sent and therefore q traces are received, we consider the random vectors $\mathbf{T}, \mathbf{Y}$ and $\mathbf{X}$.

Thus from (3.1), we can write

$$\begin{aligned}\mathbf{X} &= f(\mathbf{T} \oplus K) + \mathbf{N} \\ &= \mathbf{Y} + \mathbf{N}.\end{aligned}$$

Considering only scalars, this writes for random variables

$$\begin{aligned}X &= f(T \oplus K) + N \\ &= Y + N.\end{aligned}$$

After acquiring q traces, the attacker applies a function called *distinguisher* $\mathcal{D}$ to obtain an estimate $\widehat{K} = \mathcal{D}(\mathbf{X}, \mathbf{T})$ of the secret key from $\mathbf{X}$ and $\mathbf{T}$. This allows us to define the communication channel as depicted in Figure 3.1:

- the "encoder" models the leakage from the device: not only the composition of the algorithm which mixes the unknown key K with the known text $\mathbf{T}$ into a sensitive variable, but also the way the device leaks the sensitive variable (function f);

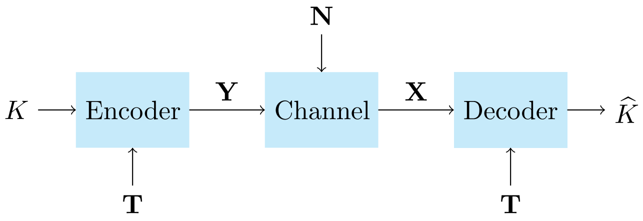


Figure 3.1: Representation of Side-Channel

- the (side) channel consists in noise addition, arising from the untargeted parts of the design and from the measurement setup; and
- the "decoder" implements the *distinguishing rule* with allows the attacker to get a key guess $\widehat{K}$ from the measured leakage $\mathbf{X}$ and the knowledge of public text bytes $\mathbf{T}$. The realizations $\mathbf{t}$ of the random vector $\mathbf{T}$ are known by the attacker.

From the model we can deduce Lemma 3.1 dealing with Markov chains.We recall that a Markov chain is *a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.*

Lemma 3.1. *The communication channel just described admits the following Markov chains:*

$$
\begin{aligned}
&(K,\mathbf{T}) \longrightarrow (\mathbf{Y},\mathbf{T}) \longrightarrow (\mathbf{X},\mathbf{T}) \longrightarrow \widehat{K} \qquad (3.3)\\
&K \longrightarrow \mathbf{Y} \longrightarrow \mathbf{X} \longrightarrow \widehat{K}\\
&(K,\mathbf{T}) \longrightarrow \mathbf{Y} \longrightarrow \mathbf{X}.
\end{aligned}
$$

Proof. The first case is easily seen by re-drawing Figure 3.1 into the different constitutive blocks as shown in Figure 3.2, where all the variables pass through different blocks corresponding to the Markov Chain. The two other cases are proved similarly. □

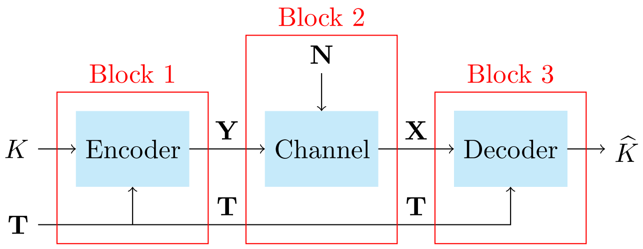


Figure 3.2: The Markov chain $(K,\mathbf{T}) \longrightarrow (\mathbf{Y},\mathbf{T}) \longrightarrow (\mathbf{X},\mathbf{T}) \longrightarrow \widehat{K}$.

3.2 Theoretical Bounds on Mutual Information

One of the important properties of a Markov chain is the *data processing inequality*[23], which is used to prove the following theorem in this section, which is our main result.

3.2.1 Main Result

Let $\mathrm{P}_s = \mathbb{P}(\widehat{K} = K)$ be the probability of success of an attack and $H_2(\mathrm{P}_s)$ its binary entropy[1] [23]:

$$H_2(\mathrm{P}_s) = -\mathrm{P}_s \log_2(\mathrm{P}_s) - (1 - \mathrm{P}_s)\log_2(1 - \mathrm{P}_s).$$

The following theorem is fundamental because it provides a trade-off for any possible type of attack.

Theorem 3.1. *The following inequality is always true for any distinguishing rule:*

$$H(K) - (1 - \mathrm{P}_s)\log_2(2^n - 1) - H_2(\mathrm{P}_s) \leq q \cdot I(X; Y \mid T). \tag{3.4}$$

The probability of success of an attack also follows the following inequality:

$$\begin{aligned} &H(K) - (1 - \mathrm{P}_s)\log_2(2^n - 1) - H_2(\mathrm{P}_s) \\ &\leq \mathbb{E}_{\mathbf{T}}\mathbb{E}_{K_1}\log_2 \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K1,\mathbf{T}} \| \mathbb{P}_{\mathbf{X}|K_2,\mathbf{T}})\right); \end{aligned} \tag{3.5}$$

where $\mathrm{D}(\mathbb{P}\|\mathbb{P}')$ *is the Kullback-Leibler divergence [23] and* K_1, K_2 *are identically distributed as* K*.*

Merging these two equations we can write:

$$\begin{aligned} &H(K) + (\mathrm{P}_s - 1)\log_2(2^n - 1) - H_2(\mathrm{P}_s) \\ &\leq \min(\mathbb{E}_{\mathbf{T}}\mathbb{E}_{K_1}\log_2 \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K1,\mathbf{T}} \| \mathbb{P}_{\mathbf{X}|K_2,\mathbf{T}})\right), qI(X; Y \mid T)). \end{aligned} \tag{3.6}$$

This theorem shows that the success rate of an attack is directly linked to the Mutual Information between the leakage and the model. Furthermore, as we consider generic attacks, this inequality remains true whatever the attacker does with the traces. In the next subsections we prove both inequalities and we show that (3.4) is more interesting for low values of q while (3.5) is a better approximation for high values of q.

To do so, we first demonstrate a preliminary lemma in Section 3.2.2 that will be useful for both Equation (3.4) and (3.5).

[1] The binary entropy is the entropy of a binary random variable with probabilities p and $1 - p$.

3.2.2 A Fundamental Lower Bound on Mutual Information $I(\mathbf{X};\mathbf{Y}\mid\mathbf{T})$

The first step of the demonstration of Theorem 3.1 is the following lemma that links the Mutual Information between the random vectors $\mathbf{X}$ and $\mathbf{Y}$ with the probability of success.

Lemma 3.2. *With the notations of Theorem 3.1, we have:*

$$H(K) - (1-\mathrm{P}_s)\log_2(2^n-1) - H_2(\mathrm{P}_s) \leq I(\mathbf{X};\mathbf{Y}\mid\mathbf{T}). \tag{3.7}$$

Proof. Using the Markov Chain (3.3) we compare two MI values thanks to the *data processing inequality*[4]. Indeed, this is a direct consequence of Lemma 3.1. This inequality states that the further two random variables are in a Markov Chain, the less MI between these variables. Here we have

$$I((K,\mathbf{T})\,;(\mathbf{X},\mathbf{T})) \leq I((\mathbf{Y},\mathbf{T})\,;(\mathbf{X},\mathbf{T})). \tag{3.8}$$

Let us expand both sides of this inequality. In the l.h.s., since the channel is memoryless and K and $\mathbf{T}$ are independent, we have:

$$\begin{aligned} I((K,\mathbf{T});(\mathbf{X},\mathbf{T})) &= H(K,\mathbf{T}) - H((K,\mathbf{T})\mid(\mathbf{X},\mathbf{T})) \\ &= H(K) + H(\mathbf{T}) - H(K\mid\mathbf{T},\mathbf{X}). \end{aligned}$$

As $\widehat{K}$ is a deterministic function of $\mathbf{T}$ and $\mathbf{X}$, adding the knowledge of $\widehat{K}$ does not change the entropy:

$$\begin{aligned} I((K,\mathbf{T});(\mathbf{X},\mathbf{T})) &= H(K) + H(\mathbf{T}) - H(K\mid\mathbf{T},\mathbf{X},\widehat{K}); \\ &\geq H(K) + H(\mathbf{T}) - H(K\mid\widehat{K}). \end{aligned}$$

The latter inequality holds since conditioning reduces entropy [23]. Now by Fano's inequality[1] [23, Page 43],

$$H(K\mid\widehat{K}) \leq H_2(\mathrm{P}_e) + \mathrm{P}_e\log_2(|\mathcal{K}|-1)$$

where P_e is the probability of error $\mathrm{P}_e = \mathbb{P}(K\neq\widehat{K})$. Since $\mathrm{P}_s = 1-\mathrm{P}_e$ and $H_2(\mathrm{P}_e) = H_2(\mathrm{P}_s) = -\mathrm{P}_e\log_2(\mathrm{P}_e) - \mathrm{P}_s\log_2(\mathrm{P}_s)$, this is rewritten as

$$H(K\mid\widehat{K}) \leq H_2(\mathrm{P}_s) + (1-\mathrm{P}_s)\log_2(2^n-1).$$

Plugging this inequality into the previous one gives

$$I((K,\mathbf{T});(\widehat{K},\mathbf{T})) \geq H(K) + qH(T) - H_2(\mathrm{P}_s) - (1-\mathrm{P}_s)\log_2(2^n-1). \tag{3.9}$$

On the other hand, the r.h.s. of the data processing inequality (3.8) is:

$$\begin{aligned} I((\mathbf{Y},\mathbf{T});(\mathbf{X},\mathbf{T})) &= H(\mathbf{X},\mathbf{T}) - H(\mathbf{X},\mathbf{T}\mid\mathbf{Y},\mathbf{T}); \\ &= H(\mathbf{X},\mathbf{T}) - H(\mathbf{X}\mid\mathbf{Y},\mathbf{T}); \\ &= I(\mathbf{X};\mathbf{Y}\mid\mathbf{T}) + H(\mathbf{T}). \end{aligned} \tag{3.10}$$

[1] Fano's inequality is an important information-theoretic result about the uncertainty of the transmission of a message, which is due to the error probability and the number of possible errors.

Combining Equations (3.9) and (3.10), we obtain the following fundamental inequality:

$$H(K) - H_2(\mathrm{P}_s) - (1 - \mathrm{P}_s)\log_2(2^n - 1) \leq I(\mathbf{X}; \mathbf{Y} \mid \mathbf{T}), \tag{3.11}$$

And proving Lemma 3.2. □

The same l.h.s. of (3.11) will be used to prove for both inequalities (3.4) and (3.5), the difference being the way that $I(\mathbf{X}; \mathbf{Y} \mid \mathbf{T})$ is evaluated. Indeed, the next part of the proofs for Equations (3.4) and (3.5) is about finding an upper-bound for $I(\mathbf{X}; \mathbf{Y} \mid \mathbf{T})$. We have to do so because there is no analytic expression for this conditional Mutual Information computed with vectors of q dimensions.

Remark 3.1. A quick analysis of the value $n + (\mathrm{P}_s - 1)\log_2(2^n - 1) - H_2(\mathrm{P}_s)$ reveals that it is always non-negative for any P_s in the range $(0, 1)$ and vanishes if and only if $\mathrm{P}_s = 1/2^n$.

Therefore, when there are no traces, $I(\mathbf{X}; \mathbf{Y} \mid \mathbf{T}) = 0$, the only probability that can respect inequality (3.11) is $\mathrm{P}_s = 1/2^n$, meaning that without information, that attacker can not have a better success rate than $1/2^n$ obtained with an equiprobable random guess, as expected. Every trace will bring additional information and therefore increase the probability of success.

3.2.3 First Upper Bound on $I(\mathbf{X}; \mathbf{Y} \mid \mathbf{T})$: Proof of Inequality (3.4)

Thanks to Lemma 3.2, the l.h.s. of Theorem 3.1 is given. Inequality (3.4) is a straightforward consequence of the following lemma.

Lemma 3.3. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two random vectors with joint distribution* $\mathbb{P}_{\mathbf{X},\mathbf{Y}}$*,* $\mathbb{P}_{\mathbf{X}}$ *be the marginal distribution of* $\mathbf{X}$*, and* $\mathbb{P}_X$ *be the marginal of one element* X *of vector* $\mathbf{X}$*. Define the distribution* $\widetilde{\mathbb{P}}_{\mathbf{X}} = \prod_{i=1}^{q} \mathbb{P}_{X_i}$*. We have*

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}) &= qI(X; Y) - \mathrm{D}\left(\mathbb{P}_{\mathbf{X}} \| \widetilde{\mathbb{P}}_{\mathbf{X}}\right); \\ &\leq qI(X; Y). \end{aligned}$$

This Lemma means that the Mutual Information of two random vectors made of identically distributed random variables is lower than q times the Mutual Information of the marginal distribution of these random vectors.

Proof. From the memoryless assumption of the channel, one has $\mathbb{P}_{\mathbf{X}|\mathbf{Y}} = \prod_{i=1}^{q} \mathbb{P}_{X_i|Y_i}$. Thus

$$\begin{aligned}
I(\mathbf{X};\mathbf{Y}) &= \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[\log_2 \frac{\mathbb{P}_{\mathbf{X}|\mathbf{Y}}(\mathbf{X}\mid\mathbf{Y})}{\mathbb{P}_{\mathbf{X}}(\mathbf{X})}\right] \\
&= \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[\log_2 \frac{\mathbb{P}_{\mathbf{X}|\mathbf{Y}}(\mathbf{X}\mid\mathbf{Y})}{\widetilde{\mathbb{P}}_{\mathbf{X}}(\mathbf{X})}\right] + \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[\log_2 \frac{\widetilde{\mathbb{P}}_{\mathbf{X}}(\mathbf{X})}{\mathbb{P}_{\mathbf{X}}(\mathbf{X})}\right] \\
&= \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[\log_2 \frac{\prod_i \mathbb{P}_{X|Y}(X_i\mid Y_i)}{\prod_i \widetilde{\mathbb{P}}_{X}(X_i)}\right] - \mathrm{D}\left(\mathbb{P}_{\mathbf{X}}\|\widetilde{\mathbb{P}}_{\mathbf{X}}\right) \\
&= \sum_i \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[\log_2 \frac{\mathbb{P}_{X|Y}(X_i\mid Y_i)}{\widetilde{\mathbb{P}}_{X}(X_i)}\right] - \mathrm{D}\left(\mathbb{P}_{\mathbf{X}}\|\widetilde{\mathbb{P}}_{\mathbf{X}}\right) \\
&= qI(X;Y) - \mathrm{D}\left(\mathbb{P}_{\mathbf{X}}\|\widetilde{\mathbb{P}}_{\mathbf{X}}\right).
\end{aligned}$$

The inequality follows since the divergence is always non-negative. □

This upper bound on MI is easily derived but is linear in q, and, therefore, will not converge to a finite value as the number of measurements increases ($q \to \infty$). This will be in contradiction with Lemma 3.4. Therefore, it is interesting to propose another bound that converges to a finite value. This will be made in the next section.

3.2.4 Second Upper Bound on $I(\mathbf{X};\mathbf{Y}\mid\mathbf{T})$ - Proof of Inequality (3.5)

Before proving (3.5) we first notice that in our side-channel model, as there is a finite number of keys, the MI is always bounded by $H(K)$.

Lemma 3.4.

$$I(\mathbf{X};\mathbf{Y}\mid\mathbf{T}) = I(K;\mathbf{X}\mid\mathbf{T}) \leq H(K)$$

Proof. We use the Markov chain defined in Equation (3.3). Notice that, adding the knowledge of $\mathbf{T}, K$ when $\mathbf{T},\mathbf{Y}$ are already known does not change the entropy of $\mathbf{X}$. Therefore,

$$\begin{aligned}
H(\mathbf{X}\mid\mathbf{T},\mathbf{Y}) &= H(\mathbf{X}\mid\mathbf{T},\mathbf{Y},K,\mathbf{T}); \\
&= H(\mathbf{X}\mid\mathbf{T},\mathbf{Y},K).
\end{aligned}$$

As $\mathbf{Y}$ is a deterministic function of K and $\mathbf{T}$, it can be removed, so we get:

$$H(\mathbf{X}\mid\mathbf{T},\mathbf{Y}) = H(\mathbf{X}\mid\mathbf{T},K).$$

Therefore, we obtain $I(\mathbf{X};\mathbf{Y}\mid\mathbf{T}) = I(\mathbf{X};K\mid\mathbf{T})$. Since $I(\mathbf{X};K\mid\mathbf{T}) = H(K) - H(K\mid\mathbf{T},\mathbf{X})$ in follows that $I(\mathbf{X};K\mid\mathbf{T}) \leq H(K)$. □

Here $H(K)$ is a constant that depends only on the distribution of K; it reaches its maximum value for a uniform distribution: $H(K) = n$ bit. As a consequence, since $I(\mathbf{X};\mathbf{Y}\mid\mathbf{T})$ increases with q, it must converge to a finite value when $q \to \infty$. This explains why the upper-bound given by (3.4) is poor when $q \to \infty$.

Therefore, we provide another bound that is more accurate for large values of q because it converges to a finite value when K is finite. First we need the following

Lemma 3.5. *For any random variables X and Y and real-valued function $(x, y) \mapsto f(x, y)$,*

$$-\mathbb{E}_Y \log_2 \mathbb{E}_X[\exp(f(X,Y))] \leq -\log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y f(X,Y))].$$

Proof. See Appendix A.2. □

Corollary 3.1. *For any random variables X and Y and positive function $(x, y) \mapsto g(x, y)$,*

$$\exp \mathbb{E}_Y \log_2 \mathbb{E}_X[g(X,Y)] \geq \mathbb{E}_X[\exp(\mathbb{E}_Y \log g(X,Y))]$$

Proof. See Appendix A.3. □

Equipped with Lemma 3.5, we compute MI as follows:

$$\begin{aligned} I(\mathbf{X}; K \mid \mathbf{T}) &= \mathbb{E}_{\mathbf{T}} \mathbb{E}_{\mathbf{X},K|\mathbf{T}} \log_2 \frac{\mathbb{P}(\mathbf{X} \mid K\mathbf{T})}{\mathbb{P}(\mathbf{X} \mid \mathbf{T})}; \\ &= \mathbb{E}_{\mathbf{T}} \mathbb{E}_K \mathbb{E}_{\mathbf{X}|K,\mathbf{T}} \log_2 \frac{\mathbb{P}(\mathbf{X} \mid K\mathbf{T})}{\mathbb{P}(\mathbf{X} \mid \mathbf{T})}; \end{aligned}$$

We introduce here K_2, a random variable following the same distribution as K.

$$\begin{aligned} I(\mathbf{X}; K \mid \mathbf{T}) &= \mathbb{E}_{\mathbf{T}} \mathbb{E}_K \mathbb{E}_{\mathbf{X}|K,\mathbf{T}} \log_2 \frac{\mathbb{P}(\mathbf{X} \mid K, \mathbf{T})}{\mathbb{E}_{K_2} \mathbb{P}(\mathbf{X} \mid K_2, \mathbf{T})}; \\ &= -\mathbb{E}_{\mathbf{T}} \mathbb{E}_K \mathbb{E}_{\mathbf{X}|K,\mathbf{T}} \log_2 \mathbb{E}_{K_2} \frac{\mathbb{P}(\mathbf{X} \mid K_2, \mathbf{T})}{\mathbb{P}(\mathbf{X} \mid K, \mathbf{T})}; \\ &= -\mathbb{E}_{\mathbf{T}} \mathbb{E}_K \mathbb{E}_{\mathbf{X}|K,\mathbf{T}} \log_2 \mathbb{E}_{K_2} \exp\left[\log_2 \frac{\mathbb{P}(\mathbf{X} \mid K_2, \mathbf{T})}{\mathbb{P}(\mathbf{X} \mid K, \mathbf{T})}\right]. \end{aligned}$$

By Lemma 3.5 we obtain

$$\begin{aligned} I(\mathbf{X}; K \mid \mathbf{T}) &\leq -\mathbb{E}_{\mathbf{T}} \mathbb{E}_K \log_2 \mathbb{E}_{K_2} \exp\left[\mathbb{E}_{\mathbf{X}|K,\mathbf{T}} \log_2 \frac{\mathbb{P}(\mathbf{X} \mid K_2, \mathbf{T})}{\mathbb{P}(\mathbf{X} \mid K, \mathbf{T})}\right]; \\ &= -\mathbb{E}_{\mathbf{T}} \mathbb{E}_K \log_2 \mathbb{E}_{K_2} \exp\left[-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K,\mathbf{T}} \mid\mid \mathbb{P}_{\mathbf{X}|K_2,\mathbf{T}})\right]. \end{aligned}$$

This proves inequality (3.5) and Theorem 3.1 [1].

[1] An alternative proof of inequality (3.5), which resorts only on convexity arguments, is given in Appendix A.4.

3.3 Application to Additive White Gaussian Noise

In this section, we develop the results of Theorem 3.1 for leakages with additional white Gaussian noise. Indeed, this is the most common case for attacks such as DPA, where the noise comes from the measurement tools.

With this model, we can link the success rate to Shannon's capacity $C = \frac{1}{2}\log(1+\mathsf{SNR})$, and therefore, to the SNR, where $\mathsf{SNR} = \frac{\mathrm{VAR}(Y)}{\sigma^2}$. Moreover, at the end of this section, we will extract a parametric estimation of the Mutual Information where the only parameter to know is the SNR.

Remark 3.2. With additive white Gaussian noise, the SNR of the traces can also been written as:

$$\mathsf{SNR} = \frac{\mathrm{Var}(Y)}{\sigma^2},$$

where σ is the standard deviation of the noise.

3.3.1 Shannon's Channel Capacity

Under the additive white Gaussian noise (AWGN) assumption, it is easily seen that the scalar mutual information $I(X;Y \mid T)$ does not exceed Shannon's capacity. Indeed, we have:

$$\begin{aligned}
I(X;Y \mid T) &= \mathbb{E}_T I(X;Y \mid T=t);\\
&= \mathbb{E}_T\left[H(X \mid T=t) - H(X \mid Y, T=t)\right];\\
&= \mathbb{E}_T\left[H(f(T\oplus K)+N \mid T=t)\right] - H(X \mid Y);\\
&= \mathbb{E}_T\left[H(f(t\oplus K)+N)\right] - H(X \mid Y);\\
&= H(f(K)+N) - H(X \mid Y);\\
&\leq \frac{1}{2}\log_2(2\pi e(\mathrm{Var}_K(f(K)) + \mathrm{Var}(N))) - H(X \mid Y);\\
&= \frac{1}{2}\log_2(1+\mathrm{SNR}).
\end{aligned}$$

Combining this with inequality (3.4) yields a lower bound on the number of traces to reach a given probability of success:

$$q \geq \frac{n + (\mathrm{P}_s - 1)\log_2(2^n - 1) - H_2(\mathrm{P}_s)}{\frac{1}{2}\log_2(1+\mathrm{SNR})} \tag{3.12}$$

Remark 3.3. The number of traces q to be sure to recover the key is lower-bounded by:

$$\lim_{\mathrm{P}_s \to 1} q \geq \frac{n}{\frac{1}{2}\log_2(1+\mathrm{SNR})}. \tag{3.13}$$

However, since as we have seen the MI can never be higher than $H(K)$, the above constant bound is not accurate for real attacks. The next subsection provides a much more accurate estimation.

3.3.2 Evaluation of the Kullback-Leibler Divergence

Inequality (3.5) gives an upper bound with a divergence term that depends on $\mathbb{P}_{\mathbf{X}|K_i,\mathbf{T}}$ $(i = 1, 2)$. In the AWGN model, $\mathbb{P}_{\mathbf{X}|K_i,\mathbf{T}}$ follows a multivariate normal distribution $\mathcal{N}(\mathbf{y}(K_i, \mathbf{T}), \sigma^2 I_q)$. For such distributions, the divergence is very easy to compute as the covariance matrix is diagonal. It is easily found that

$$\mathrm{D}(\mathbb{P}_{\mathbf{X}|K,\mathbf{T}} \| \mathbb{P}_{\mathbf{X}|K_2,\mathbf{T}}) = \frac{\|\mathbf{y}(K, \mathbf{T}) - \mathbf{y}(K_2, \mathbf{T})\|_2^2}{2\sigma^2}.$$

Inequality (3.5), when applied to the AWGN model, becomes

$$n + (\mathrm{P}_s - 1)\log_2(2^n - 1) - H_2(\mathrm{P}_s) \leq -\mathbb{E}_{\mathbf{T}}\mathbb{E}_K \log_2 \mathbb{E}_{K_2} \exp\left(-\frac{\|\mathbf{y}(K, \mathbf{T}) - \mathbf{y}(K_2, \mathbf{T})\|_2^2}{2\sigma^2}\right).$$

In order to make a precise evaluation of the r.h.s., we need several lemmas.

Lemma 3.6. *Let* $\mathbf{t} = (t_1, \dots, t_q) \in \mathcal{T}^q$ *and* $(k_1 \neq k_2) \in \mathcal{K}^2$. *One has*

$$\lim_{q\to\infty} \|\mathbf{y}(k_1, \mathbf{t}) - \mathbf{y}(k_2, \mathbf{t})\|_2^2 = +\infty \tag{3.14}$$

and more precisely:

$$\|\mathbf{y}(k_1, \mathbf{t}) - \mathbf{y}(k_2, \mathbf{t})\|_2^2 \underset{q\to\infty}{\sim} q.\alpha(k_1, k_2), \tag{3.15}$$

where $\alpha(k_1, k_2) = \frac{1}{2^n}\sum_{t=0}^{2^n-1}(y(k_1, t) - y(k_2, t))^2$.

Proof. We make use of the assumption made in Section 3.1 that $\mathbf{T}$ is *balanced*. For $k_1 \neq k_2$, we have

$$\begin{aligned}\|\mathbf{y}(k_1, \mathbf{t}) - \mathbf{y}(k_2, \mathbf{t})\|_2^2 &= \sum_{i=1}^{q}(\mathbf{y}(k_1, t_i) - \mathbf{y}(k_2, t_i))^2;\\ &= q\sum_{i=1}^{q}\frac{(\mathbf{y}(k_1, t_i) - \mathbf{y}(k_2, t_i))^2}{q};\\ &= q\sum_{t\in\mathcal{T}}\frac{n_t(\mathbf{y}(k_1, t_i) - \mathbf{y}(k_2, t_i))^2}{q};\end{aligned}$$

where n_t is the number of times that a particular $t \in \mathcal{T}$ appears in vector $\mathbf{t}$. As $\mathbf{t}$ is balanced, $\frac{n_t}{q} \to \frac{1}{|\mathcal{T}|}$ and therefore:

$$\|\mathbf{y}(k_1, \mathbf{t}) - \mathbf{y}(k_2, \mathbf{t})\|_2^2 \underset{q\to\infty}{\sim} q\sum_{t\in\mathcal{T}}\frac{(\mathbf{y}(k_1, t_i) - \mathbf{y}(k_2, t_i))^2}{|\mathcal{T}|}.$$

□

Lemma 3.7. *Let $\mathbf{t} \in \mathcal{T}^q$ be fixed and $k \in \mathcal{K}$ be a fixed key. We have*

$$\lim_{q\to\infty} -\log_2 \mathbb{E}_{K_2} \exp\left(-\frac{\|\mathbf{y}(k,\mathbf{t}) - \mathbf{y}(K_2,\mathbf{t})\|_2^2}{2\sigma^2}\right) = n \tag{3.16}$$

$$-\log_2 \mathbb{E}_{K_2} \exp\left(-\frac{\|\mathbf{y}(k,\mathbf{t}) - \mathbf{y}(K_2,\mathbf{t})\|_2^2}{2\sigma^2}\right) \underset{q\to\infty}{\sim} -\log_2 \mathbb{E}_{K_2} \exp\left(-\frac{q.\alpha(k,K_2)}{2\sigma^2}\right). \tag{3.17}$$

Proof. One has

$$-\log_2 \mathbb{E}_{K_2} \exp\left(-\frac{\|\mathbf{y}(k,\mathbf{t}) - \mathbf{y}(K_2,\mathbf{t})\|_2^2}{2\sigma^2}\right) = -\log_2\left[\sum_{k_2} \frac{1}{2^n} \exp\left(-\frac{\|\mathbf{y}(k,\mathbf{t}) - \mathbf{y}(k_2,\mathbf{t})\|_2^2}{2\sigma^2}\right)\right]$$

When q is a multiple of 2^n we have exactly

$$\|\mathbf{y}(\mathbf{t},k_1) - \mathbf{y}(\mathbf{t},k_2)\|_2^2 = q.\alpha(k_1,k_2)$$

and the proof of Equation (3.17) is trivial. Otherwise, for $k \neq k_2$ we have $\exp(-q\frac{\alpha(k,k_2)}{2\sigma^2}) \to 0$ as $q \to \infty$; and for $k = k_2$ we have $\exp(-q\frac{\alpha(k,k_2)}{2\sigma^2}) = 1$. Therefore

$$-\log_2\left[\sum_{k_2} \frac{1}{2^n} \exp\left(-\frac{\|\mathbf{y}(k,\mathbf{t}) - \mathbf{y}(k_2,\mathbf{t})\|_2^2}{2\sigma^2}\right)\right] \longrightarrow n.$$

□

Lemma 3.8. *With the assumptions made in Section 3.1, we have as $q \to \infty$:*

$$\mathbb{E}_{\mathbf{T}}\mathbb{E}_K \log_2 \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K}||\mathbb{P}_{\mathbf{X}|K_2})\right) \underset{q\to\infty}{\sim} n - \frac{n_{\min}}{2^n}\exp(-q.\min_{k_1\neq k_2}\alpha(k_1,k_2)) \tag{3.18}$$

where $n_{\min}$ is the number of indexes $k_1 \neq k_2$ reaching the minimum value of $\alpha(k_1,k_2)$.

This simple asymptotic expression can be used to upper-estimate the MI for high values of q. Notice that for any $k_1 \neq k_2$, $\alpha(k_1,k_2) = \alpha(k_2,k_1)$, hence $n_{\min}$ is an even number.

Proof. Let $\mathbf{t} = (t_1,\ldots,t_q)$ be a balanced vector. By Lemma 3.7, we have

$$-\mathbb{E}_K \log \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K\mathbf{t}}||\mathbb{P}_{\mathbf{X}|K_2\mathbf{t}})\right) \underset{q\to\infty}{\sim} -\mathbb{E}_K \log \mathbb{E}_{K_2} \exp\left(-\frac{q.\alpha(K,K_2)}{2\sigma^2}\right)$$

where

$$\begin{aligned}-\mathbb{E}_K \log \mathbb{E}_{K_2} \exp\left(-\frac{q.\alpha(K,K_2)}{2\sigma^2}\right) &= -\mathbb{E}_K \log\left[\frac{1}{2^n}\sum_{k_2}\exp\left(-\frac{q.\alpha(K,k_2)}{2\sigma^2}\right)\right];\\ &= n - \mathbb{E}_K \log\left[1 + \sum_{k_2\neq K}\exp\left(-\frac{q.\alpha(K,k_2)}{2\sigma^2}\right)\right].\end{aligned}$$

As the value inside the logarithm vanishes as $q \to \infty$, consider its first-order Taylor expansion:

$$\begin{aligned} -\mathbb{E}_K \log \mathbb{E}_{K_2} \exp\left(-\frac{q.\alpha(K,K_2)}{2\sigma^2}\right) &\underset{q\to\infty}{\sim} n - \mathbb{E}_K \left[\sum_{k_2 \neq K} \exp\left(-\frac{q.\alpha(K,k_2)}{2\sigma^2}\right)\right]; \\ &= n - \frac{1}{2^n} \sum_{k_1 \neq k_2} \left[\exp\left(-\frac{q.\alpha(K,k_2)}{2\sigma^2}\right)\right]. \end{aligned}$$

Let $k_1 \neq k_2$ be a couple such that $\alpha(k_1,k_2)$ is the minimum of all the possible α. For any other couple $k_3 \neq k_4$, there are two possibilities:

1. either $\alpha(k_3,k_4) = \alpha(k_1,k_2)$ and the corresponding exponentials will converge at the same rate;
2. or $\alpha(k_3,k_4) > \alpha(k_1,k_2)$ and $\exp\left(-\frac{q}{2\sigma^2}\alpha(k_3,k_4)\right)$ is negligible w.r.t. $\exp\left(-\frac{q}{2\sigma^2}\alpha(k_1,k_2)\right)$.

Hence we can simply count the number of occurrences of the minimum value of α. We have proven that:

$$-\mathbb{E}_K \log \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K\mathbf{t}} || \mathbb{P}_{\mathbf{X}|K_2\mathbf{t}})\right) \underset{q\to\infty}{\sim} n - \frac{n_{\min}}{2^n} \exp\left(-\frac{q.\min_{k_1\neq k_2} \alpha(k_1,k_2)}{2\sigma^2}\right).$$

As this expansion is true for any vector $\mathbf{t}$ that is balanced, and is independent of it, this proves the lemma. □

Remark 3.4. The simplification of Lemma 3.8 is useful to obtain a simple equivalent form for high values of q. However, it is also possible to compute a tight approximation of the numerical value of $\mathbb{E}_{\mathbf{T}}\mathbb{E}_K \log_2 \mathbb{E}_{K_2} \exp\left(-\mathrm{D}(\mathbb{P}_{\mathbf{X}|K} || \mathbb{P}_{\mathbf{X}|K_2})\right)$.

Remark 3.5. Interestingly, we notice that parameter $\alpha(k_1,k_2)$ is proportional to the *confusion coefficient* $\kappa(k_1,k_2)$ defined first in [32] for binary leakages, and extended in [36, Equation (45)] for any leakage:

$$\kappa(k_1,k_2) = 4\alpha(k_1,k_2).$$

3.4 Link with Guessing Entropy

Another way to quantify the quality of an attack is the *Guessing Entropy* [55], defined as $H(K \mid \mathbf{X},\mathbf{T})$. This metric quantifies the complexity of the exclusive search to recover K knowing the side-channel measurements. Besides, let N_K be the average number of tries to retrieve the secret key K with the knowledge of $\mathbf{X}$ and $\mathbf{T}$. Mathematically, we have:

$$N_K = \mathbb{E}_{\mathbf{XT}}\left[\sum_k \delta_{\mathbf{XT}}(k)\mathbb{P}(k \mid \mathbf{X},\mathbf{T})\right],$$

where $\delta_{\mathbf{XT}}(\cdot)$ is the permutation that re-orders the probabilities $\mathbb{P}(k \mid \mathbf{X}, \mathbf{T})$ into the decreasing order. There exists a relationship between N_K and $H(K \mid \mathbf{X}, \mathbf{T})$ called the inequality of Massey [55, Section 2]:

$$N_K \geq 2^{H(K|\mathbf{X},\mathbf{T})-2} + 1.$$

We propose here an improved inequality relating N_k with $H(K \mid \mathbf{X}, \mathbf{T})$.

Lemma 3.9 (Improved Inequality of Massey)**.** *The average number of tries to recover the correct key is upper-bounded by:*

$$N_K > \frac{2^{H(K|\mathbf{X},\mathbf{T})}}{e}. \tag{3.19}$$

Our inequality improves Massey's inequality as soon as the entropy is greater than $\log_2(\frac{e}{1-e/4})$.

Proof. Let $b_k = \frac{(1-1/N_K)^k}{N_K-1}$ for all $k \in \mathbb{N}^*$. As $\sum_k b_k = 1$, b_k is a distribution (geometric). Moreover, by the Gibbs inequality [23],

$$\begin{aligned}
H(K \mid \mathbf{X}, \mathbf{T}) &= -\sum_{\mathbf{t},\mathbf{x}} \mathbb{P}(\mathbf{tx}) \sum_k \mathbb{P}(k \mid \mathbf{t}, \mathbf{x}) \log_2 \mathbb{P}(k \mid \mathbf{t}, \mathbf{x}) \\
&\leq -\sum_{\mathbf{t},\mathbf{x}} \mathbb{P}(\mathbf{t}, \mathbf{x}) \sum_k \mathbb{P}(k \mid \mathbf{t}, \mathbf{x}) \log_2 b_{\delta_{\mathbf{XT}}(k)} \\
&= -\sum_{\mathbf{t},\mathbf{x}} \mathbb{P}(\mathbf{t}, \mathbf{x}) \sum_k \mathbb{P}(k \mid \mathbf{t}, \mathbf{x}) \delta_{\mathbf{X},\mathbf{T}}(k) \log_2(1 - 1/N_K) + \log_2(N_K - 1) \\
&= -\log_2(1 - 1/N_K) N_K + \log_2(N_K - 1) \\
&= N_K H_2(1/N_K)
\end{aligned}$$

In fact, the inequality is strict since equality would hold if and only if $\mathbb{P}(k \mid \mathbf{X}, \mathbf{T}) = b_{\delta_{\mathbf{X},\mathbf{T}}(k)}$, which is not the case as the support of $\mathbb{P}$ is finite and the support of b_k is not. Therefore, we have proven that:

$$H(K \mid \mathbf{X}, \mathbf{T}) < N_K H_2(1/N_K).$$

Last, we notice that the function $f(x) = x \log_2(x)$ is convex ($f'(x) = \log_2(ex)$ is increasing). Therefore, fore any x in the range $]0, 1[$, we have:

$$\frac{f(x) - f(x-1)}{x - (x-1)} \leq f'(x) = \log_2(ex).$$

When we apply this for $x = N_K$, we get:

$$\begin{aligned}
N_K H_2(1/N_K) &= N_K \log_2(N_K) - (N_K - 1) \log_2(N_K - 1) \\
&\leq \log_2(eN_K).
\end{aligned}$$

Overall, this means that $H(K \mid \mathbf{X}, \mathbf{T}) < \log_2(eN_K)$ which proves the lemma. □

The lemma can be exploited by replacing $H(K \mid \mathbf{X}, \mathbf{T})$ by $\log_2(eN_K)$ in Subsection 3.2.2. Therefore, instead of using Fano's inequality, we directly have

$$I((K, \mathbf{T}); (\mathbf{X}, \mathbf{T})) \geq H(K) + H(\mathbf{T}) - \log_2(eN_K),$$

leading to:

$$N_K \geq \frac{2^{-I(\mathbf{X};\mathbf{Y}|\mathbf{T})+H(K)}}{e}. \tag{3.20}$$

Once more, we can use Theorems (3.4) and (3.5) to estimate the mutual information.

For example, we suppose that we have a Gaussian channel, with $\mathrm{SNR} = 1/8$ and $q = 40$ traces. We apply Equation (3.4) to obtain that $I(\mathbf{X}; \mathbf{T} \mid \mathbf{T}) \leq q\frac{1}{2}\log(1 + \mathrm{SNR})$. For a $n = 8$ bits leakage, the average number of tries is lower-bounded by:

$$\begin{aligned} N_K &\geq \frac{-2^{20*\log_2(1+1/8)+8}}{e} \\ &\approx \frac{2^{4.6}}{e} \\ &\approx 8.9 \end{aligned}$$

This means that, for such a channel, it would take at least 8 tries to recover one byte of the secret key with 40 traces. However, a secret key is made of 16 or even 32 bytes. Supposing that the attacker has only 40 traces for each key-byte, after the attack, one would need at least $8.9^{16} \approx 1.6 \times 10^{15}$ tries in average to recover the entire key as there is no way to check only byte per byte.

Chapter 4

Application

In this Chapter, we apply the results of Lemma 3.2 and Theorem 3.1 to practical cases. In addition, we also discuss about the difficulty to estimate the mutual information $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ and therefore, we provide numerical estimations based on the law of grat numbers. With these estimations, we then notice that that they fit well with a parametric estimation.

Contents

4.1 Numerical Approximations for Mutual Information

4.1.1 Numerical Estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$

Theorem 3.1 gave analytic bounds to the success rate. However, one may need to obtain a precise value of $I(\mathbf{X};\mathbf{Y}|\mathbf{T})$ making the bound tighter. In this section, we propose numerical tools to obtain an accurate value of the Mutual Information as a function of the number of queries q. A full estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ by numerical integration becomes impossible for q-dimensional distributions, and we have recourse to simplifying approximations of MI. Since

$$\begin{aligned} I(\mathbf{X};\mathbf{Y} \mid \mathbf{T}) &= H(\mathbf{X} \mid \mathbf{T}) - H(\mathbf{X} \mid \mathbf{Y},\mathbf{T}) \\ &= H(\mathbf{X} \mid \mathbf{T}) - H(\mathbf{X} \mid \mathbf{Y}) \end{aligned}$$

we can estimate only the entropy $H(\mathbf{X} \mid \mathbf{T})$ because $H(\mathbf{X} \mid \mathbf{Y}) = qH(X \mid Y)$ is easily computable with classical numerical tools.

One possible approximation is from the law of large numbers [23, Chapter 3]:

$$H(\mathbf{X} \mid \mathbf{T}) = \lim_{J\to\infty} -\frac{1}{J} \sum_{\mathbf{t}\in\mathcal{T}^q} \sum_{j=1}^{J} \mathbb{P}(\mathbf{t}) \log_2 \mathbb{P}(\mathbf{x}_j \mid \mathbf{t}). \tag{4.1}$$

Unfortunately, such a computation is not tractable since it involves the sum over all balanced vectors $\mathbf{t}$, which represents $q!$ possibilities. However, we can obtain a good approximation of $H(\mathbf{X} \mid \mathbf{T})$ with only one vector $\mathbf{t}$ form the following

Lemma 4.1 (A Symmetry Property). *Let $\mathbf{t} = (t_1, \ldots, t_q) \in \mathcal{T}$ and τ be a permutation in $\{1, \ldots, q\}$. Noting $\tau(\mathbf{t}) = (t_{\tau(1)}, \ldots, t_{\tau(q)})$, we have:*

$$H(\mathbf{X} \mid \mathbf{T} = \mathbf{t}) = H(\mathbf{X} \mid \mathbf{T} = \tau(\mathbf{t})). \tag{4.2}$$

Proof. See Appendix A.1. □ □

As a consequence of the symmetry of Lemma 4.1, one needs only one balanced vector $\mathbf{t}$ to estimate $H(\mathbf{X} \mid \mathbf{T})$. Therefore, by the law of large numbers,

$$H(\mathbf{X} \mid \mathbf{T}) \approx \lim_{J \to \infty} -\frac{1}{J} \sum_{j=1}^{J} \log_2 \mathbb{P}(\mathbf{x}_j \mid \mathbf{t}). \tag{4.3}$$

This leads to Algorithm 1 to evaluate the entropy $H(\mathbf{X} \mid \mathbf{T})$.

Algorithm 1: Computation of the entropy using the law of large numbers.

```
input  : A balanced vector $\mathbf{t}$
         An integer $J$
         The probability distribution $\mathbb{P}(\mathbf{x} \mid \mathbf{t})$
output : An approximation of $H(\mathbf{X} \mid \mathbf{T})$
Hxt ← 0 ;
Generate a secret key byte $k^*$ ;
for $j \leftarrow 0$ to $J$ do
    Generate the traces $\mathbf{x}$ with the model ;
    Hxt ← Hxt $-\frac{1}{J} \log_2 \mathbb{P}(\mathbf{x} \mid \mathbf{t})$;
end
return Hxt
```

When the leakage models are not perfectly known (e.g. template attacks), a possible way to estimate Mutual Information is to approximate numerically the distributions. An example is given in [35].

Other estimation methods can be used, depending on the distribution of the noise. As an example, for Gaussian noise, we may consider Gaussian mixtures as discussed in [43].

Such numerical estimations are all the more accurate as J is taken large, which means that they make take a tremendous amount of time to compute. Having $I(\mathbf{X}; \mathbf{T}|\mathbf{T})$ as a function of q, even numerically estimated, is very useful as we have the link between the success rate and the minimum number of traces to reach such probability of success.

4.1.2 Graphical Comparison

In order to visualize the difference between the two upper bounds given above, we have plotted the mutual information $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T}=\mathbf{t})$, where $\mathbf{t}$ is a fixed balanced vector. The leakage model chosen is given by the equation

$$y(k,t_i) = \mathrm{H_w}(\mathrm{S_{box}}(t_i \oplus k)) \qquad (i = 1, 2, \ldots, q)$$

where $\mathrm{H_w}(\cdot)$ is the Hamming weight (of the value written in binary), and $\mathrm{S_{box}}(\cdot)$ is the AES substitution box[24]. We suppose that the zero-mean additive white Gaussian noise (AWGN) has standard deviation $\sigma = 4$. This gives a signal-noise ratio $\mathsf{SNR} = 1/8$.

Figure 4.1 shows the results on $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T}=\mathbf{t})$ obtained by Monte-Carlo simulation. We notice that

- as expected in Subsection 3.2.3, the first upper bound (3.4) is linear in q;
- as expected in Subsection 3.2.4, the second upper bound (3.5) converges to $H(K) = n = 8$.

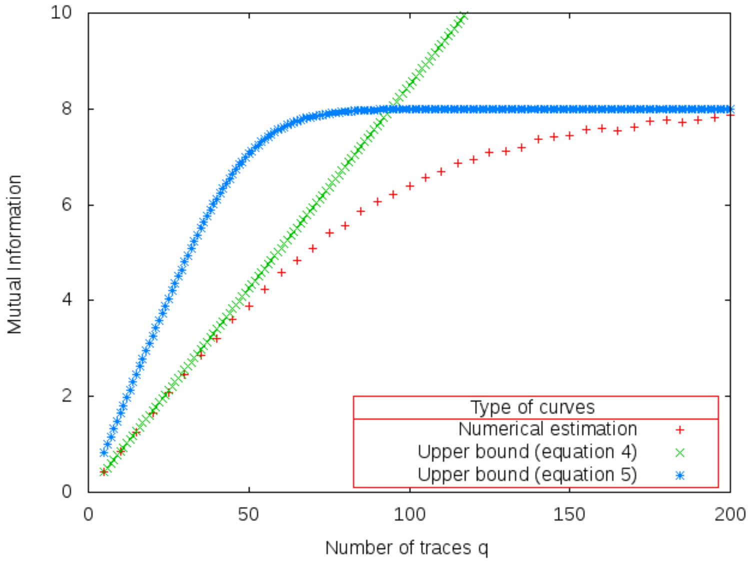


Figure 4.1: Comparison of the two upper bounds (3.4) and (3.5).

4.1.3 A Parametric Estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$

An estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ with a simple analytic expression can be obtained by a parametric estimation of the mutual information. This study is based on an empirical model that fits correctly with $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$. The information function $I(q) = I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ can be approximated by the error function such as

$$I(q) \approx n.\operatorname{erf}(q.\alpha), \tag{4.4}$$

where α is a constant, and erf the error function defined as:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t.$$

In order to verify this hypothesis numerically, for a Hamming weight leakage with additive Gaussian noise, we have plotted in Figure 4.2 the estimated parameter α for different values of σ and different number of traces. The mutual information is estimated using the law of large numbers and therefore, the parameter α is obtained by:

$$\alpha = \frac{\operatorname{erf}^{-1}(I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})/n)}{q}$$

Notice that for each value of σ, α is constant, which suggest that our empirical model fits the MI well.

We can go even further and find the analytic value of α. Indeed, the first order derivative of our model is $n\alpha\frac{2}{\sqrt{\pi}}e^{-q^2}$, therefore, the slope at the origin is $n\alpha\frac{2}{\sqrt{\pi}}$. We know that $I(0) = 0$ and $I(1) = I(X;Y \mid T) \approx \frac{1}{2}\log_2(1 + \mathsf{SNR})$. This means that if we approximate $\frac{\partial I(q)}{\partial q}(0)$ by $I(1) - I(0)$, we have:

$$\frac{1}{2}\log_2(1 + \mathsf{SNR}) = n\alpha\frac{2}{\sqrt{\pi}}, \tag{4.5}$$

and therefore,

$$\alpha = \frac{\sqrt{\pi}}{4n}\log_2(1 + \mathsf{SNR}). \tag{4.6}$$

Therefore, given the value of the SNR, one can predict the value of MI for additive Gaussian noise. We can see that the approximation (4.4) holds very well for $\sigma > 2$. This happens for low values of SNR as we encounter in practice when evaluating cryptographic devices. The number of traces needed to reach a given success rate P_s is therefore lower-bounded by:

$$q \geq \frac{4n}{\sqrt{\pi}\log_2(1 + \mathsf{SNR})} \operatorname{erf}^{-1}\Big(\frac{n - H_2(\mathrm{P}_s) - (1 - \mathrm{P}_s)\log_2(2^n - 1)}{n}\Big) \tag{4.7}$$

The interest of such bound is that it requires only the knowledge of an additive Gaussian noise and the calculation of the SNR to be exploited and to therefore predict a tight bound on the number of traces to reach a given success rate.

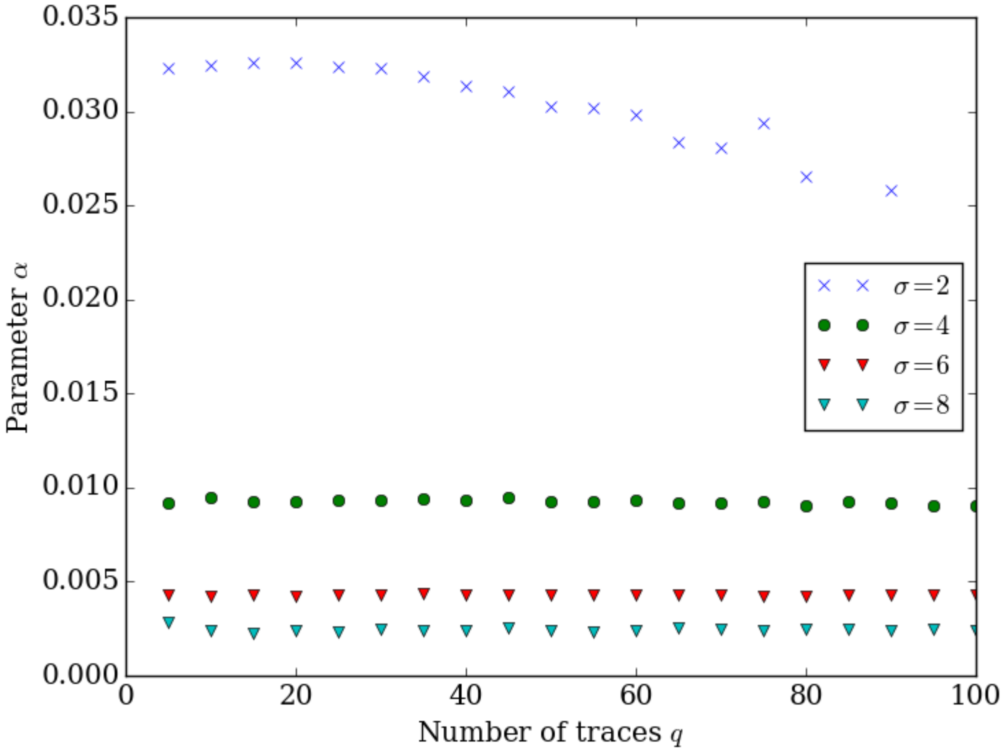


Figure 4.2: Estimation of parameter α

4.2 Application to Two Leakage Models

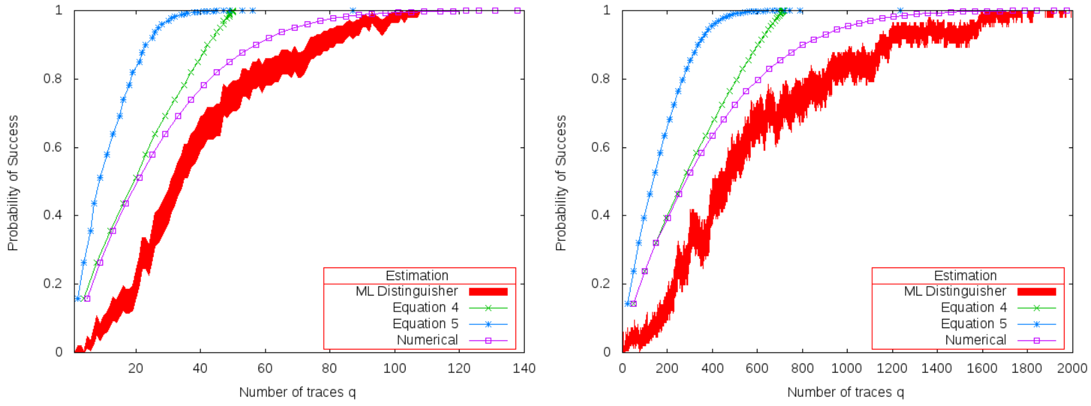
4.2.1 Example for Monobit Leakage

In this subsection, we consider a *monobit* leakage model:

$$f(t_i \oplus k) = \mathrm{LSB}(\mathrm{S_{box}}(t_i \oplus k)) \qquad (i = 1, 2, \ldots, q)$$

where $\mathrm{S_{box}}$ is the AES substitution box and LSB is the least significant bit of a number. Figure 4.3 represents the success rate of a monobit leakage with additive Gaussian noise (standard deviation $\sigma = 4$). The distinguisher used is the maximum likelihood distinguisher which is optimal [39]. The other curves are the bounds obtained with:

- a numerical estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ (using the law of large numbers, as described in Section 4.1.1);
- MI's upper bound (3.4);
- MI's upper bound (3.5).



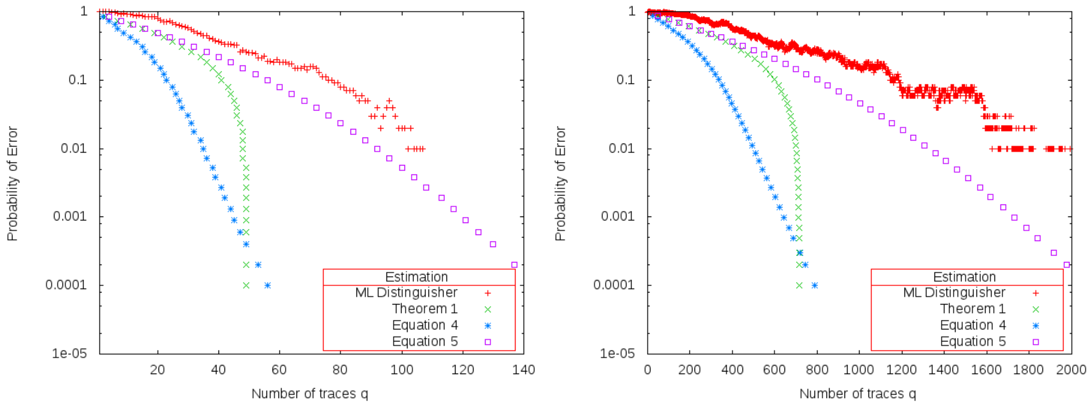
(a) $\sigma = 1$

(b) $\sigma = 4$

Figure 4.3: Success rates with monobit leakage.

The three bounds curves lie above the success rate curve as expected, the one obtained with a numerical estimation of $I(\mathbf{X};\mathbf{Y} \mid \mathbf{T})$ being the tightest (since it gives the closest approximation of the MI). The two other curves obtained with Equations (3.4) and (3.5) are not as tight but very easy to calculate. Theses results show that the better approximation of the MI we have, the closer we are from the optimal success rate.

In Figure 4.4, we have plotted the error rate in a semilog scale, so that one can observe that the curves obtained with Equations (3.4) and (3.5) actually cross each other. This shows that, closer to $\mathrm{P}_s = 1$ it is more interesting to choose the approximation of Equation (3.5), rather than Equation (3.4).



(a) $\sigma = 1$

(b) $\sigma = 4$

Figure 4.4: Error rate for a monobit leakage in a logarithmic scale

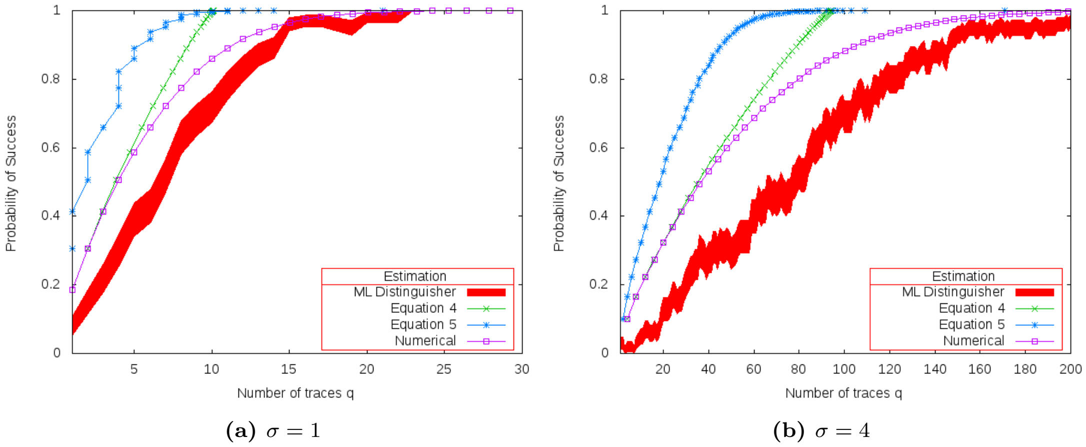
Remark 4.1. For this leakage model, with a balanced vector $\mathbf{t}$, one needs at least 8 traces to obtain 256 different vectors $\mathbf{y}$, since the function $k \mapsto \mathbf{y}(k)$ is one-to-one.

4.2.2 Example for Hamming Weight Leakage

In practice, the AES algorithms compute SubBytes with 8 bits. The leakage function are therefore different if we take this into account. Our conclusion is the same. We now consider the leakage model based on the Hamming Weight:

$$y_i = f(t_i \oplus k) = \mathrm{H_w}(\mathrm{S_{box}}(t_i \oplus k)) \qquad (i = 1, 2, \ldots, q)$$

where $\mathrm{S_{box}}$ is the AES substitution box and $\mathrm{H_w}$ is the Hamming weight function. Figure 4.5 shows the success rate compared with the three other types of estimation with an additive Gaussian noise with two values of standard deviation σ. For this model, we recall that $\mathsf{SNR} = 2/\sigma^2$. Once again, we notice that our bounds are above the optimal distinguisher and that the closest estimation of the MI gives the tightest bound.



(a) $\sigma = 1$

(b) $\sigma = 4$

Figure 4.5: Success rate for a Hamming weight leakage

4.2.3 Comparison with Duc's Bound

In order to show that our bounds are tight, we have plotted the number of traces needed to reach a success rate of 90% for a monobit leakage with additional Gaussian noise (same leakage as Section 4.2.1). In this figure, we compare our bound with the ML distinguisher and the success rate proposed by Duc et al. in [29].

To compute our bound, we only suppose that the noise is AWGN and we apply the parametric estimation of the SNR, proposed in the previous subsection (cf. Equation (4.7)).

In Figure 4.6, we notice that our bound is always very close to the real success rate, calculated for the best case for the attacker. This means that our predictions give a good idea of the security of any device, and we recall that this prediction has been made with the only knowledge of a Gaussian noise. Therefore, with very low assumptions and very few measurements (needed to calculate the SNR), we are able to predict the number of traces to reach a given success rate with a good approximation.

4.3 Practical Applications

In practice, the estimation of the SNR is therefore crucial to estimate the protection level of a device. In this section, we propose an algorithm that extracts the SNR of a leakage. Then, in order to compare our results with real world data sets, we apply our methods to that obtained within the framework of the "DPA Contest" challenge.

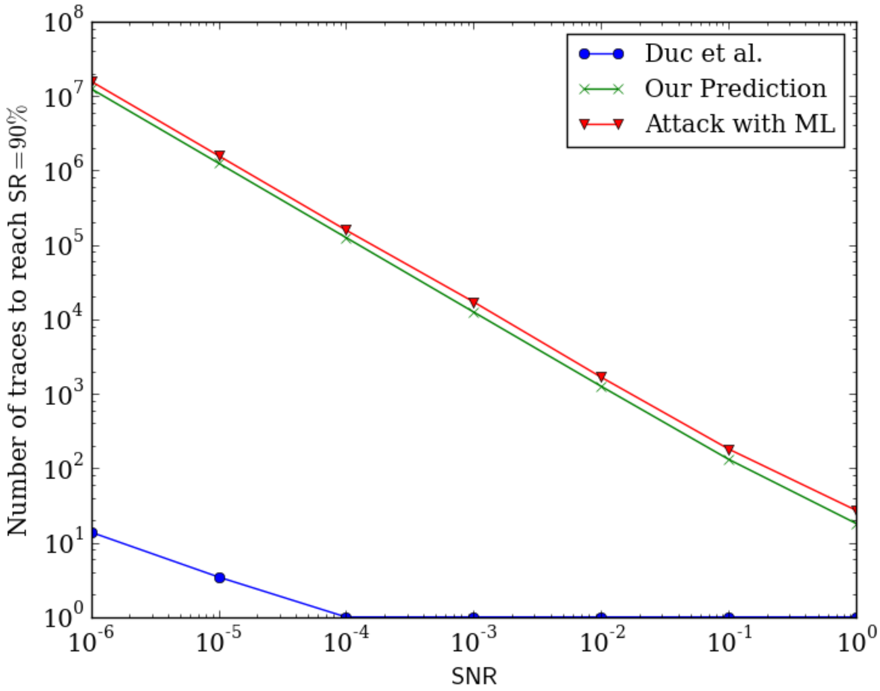


Figure 4.6: Comparison of our prediction with Duc's bound

4.3.1 The SNR estimation

In order to apply Theorem 3.1 or Equation 4.4 with the parametric estimation of the Mutual Information, one shall estimate the SNR of the leakage. When the leakage is monovariate, meaning that the attacker has at her disposal one share of the leakage, it is possible to estimate the SNR on-the-fly. The SNR of the leakage can be written as follows:

$$\begin{aligned}\mathsf{SNR} &= \frac{\mathrm{Var}(Y)}{\mathrm{Var}(N)}\\ &= \frac{\mathrm{Var}(Y)}{\mathrm{Var}(X-Y)}\\ &= \frac{\mathrm{Var}(Y)}{\mathrm{Var}(X)-\mathrm{Var}(Y)}.\end{aligned}$$

We also notice that since $X = Y + N$, where the noise N is independent from the signal Y (which depends only on the plain/cipher-text T), we have $Y = \mathbb{E}[X \mid T]$. This means that the SNR can be estimated with:

$$\mathsf{SNR} = \frac{\mathrm{Var}(\mathbb{E}[X \mid T])}{\mathrm{Var}(X) - \mathrm{Var}(\mathbb{E}[X \mid T])}. \tag{4.8}$$

This equation is valid for algorithms such as AES, since the leakage model of AES does not depend on anything else than the 8 bits of the plaintext T.

When the leakage is multivariate, it is possible to compute dimensionality reduction (c.f. [15, Corollary 4]). In such case, a profiling phase is needed to estimate the noise covariance matrix. Besides, other methods to estimate the SNR can be used such as Linear Discriminant Analysis (LDA) [81].

4.3.2 A Real World Case: the DPA Contest

In order to compare our theoretical results with practical evaluations, we used the data set of the DPA Contest v1 [84]. In the first version of this contest, the goal is to recover the 56-bit key of the DES encrypting algorithm. The device is a Side-channel Attack Standard Evaluation Board (SASEBO) developed by the Japan AIST / RCIS.

According to the data given in the DPA contest, the attacker has at her disposal a high number of traces, each made up of 20003 samples. An example is given in Fig. 4.7. We will consider here the first round of the algorithm (some attacks consider the last round but the results are very similar).

For example, we have plotted in Fig. 4.8 the SNR of this leakage considering the first substitution box. In this figure, we notice that the maximum value of the SNR is 0.144 but we

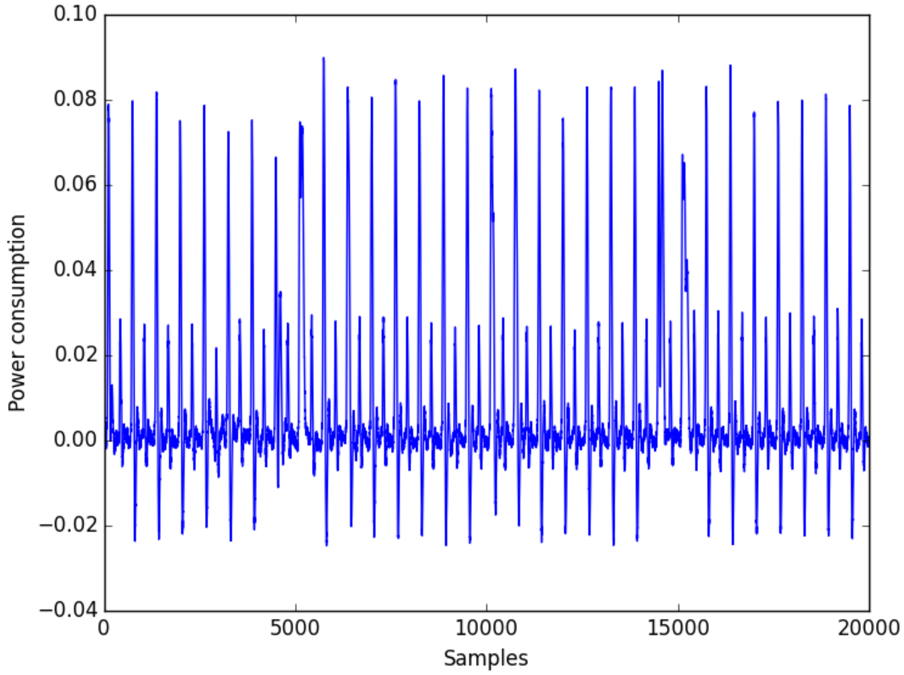


Figure 4.7: One trace of DES leakage (from DPA contest v1 [84])

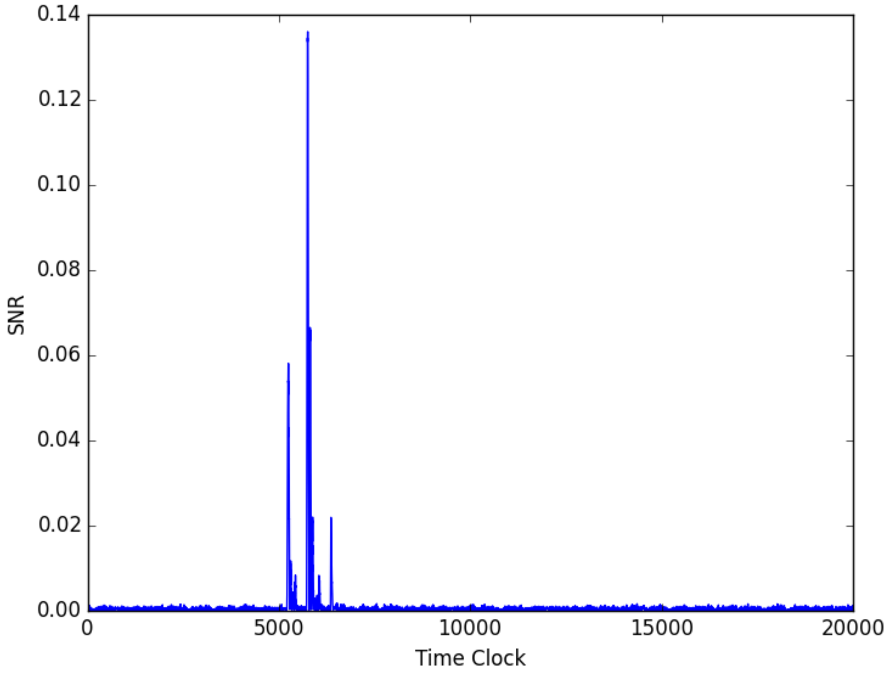


Figure 4.8: SNR of the first Sbox for the first round of DES.

Sbox #	SNR	Prediction for 99%	CPA 99%
0	0.144	112	230
1	0.077	203	350
2	0.075	208	350
3	0.071	220	450
4	0.064	243	300
5	0.151	107	190
6	0.079	198	330
7	0.136	118	270

Table 4.1: SNR for each Sbox for the DPA contest

notice that other points of interest may be used.

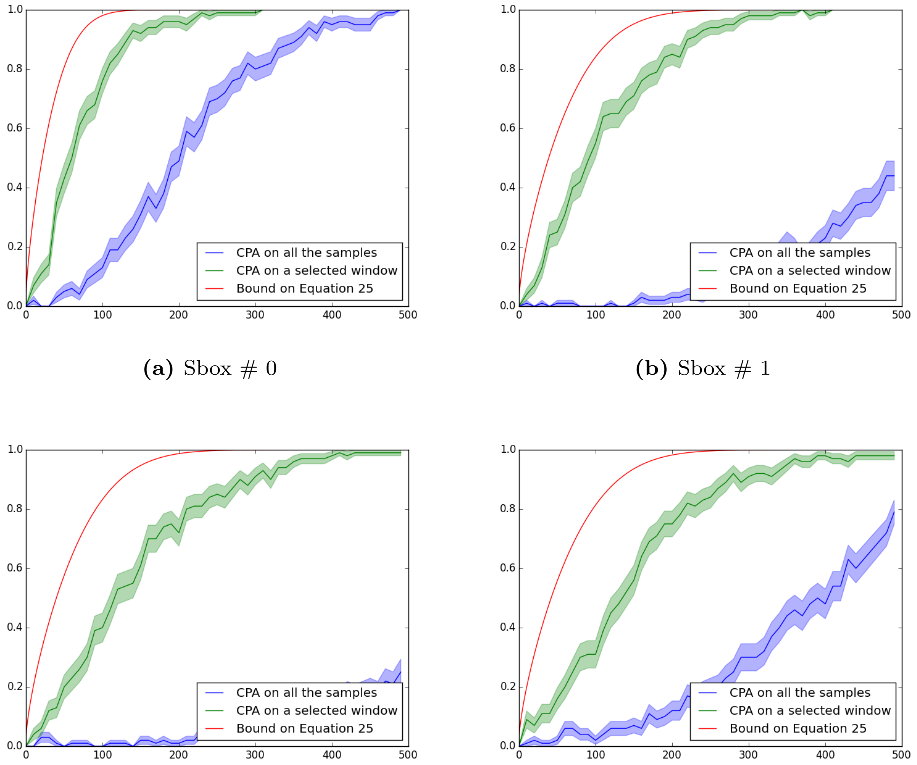
We have computed a simple CPA on the first round of DES with this data set to recover 6 bits of key. Figure 4.9 shows the partial success rate for all the substitution boxes. This success rate has been obtained with 100 experiments. We have plotted the CPA for the best time sample (the one that maximizes the SNR) in the green curve and the CPA over all the time samples (the blue curve). The red curves corresponds to the bound of Equation 4.7.

According to the figures of the table, without any pre-processing the attacker will need at least 243 traces to recover the secret key with one sample and 138 traces with two samples. This corresponds to the results obtained without pre-processing or Build-up Sub-keys.

However, in practice, other methods may help the attacker to increase the SNR of the leakage such as BS-CPA [46] where the attacker takes into account one broken subkey to recover others. For such method, the upper-bound is the best SNR i.e. 0.151 for one sample leading to 107 traces for key extraction.

4.4 Conclusion

In this chapter, we have linked two metrics used in the field of side-channel analysis: the probability of success of an attack (also known as the success rate) and the mutual information between the leaked traces and the secret key. With such links, designers will be given more precise tools to secure their cryptographic chips. Our results are of interest to better understand the different factors that impact the success rate of an attack. This is the first time that a study gives *universal* tight bounds to the success rate, in the sense that these bounds are independent of what the attacker may exploit with the measurements.



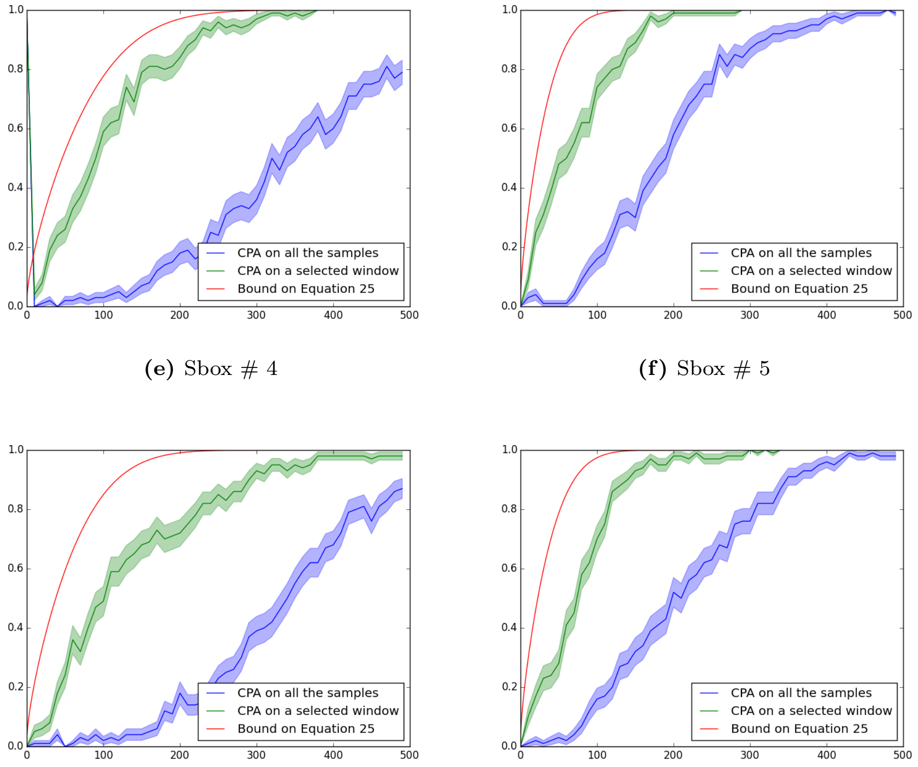
(a) Sbox # 0

(b) Sbox # 1

(c) Sbox # 2

(d) Sbox # 3

Figure 4.9: Success rate for each DES Sbox



(e) Sbox # 4

(f) Sbox # 5

(g) Sbox # 6

(h) Sbox # 7

Figure 4.8: Success rate for each DES Sbox

This is therefore a great improvement for designers. Indeed, in practice they are not able to know how their devices will be attacked in the future, but here, we allow them that to ensure the minimal security of their device in *any* adversarial context.

In addition, the link that we have made with the notion of guessing entropy gives an idea of how many attempts have to be made to recover the key after an attack.

Part III

A Mathematical Study of Distinguishers

Chapter 5

When Monobit Leakages are Defined with the Confusion Coefficient

This chapter presents the work accepted at InsCrypt 2018 conference. The conference will take place in Fuzhou, China http://xxhb.fjnu.edu.cn/inscrypt2018/.

Contents

5.1 Introduction

Today's ciphering algorithms such as AES are considered resistant to cryptanalysis. This means that the best possible way to recover a 128 bit key is about as complex as to compute an exhaustive search over the 2^{128} possibilities. With our current computational power, this is not achievable within a reasonable amount of time. However, it is possible to use plaintexts, ciphertexts, along with additional side information in order to recover the secret key of a device. Indeed, the secret key may leak via *side-channels*, such as the time to compute the algorithm, the power consumption of the device during the computation of the algorithm, or the electro-magnetic radiations of the chip.

In order to secure chips from side-channel attacks, designers have to understand how these work and what could be the future security breaches in the cryptographic algorithm as well as in the hardware computation. A preliminary step is to identify how the secret keys leak and deduce leakage models. Then, there are mathematical functions—called *distinguishers*—that take the leakage as argument and return an estimation of the secret key. They come in many flavours[1]

[1] We cover in this chapter the following distinguishers: Difference of Means or DoM [45], Correlation Power Analysis or CPA [11], Euclidean distance [39, §3], Kolmogorov-Smirnov Analysis or KSA [90], and Mutual Information Analysis or MIA [34].

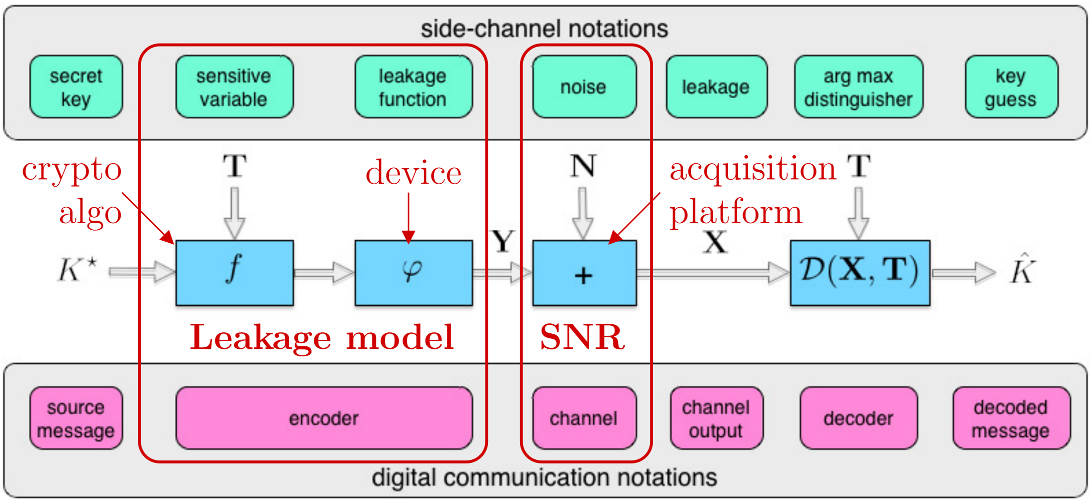


Figure 5.1: Illustration of the two parts of the side-channel analysis context (in red).

and have different figures of merit in different contexts. A given context not only involves the cryptographic algorithm and the device through the leakage model, but also the side-channel acquisition setup through the measurement characterized by its signal-to-noise ratio (SNR). This is illustrated in Fig. 5.1 borrowed from Heuser *et al.* [39] (with our annotations in red).

In practice one may encounter *monobit* leakages, meaning that the output of the leakage model can only take two values. In this case, as we shall see, the mathematical computations turn to be simpler and information theoretic tools may be used to precisely describe the link between the leakage model and the real-world leaking traces. From another perspective, considering monobit leakages can also be seen as an “abstraction” trick meant to intentionally ignore the complex effect of the way the device leaks, thereby keeping only the contribution from the cryptographic algorithm in the leakage model.

A related question is how the choice of the substitution box in the cryptographic algorithm may “help” the attacker. The standard AES substitution box was designed to be very secure against linear and differential cryptanalysis [25]. On the contrary, under side-channel analysis, the substitution box may be helpful for the attacker, especially for monobit leakages as shown below.

Related Work. Distinguishers were often studied empirically, yet such an approach does not allow for generalizations to other contexts and measurement campaigns. A theoretical approach consists in analyzing the formal expressions of the distinguishers as mathematical functions.

Fei et al. have shown that distinguishers such as DoM and CPA can be expressed in terms of a *confusion coefficient* [32]. They gave the impetus to extend this formal analysis to other types of distinguishers. In 2014, Heuser *et al.* [38] relate KSA to the confusion coefficient, and also noticed that the confusion coefficient can be related to the resistance of a substitution box against differential cryptanalysis.

Whitnall and Oswald [89] have proposed the *relative distinguishing margin* metric to compare distinguishers. However, it has been shown [73] that this metric may not be relevant in all contexts. Another way to compare distinguishers is to contrast how their success rate (SR) in key recovery depends on the number q of side-channel traces. However, even if works such as [32] and [48] were able to provide mathematically models for the SR, the comparison between different distinguishers has never been actually carried out based on such frameworks. We shall leverage instead on the so-called *success exponent* (SE) [36] which allows to compare the SR of various distinguishers based on only one exponent parameter.

Our Contributions. In this chapter we consolidate the knowledge on side-channel attacks exploiting monobit leakages. We provide a rigorous proof that any distinguisher acting on monobit leakages depends on only two parameters: the confusion coefficient and the standard deviation of the noise. Some distinguishers, namely DoM, CPA and KSA, have already been expressed as a function of those two parameters [32, 38]. In this chapter, we derive this expression for MIA and we obtain a simple analytic function when the non zero values of the confusion coefficient are near $^1/_2$ (which is the case of leakages occurring at cryptographically strong substitution boxes [18]).

Success exponents allow to characterize the efficiency (in terms of number of traces) of distinguishers to recover the key. We derive the success exponent of these distinguishers in terms of the confusion coefficient and the standard deviation of the noise. These closed-form expressions of the success exponent enable the comparison of distinguishers based only on these two parameters. The flow chart of Fig. 5.2 situates our contributions in relation to the current state of the art.

Organization. The paper is organized as follows. In Section 5.2, we recall the main definitions. In Section 5.3, we mathematically unify all the distinguishers and we show that they are only functions of two parameters. In Section 5.4, we compare the distinguishers thanks to the success exponent. Section 8.3 concludes. Appendices provide proofs for technical lemmas.

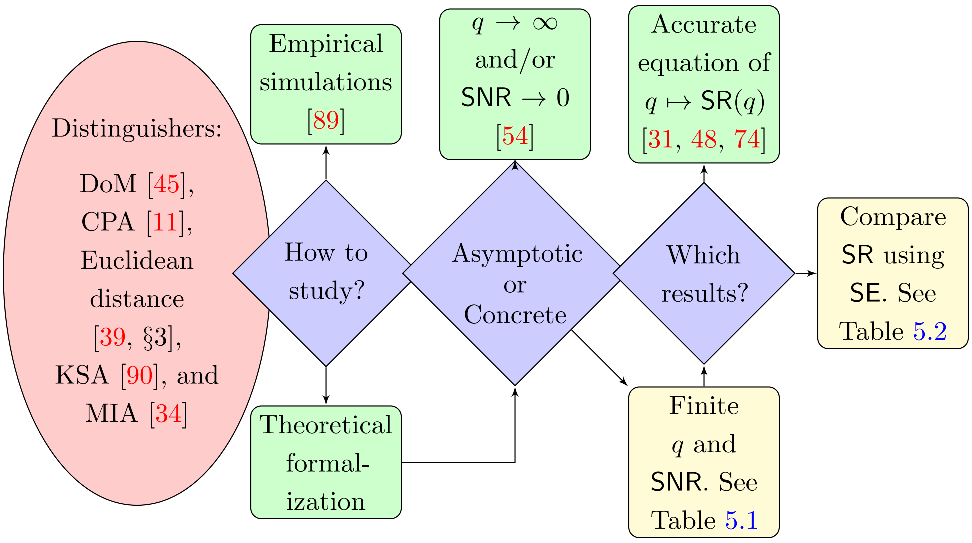

Distinguishers:
DoM [45], CPA [11], Euclidean distance [39, §3], KSA [90], and MIA [34]
How to study?
Empirical simulations [89]
Theoretical formalization
Asymptotic or Concrete
$q \to \infty$ and/or SNR $\to 0$ [54]
Which results?
Accurate equation of $q \mapsto$ SR(q) [31, 48, 74]
Finite q and SNR. See Table 5.1
Compare SR using SE. See Table 5.2

Figure 5.2: The state of the art in relation to our contributions (in yellow boxes—see also Tables 5.1 and 5.2 below).

Notations. Throughout this chapter, we use calligraphic letters to denote sets and lower-case letters for elements in this set (e.g. $x \in \mathcal{X}$). Capital letters denote random variables. For example, X is a random variable taking values in $\mathcal{X}$ and $x \in \mathcal{X}$ is a realization of X. The probability that X is x is noted $\mathbb{P}(X = x)$ or simply $\mathbb{P}(x)$ when there is no ambiguity. The expectation of a random variable is noted $\mathbb{E}[X]$ and its variance $\mathrm{Var}(X)$. The differential entropy $h(X)$ of a random variable X following distribution $p(x)$ is defined as

$$h(X) = -\int_{\mathbb{R}} p(x) \log_2 p(x) \, \mathrm{d}x. \tag{5.1}$$

The mutual information between two random variables X and Y is defined as

$$I(X;Y) = h(X) - h(X|Y) = \mathbb{E}\left[\log_2 \frac{\mathbb{P}(X,Y)}{\mathbb{P}(X)\mathbb{P}(Y)}\right]. \tag{5.2}$$

5.2 Modelization and Definitions

5.2.1 The Leakage Model

In order to compare the different distinguishers for monobit leakages, we need a leakage model upon which our computations will be based. A plaintext t meets the secret key k^* through a leakage function $f(t, k^*)$. The resulting variable $y(k^*)$ is called the *sensitive* variable. The dependence in the plaintext t will be omitted to make equations easier to read when there is no ambiguity.

The attacker measures a noisy version of $y(k^*)$ called *trace* and denoted by x. When the key is unknown, the attacker computes a sensitive variable with a key hypothesis k, that is, $y(k) = f(t, k)$. Thus our model takes the form

$$\begin{cases} y(k) = f(t,k) \\ \quad x = y(k^*) + n \end{cases} \tag{5.3}$$

where n is an independent measurement noise.

As we consider monobit leakages, we suppose that $y(k)$ can take only two values. In practice, t (resp. k) are subsets of the full plaintext (resp. key). Typically, in the case of AES where attacks can be conducted using a divide-and-conquer approach on a per substitution box basis, t and k are 8-bit works (i.e., bytes).

The above leakage model can also be written using random variables. Let T the random variable for the plaintext, $Y(k)$ for the sensitive variable, X for the measurement, and N for the Gaussian noise. We have:

$$\begin{cases} Y(k) = f(T,k) \\ \quad X = Y(k^*) + N. \end{cases} \tag{5.4}$$

In a view to simplify further mathematical computations, we suppose that the leakage random variable is reduced, that is, centered ($\mathbb{E}[Y(k)] = 0$ for all k) and of unit variance ($\mathbb{E}[Y(k)^2] = 1$ for all k). The noise is also assumed Gaussian of zero mean and its standard deviation is noted $\sigma > 0$. Moreover, we assume that for any key hypothesis the sensitive variable is *balanced*, that is, $\mathbb{P}(y(k)) = \frac{1}{2}$. Since $Y(k)$ is a binary random variable, we necessarily have that $Y(k) \in \{\pm 1\}$ in our model, and consequently the signal-to-noise ratio equals $\mathsf{SNR} = 1/\sigma^2$.

Last, we suppose that the attacker has at his disposal a number of q traces $x_1, \ldots, x_q$ obtained from leaking sensitive variables $y_1(k^*), \ldots, y_q(k^*)$ under additive noise $n_1, \ldots, n_q$.

5.2.2 The Confusion Coefficient

In the side-channel context, the confusion coefficient was defined by Fei et al. as the probability that two sensitive variables arising from two different key hypotheses are different [32, Section 3.1]. Mathematically, the confusion coefficient is written as

$$\kappa(k, k^*) = \mathbb{P}(Y(k) \neq Y(k^*)). \tag{5.5}$$

As the secret key k^* is constant and understood from the context, we can write $\kappa(k, k^*) = \kappa(k)$. Notice that in practical situations, the EIS (Equal Images under different Subkeys [77, Def. 2]) assumption holds, therefore κ is actually a function of the key bitwise XOR difference $k \oplus k^*$.

Figure 5.3 illustrates the confusion coefficient for a monobit leakage $Y(k) = \mathsf{SubBytes}(T \oplus k) \bmod 2$, where SubBytes is the AES substitution box (application $\mathbb{F}_2^8 \to \mathbb{F}_2^8$) and $\oplus$ is the bitwise exclusive or. We notice that except for $k = k^*$ (here taken $= 178$), the confusion coefficient for the AES SubBytes is close to $^1/_2$. This results from the fact the AES SubBytes has been designed to be resistant against differential cryptanalysis. Specifically, Heuser et al. [38, Proposition 6] noticed that a "good" substitution box leads to confusion coefficients near $^1/_2$.

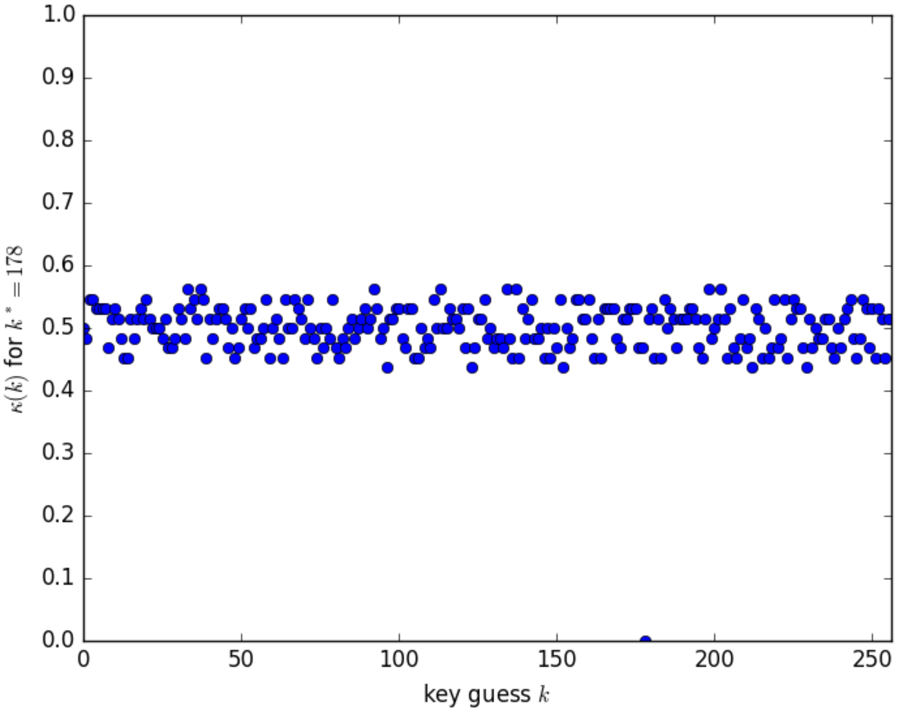


Figure 5.3: Confusion Coefficient for the AES SubBytes

The original definition of the confusion coefficient [32] considers only monobit leakages. An extension for any type of leakage was proposed in [36] where $\kappa(k)$ is defined by

$$\kappa(k) = \mathbb{E}\left[\left(\frac{Y(k^*) - Y(k)}{2}\right)^2\right]. \tag{5.6}$$

Equation (5.5) can be easily recovered from this more general expression by noting that when $Y(k)$ and $Y(k^*) \in \{\pm 1\}$, $\left(\frac{Y(k^*)-Y(k)}{2}\right)^2$ is 0 or 1 according to whether $Y(k) = Y(k^*)$ or $Y(k) \neq Y(k^*)$.

5.2.3 Distinguishers

Distinguishers aim at recovering the secret key k^* from the traces and the model. For every key k, the attacker computes the associated distinguisher. The key hypothesis that gives the highest value of the distinguisher is the estimated key. The attack is successful if the estimated key is equal to the secret key.

For every key hypothesis k, a distinguisher is noted $\widehat{\mathcal{D}}(k)$ and the estimated key is $\widehat{k} = \arg\max_k \widehat{\mathcal{D}}(k)$. Five classical distinguishers are:

- Difference of Means (DoM) [32], also known as the Differential Power Analysis (DPA) [45] where the attacker computes

$$\widehat{\mathcal{D}}(k) = \frac{\sum_{i|y_i(k)=+1} x_i}{\sum_{i|y_i(k)=+1}} - \frac{\sum_{i|y_i(k)=-1} x_i}{\sum_{i|y_i(k)=-1}}. \tag{5.7}$$

- Correlation Power Analysis (CPA) [11] where the attacker computes the absolute value of the Pearson coefficient

$$\widehat{\mathcal{D}}(k) = \left| \frac{\frac{1}{q}\sum_{i=1}^{q} x_i y_i(k) - \frac{1}{q}\sum_{i=1}^{q} x_i \cdot \frac{1}{q}\sum_{i=1}^{q} y_i(k)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y_i(k))}} \right|. \tag{5.8}$$

  Notice that $\mathrm{Var}(Y_i(k))$ do not depend on the index i, since repeated measurements are i.i.d.

- Euclidean distance, which corresponds to the Maximum Likelihood (ML) attack under the Gaussian noise hypothesis, where the attacker actually computes the negative Euclidean distance between the model and the trace

$$\widehat{\mathcal{D}}(k) = -\frac{1}{q}\sum_{i=1}^{q}(x_i - y_i(k))^2. \tag{5.9}$$

  Maximizing the value of the distinguisher amounts to minimizing the Euclidean distance. According to [39], as the noise is Gaussian and additive, the Euclidean distance is the optimal distinguishing rule (ML rule) that maximizes the success probability.

- Kolmogorov-Smirnov Analysis (KSA) [90] where the traces are used to build an estimation of the cumulative density function $\widehat{F}(x)$, and the distinguisher is

$$\widehat{\mathcal{D}}(k) = -\mathbb{E}_{Y(k)}\left[\|\widehat{F}(x|Y(k)) - \widehat{F}(x)\|_\infty\right] \tag{5.10}$$

  where the infinite norm is defined as $\|\widehat{F}(x)\|_\infty = \sup_x |\widehat{F}(x)|$. Maximizing the value of the distinguisher amounts to minimizing the expected infinite norm.

- Mutual Information Analysis (MIA) [34] where the attacker computes the mutual information between the traces and each model. The traces are used to build an estimation of the joint distribution of X and $Y(k)$, denoted by $\widehat{p}(X,Y(k))$, and with this estimation, we calculate the mutual information
$$\widehat{\mathcal{D}}(k) = \sum_{x,y(k)} \widehat{p}(x,y(k)) \log_2 \frac{\widehat{p}(x,y(k))}{\widehat{p}(x)\cdot\widehat{p}(y(k))}. \tag{5.11}$$

Given the available data, the attacker computes the distinguisher as a function of $x_1,\ldots,x_q$ and $y_1(k),\ldots,y_q(k)$. To emphasize the dependence on the data, we may write $\widehat{\mathcal{D}}(k) = \widehat{\mathcal{D}}(X_1,\ldots,X_q,Y_1(k),\ldots,Y_q(k))$. As these traces are realizations of random variables, we may also consider $\widehat{\mathcal{D}}(k)$ as a random variable which is a function of $X_1,\ldots,X_q$ and $Y_1(k),\ldots,Y_q(k)$, with expectation $\mathbb{E}[\widehat{\mathcal{D}}(k)]$ and a variance $\mathrm{Var}(\widehat{\mathcal{D}}(k))$.

When the number of queries q tends to infinity, we assume that the distinguisher converges in the mean-squared sense:

Definition 5.1 (Theoretical Distinguisher [36]). The theoretical value of the distinguisher is defined as the limit in the mean square sense when $q\to\infty$ of the distinguisher. The notation for the theoretical distinguisher is $\mathcal{D}(k)$, which is therefore implicitly defined as:
$$\mathbb{E}[(\widehat{\mathcal{D}}(k)-\mathcal{D}(k))^2] \longrightarrow 0 \text{ as } q\to\infty. \tag{5.12}$$

Put differently, $\widehat{\mathcal{D}}(k)$ can be seen as an estimator of $\mathcal{D}(k)$. An illustration of theoretical distinguishers is provided in the lower right graph of Figure 5.4. It is easily seen that as $q\to+\infty$ the distinguishers presented previously have the following theoretical distinguishers:

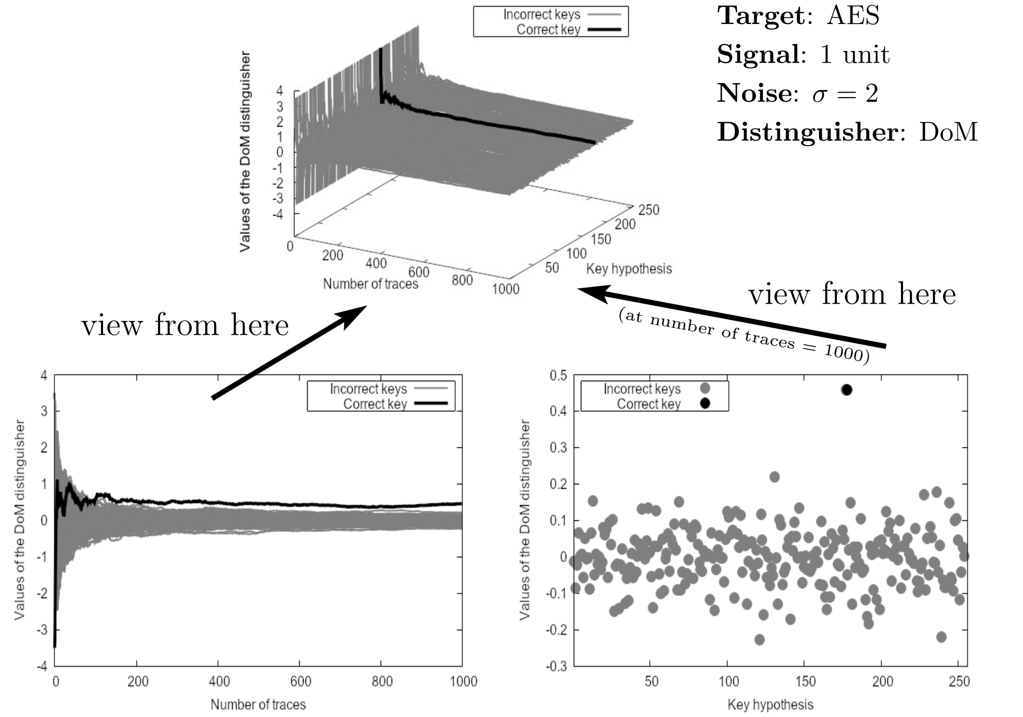
- For DoM, the theoretical distinguisher is
$$\mathcal{D}(k) = \mathbb{E}[XY(k)]. \tag{5.13}$$

- For CPA, the theoretical distinguisher is
$$\mathcal{D}(k) = \frac{\left|\mathbb{E}[XY(k)] - \mathbb{E}[X]\mathbb{E}[Y(k)]\right|}{1+\sigma^2}. \tag{5.14}$$

- For Euclidean distance (ML) distinguisher, we have:
$$\mathcal{D}(k) = -\mathbb{E}\big[(X-Y(k))^2\big]. \tag{5.15}$$

- For KSA, we have:
$$\mathcal{D}(k) = \mathbb{E}_{Y(k)}\big[\|F(x|Y(k)) - F(x)\|_\infty\big]. \tag{5.16}$$

- For MIA, it is the mutual information
$$\mathcal{D}(k) = I(X;Y(k)). \tag{5.17}$$



Target: AES
Signal: 1 unit
Noise: $\sigma = 2$
Distinguisher: DoM

view from here

view from here

(at number of traces = 1000)

Figure 5.4: Illustration of a theoretical distinguisher

5.3 Theoretical Expressions for Distinguishers

In this section, we show that all distinguishers for monobit leakages are functions of only two parameters: the confusion coefficient $\kappa(k)$ and the $\mathsf{SNR} = 1/\sigma^2$. This is confirmed by the closed-form expressions for classical distinguishers. In particular we derive the one corresponding to MIA.

5.3.1 A Communication Channel Between $Y(k)$ and $Y(k^*)$

To understand the link between any sensitive variable $Y(k)$ and the leaking sensitive variable $Y(k^*)$, consider the following information-theoretic communication channel between these two variables described in Fig. 5.5. This communication channel is simply a theoretical construction that helps explain the link between $Y(k)$ and $Y(k^*)$, which are both binary and equiprobable random variables taking their values in $\{\pm 1\}$. The parameters p and p' are the transition probabilities defined as $p = \mathbb{P}(Y(k^*) = +1|Y(k) = -1)$ and $p' = \mathbb{P}(Y(k^*) = -1|Y(k) = +1)$.

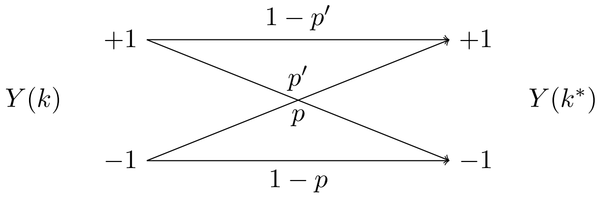


Figure 5.5: Abstract communication channel between $Y(k)$ and $Y(k^*)$

Lemma 5.1. *The communication channel defined in Fig. 5.5 is a binary symmetric channel (BSC) with transition probability equal to the confusion coefficient $\kappa(k)$.*

Proof. To prove that the channel is symmetric, we show that both transition probabilities coincide: $p = p'$. In fact, from Fig. 5.5, $\frac{1}{2} = \mathbb{P}(Y(k^*) = 1) = p\mathbb{P}(Y(k) = -1) + (1-p')\mathbb{P}(Y(k) = 1) = \frac{1}{2}(p + 1 - p')$ hence $p = p'$. Now the confusion coefficient $\kappa(k) = \mathbb{P}(Y(k) \neq Y(k^*))$ can be expanded as

$$\kappa(k) = \tfrac{1}{2}\big(\mathbb{P}(Y(k) \neq Y(k^*)|Y(k) = 1) + \mathbb{P}(Y(k) \neq Y(k^*)|Y(k) = -1)\big) \tag{5.18}$$

$$= \tfrac{1}{2}\big(\mathbb{P}(Y(k^*) = -1|Y(k) = 1) + \mathbb{P}(Y(k^*) = 1|Y(k) = -1)\big) \tag{5.19}$$

$$= \tfrac{1}{2}\big(p + p'\big) = p = p'. \tag{5.20}$$

This proves that the BSC has transition probability equal to $\kappa(k)$. □

According to a well-known information theoretic result [23, p. 187], the Shannon's *capacity* in bits per bit of this channel is

$$\mathcal{C} = 1 - H_2(\kappa(k)), \tag{5.21}$$

where $H_2(x)$ is the binary entropy function defined by

$$H_2(x) = x \log_2\Big(\frac{1}{x}\Big) + (1-x)\log_2\Big(\frac{1}{1-x}\Big). \tag{5.22}$$

This is represented in Fig. 5.6 as a function of $\kappa(k)$. Interestingly, the value $\kappa(k) = 1/2$ corresponds to null capacity while the capacity is evidently 1 bit per bit for $\kappa(k^*) = 0$, since in this case the above communication channel reduces to the identity.

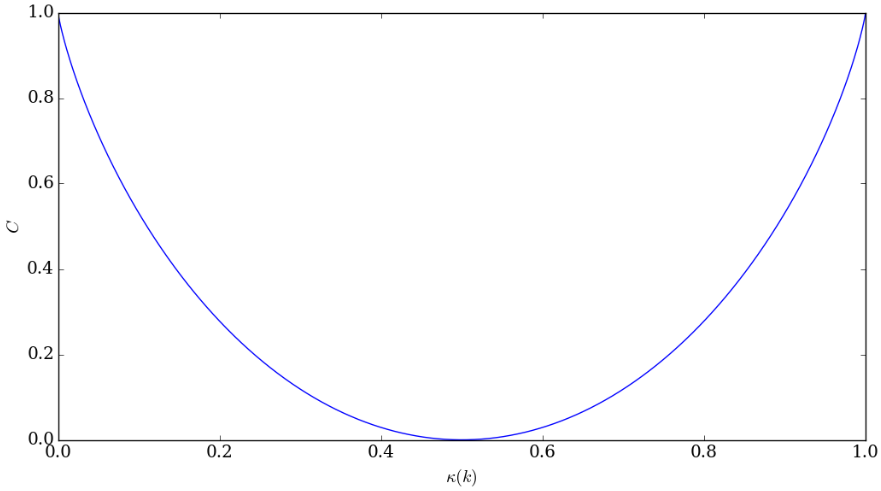


Figure 5.6: Representation of the channel capacity according to $\kappa(k)$

5.3.2 A General Result

We can now explain why all distinguishers for monobit leakages depend only on the two parameters $\kappa(k)$ and $\mathsf{SNR} = \sigma^{-2}$.

Theorem 5.1. *Any theoretical distinguisher $\mathcal{D}(k)$ for a binary leakage y can be expressed as a function of $\kappa(k)$ and σ.*

Proof. Any theoretical distinguisher is defined in terms of the joint probability distribution of X and $Y(k)$, noted $p(x, y(k))$. Now for any $x \in \mathbb{R}$ and $y(k) = \pm 1$,

$$\begin{aligned} p(x, y(k)) &= \mathbb{P}(y(k))\, p(x \mid y(k)) && (5.23)\\ &= \frac{1}{2} p(y(k^*) + n \mid y(k)) && (5.24)\\ &= \frac{1}{2} \sum_{y(k^*)} p(y(k^*) + n \mid y(k), y(k^*))\, \mathbb{P}(y(k^*) \mid y(k)) && (5.25) \end{aligned}$$

where $\mathbb{P}(y(k^*) \mid y(k))$ is the transition probability of the channel defined in Fig. 5.5. There are two possibilities. Either $y(k) = y(k^*)$, and in this case $\mathbb{P}(y(k^*)|y(k)) = 1 - \kappa(k)$, or $y(k) \neq y(k^*)$ and in this case $\mathbb{P}(y(k^*)|y(k)) = \kappa(k)$. The sum over $y(k^*)$ has two terms and both cases are represented. Moreover, the Gaussian noise is independent from every other random variable. Therefore, we have two possibilities for the joint probability:

$$p(x, y(k)) = \begin{cases} \frac{1}{2}\Big(\varphi(\frac{1+n}{\sigma})\kappa(k) + \varphi(\frac{-1+n}{\sigma})(1 - \kappa(k))\Big) \\ \frac{1}{2}\Big(\varphi(\frac{-1+n}{\sigma})\kappa(k) + \varphi(\frac{1+n}{\sigma})(1 - \kappa(k))\Big) \end{cases} \tag{5.26}$$

where $\varphi(x)$ is the probability density function of a standard normal random variable. As the noise is centered and Gaussian, the only parameter that characterizes φ is its standard deviation σ. Therefore, a joint distribution of a monobit leakage is fully characterized by σ and $\kappa(k)$. □

This proves that the knowledge of the confusion coefficient and the noise power are essential to predict the performances of the side-channel attacks for monobit leakages.

5.3.3 Classical Distinguishers as Functions of $\kappa(k)$ and σ^2

To highlight the result of section 5.3.2, we compute the classical distinguishers according to the confusion coefficient and the noise power. As we mentioned in the introduction, some of them have already been expressed according to these variables: we recall these results in Table 5.1 with references to the articles where the expression of the distinguisher in terms of $\kappa(k)$ is proven. These expressions confirm the strong link between confusion coefficient (recall Fig. 5.3) and the values of the theoretical distinguisher (for all key hypotheses, recall Fig. 5.4).

Distinguisher	**Original paper**	**Theoretical expression with $\kappa(k)$**	**Reference**
DoM	[45]	$\mathcal{D}(k) = 2(1/2 - \kappa(k))$	[52]
CPA	[11]	$\mathcal{D}(k) = 2\frac{\lvert 1/2 - \kappa(k) \rvert}{\sqrt{1+\sigma^2}}$	[52]
Euclidean distance	[39, §3]	Lemma 5.2	This chapter
KSA	[90]	$\mathcal{D}(k) = \text{erf}\left(\frac{1}{2\sigma^2}\right) \lvert 1/2 - \kappa(k) \rvert$	[38]
MIA	[34]	Lemma 5.3	This chapter

Table 5.1: Summary of classical distinguishers. Among all the classical theoretical distinguishers, we notice that the expression of the theoretical value of DoM with $\kappa(k)$ does not depend on σ.

The new results are given by the following lemmas.

Lemma 5.2. *For monobit leakages, the Euclidean distance distinguisher can be expressed as:*

$$\mathcal{D}(k) = 4({}^1\!/_2 - \kappa(k)) - (\sigma^2 + 2). \tag{5.27}$$

Proof. We have $\mathcal{D}(k) = -\mathbb{E}\big[(X - Y(k))^2\big] = -\mathbb{E}\big[(Y(k^*) - Y(k) + N)^2\big] = -\mathbb{E}\big[(Y(k^*) - Y(k))^2\big] - \sigma^2$ since the noise is independent from $Y(k^*) - Y(k)$. Then by (5.6), $\mathcal{D}(k) = -4\kappa(k) - \sigma^2 = 4({}^1\!/_2 - \kappa(k)) - 2 - \sigma^2$ where we have stressed the dependence in ${}^1\!/_2 - \kappa(k)$ as in Table 5.1. □

Lemma 5.3. *For monobit leakages, when $\kappa(k) \approx {}^1\!/_2$ for $k \neq k^*$, the MIA distinguisher can be expressed at first order as:*

$$\mathcal{D}(k) = 2\log_2(e)(\kappa(k) - {}^1\!/_2)^2 g(\sigma) \tag{5.28}$$

where

$$g(\sigma) = \frac{1}{2}\mathbb{E}\Big[\tanh^2\Big(\frac{Z}{\sigma} + \frac{1}{\sigma^2}\Big) + \tanh^2\Big(\frac{Z}{\sigma} - \frac{1}{\sigma^2}\Big)\Big] \tag{5.29}$$

and $Z \sim \mathcal{N}(0, 1)$. The function g satisfies

$$\lim_{\sigma \to 0} g(\sigma) = 1 \qquad \text{and} \qquad \lim_{\sigma \to \infty} \sigma^2 \times g(\sigma) = 1. \tag{5.30}$$

Proof. See Appendix B.1. □

Figure 5.7 plots the shape of $g(\sigma)$ which tends to 1 when $\sigma \to 0$ and is equivalent to $\frac{1}{\sigma^2}$ when $\sigma \to \infty$.

When $k = k^*$ the MIA distinguisher also has a simple expression since it reduces to the known expression of the channel capacity for channels with binary input and additive Gaussian noise [9, p. 274]:

$$\mathcal{D}(k^*) = \frac{1}{\sigma^2} - \int_{\mathbb{R}} \frac{e^{-\frac{1}{2}y^2}}{2\pi} \log_2 \cosh\Big(\frac{1}{\sigma^2} - \frac{y}{\sigma^2}\Big) \mathrm{d}y. \tag{5.31}$$

Remark 5.1. With respect to their theoretical distinguishers, DoM is in bijection with the Euclidean distance, and CPA is in bijection with KSA. Indeed, the Euclidean distance is $\mathcal{D}(k) = 4({}^1\!/_2 - \kappa(k)) - 2 - \sigma^2$ and σ is independent from the choice of the key. Therefore, there is a bijection between $4({}^1\!/_2 - \kappa(k)) - 2 - \sigma^2$ and $2({}^1\!/_2 - \kappa(k))$ which is the theoretical value of DoM. Regarding CPA and KSA, both distinguishers are functions of $|{}^1\!/_2 - \kappa(k)|$.

We also notice that MIA is in bijection with CPA (and therefore KSA). Indeed, according to the value of MIA with $\kappa(k)$, the distinguisher is a function of $({}^1\!/_2 - \kappa(k))^2$ which is in bijection with $|{}^1\!/_2 - \kappa(k)| = \sqrt{({}^1\!/_2 - \kappa(k))^2}$. This means that for monobit leakages, any attack that works with one of these distinguishers will also work with another, and *vice versa*.

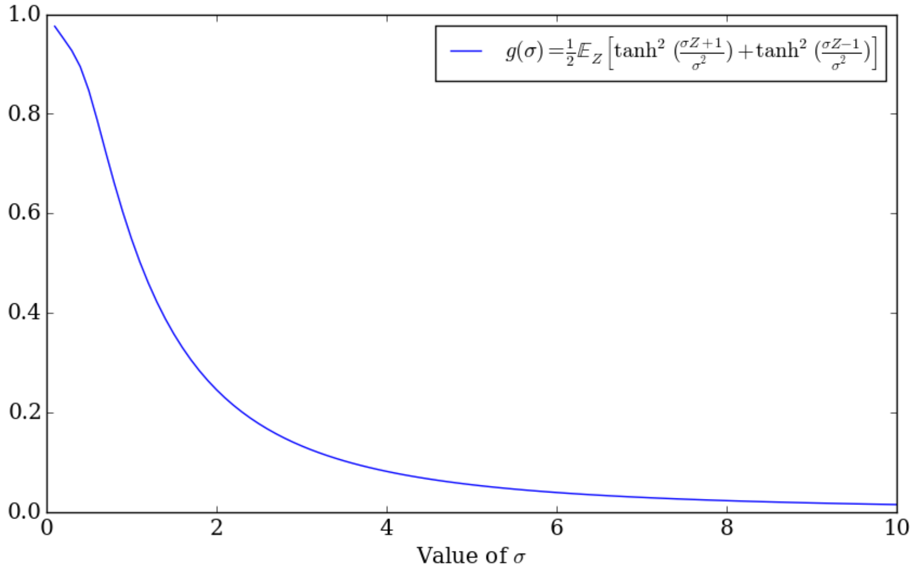


Figure 5.7: Representation of $g(\sigma)$

5.4 Comparing Distinguishers with the Success Exponent

5.4.1 Mathematical Expression of SE

In the previous section, we have computed the theoretical values of the classical distinguishers in terms of $\kappa(k)$ and σ. Now, we wish to compare their success rate. As we mentioned Section 5.2.3, the attacker computes the estimated distinguisher $\widehat{\mathfrak{D}}(k)$ to recover the secret key. This is the main reason why all distinguishers do not perform equally in key recovery; indeed, they do not converge at the same speed towards their theoretical value.

In order to compare them, we have computed their *success exponent*, a metric proposed by Guilley *et al.* in [36] that evaluates how fast the success rate of a distinguisher converges to 100%. With a Gaussian assumption, they prove that the success rate can be modeled as

$$\mathsf{SR} = 1 - \exp(-q \times \mathsf{SE}), \tag{5.32}$$

where q is the number of traces and $\mathsf{SE} \in \mathbb{R}^+$ is the so-called success exponent. Therefore, the greater the success exponent is, the faster the convergence of the success rate.

We present the theoretical values of the success exponent for the different distinguishers in Table 5.2. As a direct consequence of Theorem 5.1, all of these success exponents are function

Distinguisher	**Closed form SE with $\kappa(k)$ and σ**	**Reference**	**Numerical value for AES $\mathsf{SubBytes}$**
DoM	$\frac{1}{2}\min_{k\neq k^*}\frac{\kappa(k)}{1+\sigma^2-\kappa(k)}$	[36, Proposition 4]	3.39×10^{-3}
CPA	Lemma 5.4	This chapter	3.39×10^{-3}
Euclidean distance	$\frac{1}{2}\min_{k\neq k^*}\frac{\kappa(k)}{1+\sigma^2-\kappa(k)}$	[36, Proposition 5]	3.39×10^{-3}
KSA	Lemma 5.5	This chapter	1.08×10^{-3}
MIA	Lemma 5.6	[36, Proposition 6]	8.52×10^{-5}

Table 5.2: Success exponents for the classical distinguishers. The numerical values of SE are obtained for AES $\mathsf{SubBytes}$ least significant bit leakage model and noise of standard deviation $\sigma = 4$. Notice that in the monobit case, Euclidean distance and DoM have strictly the same success rate because $-(X-Y(k))^2 = -X^2 + 2XY(k) - 1$, and X^2 is independent of the choice of the key.

of $\kappa(k)$ and σ. Therefore, if the attacker only knows the type of substitution box that is used and the SNR of the leakage, he can predict how fast he recovers the secret key.

Lemma 5.4 (Success exponent of CPA). *The success exponent of CPA*[1] *is:*

$$\mathsf{SE} = \frac{1}{2}\min_{k\neq k^*}\frac{1-2|1/2-\kappa(k)|}{1+2\sigma^2+2|1/2-\kappa(k)|}. \tag{5.33}$$

Proof. See Appendix B.2. □

Lemma 5.5 (Success exponent of KSA). *Assuming that the distributions are estimated with the kernel method using Heaviside step function, the success exponent of KSA is*

$$\mathsf{SE} = \frac{1}{2}\min_{k\neq k^*}\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1/2-|1/2-\kappa(k)|)}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1/2-|1/2-\kappa(k)|)}. \tag{5.34}$$

Proof. See Appendix B.3. □

Lemma 5.6 (Success exponent of MIA). *When $\sigma \gg 1$, the success exponent for an MIA computed with histograms is*

$$\mathsf{SE} = \frac{4\log_2(e)^2}{\sigma^4}\min_{k\neq k^*}\kappa(k)^2(1-\kappa(k))^2. \tag{5.35}$$

Proof. See Appendix B.4. □

In order to validate our theoretical results, we have simulated attacks within the monobit model presented in Sec. 5.2. The success rates of these attacks are presented in Fig. 5.8. In

[1] In [36], CPA is treated as a distinguisher, but without the absolute values. Those remove false positives which occur in monobit leakages when there are anti-correlations. Our value of the success exponent is, therefore, different from theirs.

this figure, we notice that, as expected, the Euclidean distance (ML) is the best distinguisher, closely followed by CPA. Both have similar same success rate. The small difference is due to the use the the absolute values in the distinguishing function of CPA (see discussion in Remark 9 of [39]). The KSA is requiring a bit less than the double of traces, compared to Euclidean distance, DoM and CPA. The MIA performs really bad compared to the other distinguishers. Error bars represent the inaccuracy while estimating the SR (here, we ran 100 simulations).

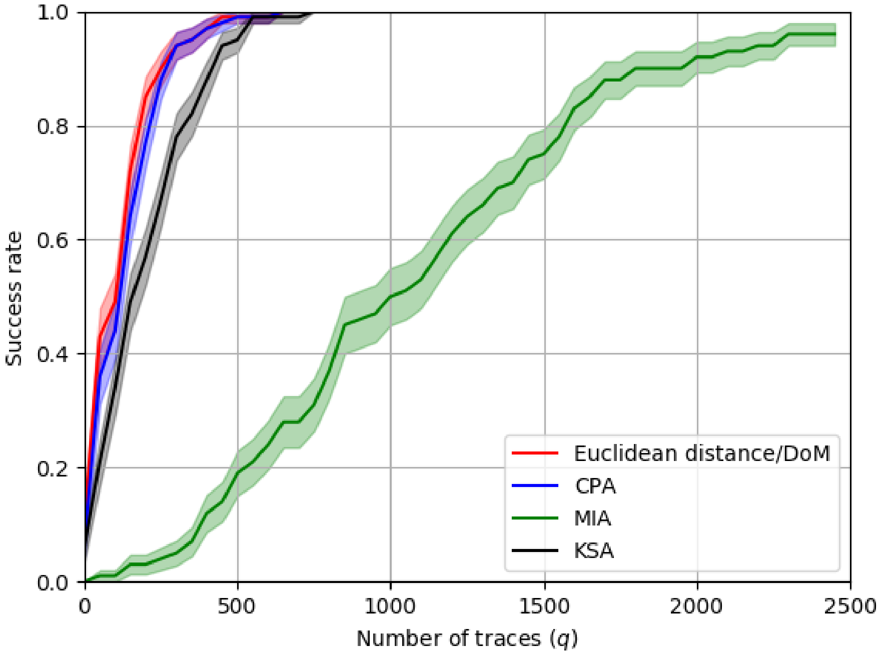


Figure 5.8: Success rate for classical distinguishers ($\sigma = 4$)

These simulations are therefore in complete coherence with the theoretical results of Table 5.2. Indeed, the order of the distinguishers is the same w.r.t. the success rate and w.r.t. the success exponent. In addition, according to the definition of the success exponent SE in (5.32), the number of traces q to reach a given success rate (e.g., SR $= 80\%$) is proportional to the inverse of SE. This quantitative law is satisfied in the simulation of Fig. 5.8. For an accurate validation, we have plotted in Fig. 5.9 the success exponent vs the success rate, and indeed points are aligned.

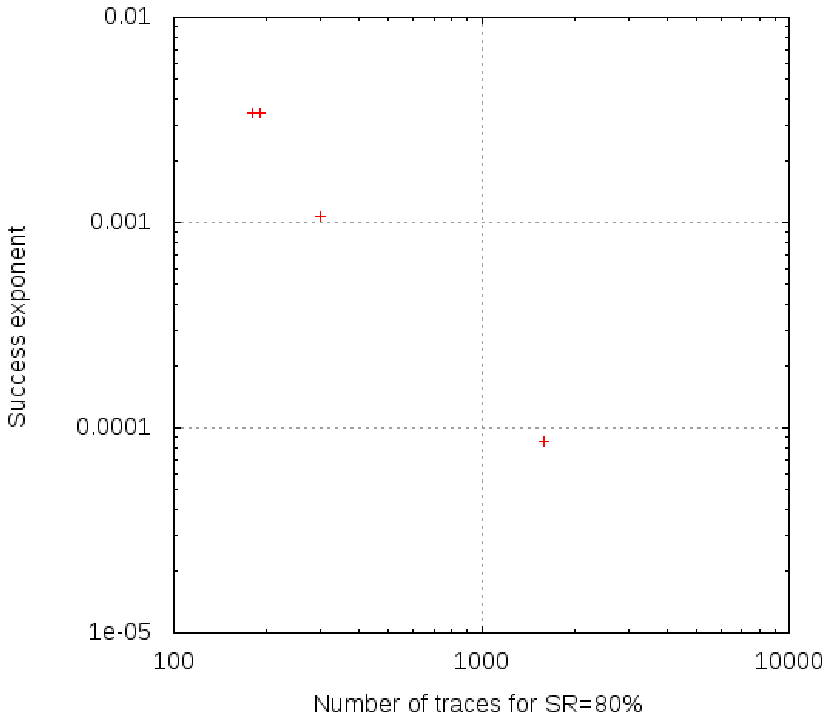


Figure 5.9: Success rate versus success exponent

5.4.2 Comparison of Distinguishers Based on their Success Exponent

With the theoretical expressions of the Success Exponent, it is now possible to rank distinguishers for a given value of σ and a specific set of confusion coefficients $(\kappa(k))_{k\neq k^*}$.

We first show that for all of the distinguishers presented in Table 5.2, the key that minimizes the expression of the success exponent is either the one that minimizes $\kappa(k)$ or the one that maximizes $\kappa(k)$ (for $k \neq k^*$). Indeed, there are only two values of $\kappa(k)$, $k \neq k^*$, which are relevant for the comparison of distinguishers presented in Table 5.2.

Lemma 5.7. *For all value of σ, the value of $k \neq k^*$ which minimizes the formal expressions of monobit distinguishers is either*

$$k_{min} = \operatorname*{argmin}_{k\neq k^*} \kappa(k) \qquad \textit{(nearest rival)}$$

or

$$k_{max} = \operatorname*{argmax}_{k\neq k^*} \kappa(k) \qquad \textit{(furthest rival).}$$

Proof. For each distinguisher, let us replace $\kappa(k)$ by a real-valued variable $x \in [0,1]$ in the formal expression of the theoretical distinguishers given in the second column of Tab. 5.2. If the expression which is a function of x is increasing, then its minimum value over $\{\kappa(k), k \neq k^*\} \subset [0,1]$ is $\kappa_{\min} = \kappa(k_{\min})$. Symmetrically, if the function of x is decreasing, than its minimum value over $\{\kappa(k), k \neq k^*\} \subset [0,1]$ is $\kappa_{\max} = \kappa(k_{\max})$. The argument of the minimum operator is always either strictly increasing or decreasing with x. Indeed, we take the derivative of each function and we notice that it is always positive for any value of $x \in [0,1]$:

- For DoM and Euclidean distance: $\frac{\partial}{\partial x}\frac{1}{2}\frac{x}{1+\sigma^2-x} = \frac{1}{2}\frac{1+\sigma^2}{(1+\sigma^2-x)^2} > 0$.
- For CPA, we distinguish to cases. Either, the minimum is reached for a value k_0 such that $(1/2 - \kappa(k_0))$ is greater than 0, and in this case, the value of the success exponent is equal to the Success Exponent of DoM (and thus $k_0 = \kappa_{\min}$), or $(1/2 - \kappa(k_0))$ is lower than 0. In this last case, the value of the Success Exponent is $\frac{1}{2}\frac{1-\kappa(k)}{\sigma^2+\kappa(k)}$. The derivative of this function is $\frac{\partial}{\partial x}\frac{1}{2}\frac{1-x}{\sigma^2+x} = -\frac{\sigma^2+1}{(\sigma^2+x)^2} < 0$. This means that the higher x is, the smaller the success exponent is. Hence, $k_0 = \kappa_{\max}$.
- For KSA, the computation is similar to the case of CPA. Either the value which minimizes the expression in Lemma 5.5 is k_0, such that $(1/2 - \kappa(k_0))$ is greater than 0, in which case the Success Exponent is equal to $\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2}\frac{\kappa(k_0)}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2\kappa(k_0)}$, or $(1/2 - \kappa(k_0))$ is negative, in which case the Success Exponent is equal to $\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2}\frac{1-\kappa(k_0)}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1-\kappa(k_0))}$. Now, $\frac{\partial}{\partial x}\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2}\frac{x}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2 x} = \frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{(2-x)^2} > 0$, hence $\kappa(k_0) = \kappa_{\min}$, and $\frac{\partial}{\partial x}\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2}\frac{1-x}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1-x)} = -\frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{(2-x)^2} < 0$, hence $\kappa(k_0) = \kappa_{\max}$.

- For MIA, $\frac{\partial}{\partial x}x^2(1-x)^2 = 4x(1-x)(\frac{1}{2}-x)$, which has the sign of $\frac{1}{2}-x$. As $0 \leq x \leq 1$, the Success Exponent for MIA is thus increasing on $[0, {}^1\!/_2]$ and decreasing on $[{}^1\!/_2, 1]$. Thus, the minimum value of the Success Exponent is either occurring for $\kappa_{\min}$ (if $\kappa_{\min} < 1-\kappa_{\max}$) or for $\kappa_{\max}$.

□

This means that the wrong key k that determines the minimum value in the closed form of the Success Exponent in the expressions from Tab. 5.2:

- (regarding DoM and Euclidean distance): is also the one which happens to minimize $\kappa(k)$, $k \neq k^*$, i.e., the nearest rival of the correct key k^*;
- (regarding CPA, KSA and MIA): is also the one which corresponds to the nearest confusion coefficient from its bounds (either 0 or 1).

Lemma 5.7 validates the role of Relative Distinguishing Margin (RDM [89]) metric while comparing distinguishers. Indeed, the distinguishers DoM and Euclidean distance (resp. CPA, and at first order, KSA and MIA) only consider the nearest rivals (resp. nearest or furthest) in terms of confusion coefficient. This is general property of the distinguisher, since it does not depend on the noise variance σ^2.

However, we will highlight other factors that determine the efficiency in terms of "data-complexity" of the classical distinguishers. To illustrate Lemma 5.7, we have reported in Table 5.3 the 32 $\kappa_{\min}$ and $\kappa_{\max}$ for the 32 possible fanout bits of Data Encryption Standard (DES) substitution boxes (sboxes). These values correspond to the 8 sboxes in DES multiplied by the 4 bits of the output of the LUT. In this table, we notice that each sbox has a particular behaviour. The value which determines the success exponent represented with a grey background color. It is interesting to see that CPA, KSA and MIA are actually limited by $\kappa_{\max}$ most of the time (i.e., $1-\kappa_{\max} < \kappa_{\min}$, for about 89% of the bits, excluding ties).

Remark 5.2. It was previously unnoticed that, in the case of DES, distinguishers with "absolute values", such as CPA, were better than without the absolute values.

Regarding AES, we notice that for the AES SubBytes function, the values of $\kappa_{\min}$ and $\kappa_{\max}$ are always the same regardless to the leaking bit. Moreover, we also notice that $\kappa_{\min} = 1-\kappa_{\max}$. Indeed, for all output bits of SubBytes, we have:

- $\kappa_{\min} = 0.4375$
- $\kappa_{\max} = 0.5625$.

Element	$\kappa_{\min}$	$\kappa_{\max}$
Bit 1 – Sbox 1	0.2500	0.7500
Bit 2 – Sbox 1	0.1875	0.7500
Bit 3 – Sbox 1	0.3125	0.6875
Bit 4 – Sbox 1	0.2500	0.7500
Bit 1 – Sbox 2	0.2500	0.8125
Bit 2 – Sbox 2	0.3125	0.7500
Bit 3 – Sbox 2	0.1875	0.9375
Bit 4 – Sbox 2	0.2500	0.8125
Bit 1 – Sbox 3	0.2500	0.8750
Bit 2 – Sbox 3	0.3125	0.7500
Bit 3 – Sbox 3	0.2500	0.8750
Bit 4 – Sbox 3	0.2500	0.8125
Bit 1 – Sbox 4	0.3125	0.6875
Bit 2 – Sbox 4	0.3125	0.6875
Bit 3 – Sbox 4	0.3125	0.6875
Bit 4 – Sbox 4	0.3125	0.6875
Bit 1 – Sbox 5	0.3750	0.6875
Bit 2 – Sbox 5	0.3125	0.7500
Bit 3 – Sbox 5	0.3125	0.8125
Bit 4 – Sbox 5	0.2500	0.7500
Bit 1 – Sbox 6	0.2500	0.8125
Bit 2 – Sbox 6	0.3125	0.8125
Bit 3 – Sbox 6	0.2500	0.7500
Bit 4 – Sbox 6	0.1250	0.7500
Bit 1 – Sbox 7	0.2500	0.8750
Bit 2 – Sbox 7	0.2500	0.7500
Bit 3 – Sbox 7	0.1875	0.8750
Bit 4 – Sbox 7	0.2500	0.7500
Bit 1 – Sbox 8	0.3125	0.8125
Bit 2 – Sbox 8	0.2500	0.8125
Bit 3 – Sbox 8	0.2500	0.8125
Bit 4 – Sbox 8	0.2500	0.7500

Table 5.3: Numerical values of $\kappa_{\min}$ and $\kappa_{\max}$ for DES

Lemma 5.8 (CPA correlating positively or negatively). *The CPA correlates negatively (i.e., $|1/2 - \kappa(k)|$ is minimum for $k \neq k^*$ when $1/2 - \kappa$ is positive) when $\kappa_{min} < 1 - \kappa_{max}$. And vice-versa.*

Proof. It is easy to check that

$$\begin{aligned}\frac{1}{2}\frac{\kappa_{\min}}{1+\sigma^2-\kappa_{\min}} < \frac{1}{2}\frac{1-\kappa_{\max}}{\sigma^2+\kappa_{\max}} &\iff \kappa_{\min}(\sigma^2+\kappa_{\max}) < (1+\sigma^2-\kappa_{\min})(1-\kappa_{\max}) \\ &\iff \kappa_{\min} < 1-\kappa_{\max}.\end{aligned}$$

□

Lemma 5.9 (KSA expression of SE). *The expression of the Success Exponent for KSA (for large values of σ) is:*

$$\begin{cases} \frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2} \frac{\kappa_{min}}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2\kappa_{min}} & \textit{if } \kappa_{min} < 1-\kappa_{max}, \\ \frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2} \frac{1-\kappa_{max}}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1-\kappa_{max})} & \textit{if } \kappa_{min} > 1-\kappa_{max}. \end{cases}$$

Proof. It is a direct consequence of the fact the function $x \in [0,1] \mapsto \frac{\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2}{2} \frac{x}{2-\operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2 x}$ is increasing (recall Lemma 5.7). □

Lemma 5.10 (MIA expression of SE). *The expression of the Success Exponent for MIA (for large values of σ) is:*

$$\begin{cases} \frac{4\log_2(e)^2}{\sigma^4}\kappa_{min}^2(1-\kappa_{min})^2 & \textit{if } \kappa_{min} < 1-\kappa_{max}, \\ \frac{4\log_2(e)^2}{\sigma^4}\kappa_{max}^2(1-\kappa_{max})^2 & \textit{if } \kappa_{min} > 1-\kappa_{max}. \end{cases}$$

Proof. We have:

$$\begin{aligned}\frac{4\log_2(e)^2}{\sigma^4}\kappa_{\min}^2(1-\kappa_{\min})^2 < \frac{4\log_2(e)^2}{\sigma^4}\kappa_{\max}^2(1-\kappa_{\max})^2 &\iff \kappa_{\min}(1-\kappa_{\min}) < \kappa_{\max}(1-\kappa_{\max}) \\ &\iff (\kappa_{\max}-\kappa_{\min})\left[\kappa_{\max}+\kappa_{\min}-1\right] < 0 \\ &\iff \kappa_{\min} < 1-\kappa_{\max}.\end{aligned}$$

□

Corollary 5.1 (Revised expressions of the Success Expoennt for the 5 distinguishers). *With κ_{min} and κ_{max} defined in Lemma 5.7, the success exponents of the 5 distinguishers are written in Table 5.4. For CPA, KSA and MIA, there are two expressions depending whether $\kappa_{min} \lesseqgtr \kappa_{max}$.*

Now, we can proceed to compare distinguishers:

Proposition 5.1 (DoM is always better than CPA). *For any value of σ, the success exponent of DoM is always greater than the success exponent of CPA.*

Condition Distinguisher	$\kappa_{\min} < 1 - \kappa_{\max}$	$\kappa_{\min} > 1 - \kappa_{\max}$
DoM	$\frac{1}{2}\frac{\kappa_{\min}}{1+\sigma^2-\kappa_{\min}}$	
CPA	$\frac{1}{2}\frac{\kappa_{\min}}{1+\sigma^2-\kappa_{\min}}$	$\frac{1}{2}\frac{1-\kappa_{\max}}{\sigma^2+\kappa_{\max}}$
Euclidean distance	$\frac{1}{2}\frac{\kappa_{\min}}{1+\sigma^2-\kappa_{\min}}$	
KSA	$\frac{1}{2}\frac{\mathrm{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2\kappa_{\min}}{2-\mathrm{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2\kappa_{\min}}$	$\frac{1}{2}\frac{\mathrm{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1-\kappa_{\max})}{2-\mathrm{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)^2(1-\kappa_{\max})}$
MIA	$\frac{4\log_2(e)^2}{\sigma^4}\kappa_{\min}^2(1-\kappa_{\min})^2$	$\frac{4\log_2(e)^2}{\sigma^4}\kappa_{\max}^2(1-\kappa_{\max})^2$

Table 5.4: Expression of the Success Exponent for the 5 studied distinguishers

Proof. The expression of CPA is either with $\kappa_{\min}$ or with $\kappa_{\max}$. We therefore have two cases:

- If $\kappa_{\min} < 1 - \kappa_{\max}$, then the expression of CPA is the same as the expression of DoM (cf. Table 5.4). Therefore, the success exponents of the two distinguishers are the same in this case.
- In the other case, the expression of the success exponent is $\frac{1}{2}\frac{1-\kappa_{\max}}{\sigma^2+\kappa_{\max}}$. However, as $1 - \kappa_{\max} < \kappa_{\min}$, we have $\frac{1}{2}\frac{1-\kappa_{\max}}{\sigma^2+\kappa_{\max}} < \frac{\kappa_{\min}}{1+\sigma^2-\kappa_{\min}}$, which is the expression of DoM. Therefore, in this case, the success exponent of CPA is smaller than the success exponent of DoM.

Overall, DoM is therefore a better distinguisher than CPA in terms of success exponent. □

Proposition 5.2 (CPA vs KSA)**.** *When $\sigma \gg 1$ the success exponent of CPA is always higher than the success exponent of KSA.*

Proof. We consider that $\sigma \gg 1$. This means that the success exponent of CPA (we consider the formula with $\kappa_{\min}$) is equivalent to $\frac{1}{2}\frac{\kappa_{\min}}{\sigma^2}$. For KSA, the success exponent is equivalent to $\frac{1}{2}\frac{\kappa_{\min}}{\pi\sigma^2}$. Indeed, when $\sigma \gg 1$, $\left(\frac{1}{\sqrt{2}\sigma}\right)^2$ is equivalent to $\frac{2}{\pi\sigma^2}$.

Therefore, when $\sigma \gg 1$, the success exponent of CPA is always higher than the success exponent of KSA. The calculations are the same if we consider $\kappa_{\max}$. □

Proposition 5.3 (MIA vs DoM)**.** *For $\sigma > 1$, the success exponent of MIA is always smaller than the success exponent of CPA.*

Proof. For DoM, the expression of the success exponent is proportional to $\frac{1}{\sigma^2}$ while the expression of the success exponent for MIA is proportional to $\frac{1}{\sigma^4}$. □

To highlight these lemmas, we have plotted in Figure 5.10 the success exponents obtained for every distinguisher with respect to the value of σ. We notice that the order obtained in the previous lemmas is verified[1] In this case, the value of $\kappa_{\min}$ is lower than the value of $1 - \kappa_{\max}$. Therefore, the Success Exponent based on the value $\kappa_{\min}$ will be used.

[1] We did not plot for values of σ lower than 5, since the lemmas are true for large values on σ.

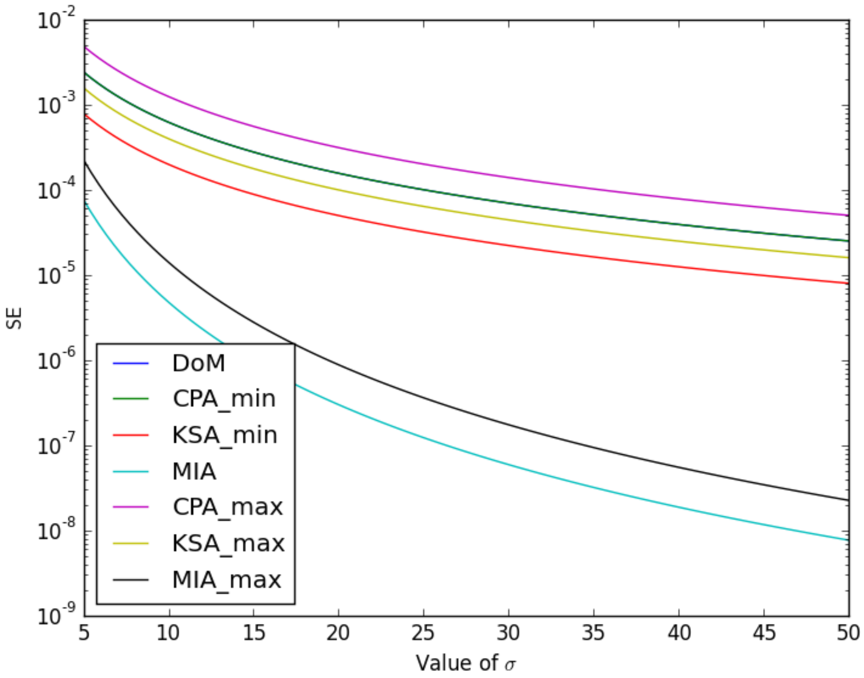


Figure 5.10: Success Exponent for DES ($\kappa_{\min} = 0.125$, $\kappa_{\max} = 0.75$)

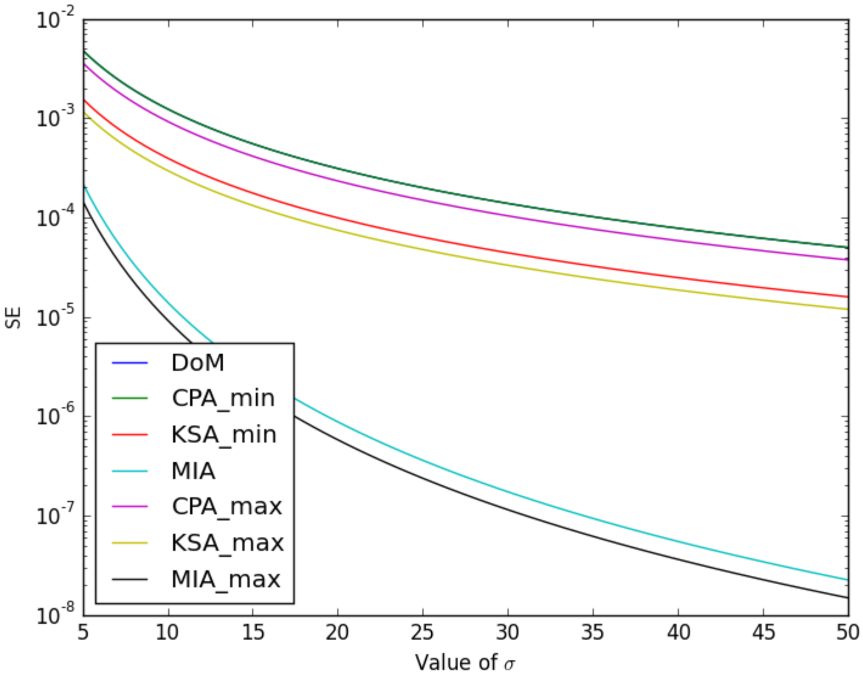


Figure 5.11: Success Exponent for DES ($\kappa_{\min} = 0.25$, $\kappa_{\max} = 0.8125$)

On the contrary, we have plotted in Figure 5.11 the Success Exponents for values of $\kappa_{\min}$ and $\kappa_{\max}$ such that $\kappa_{\min} > 1 - \kappa_{\max}$. In this case, the Success Exponents of CPA, KSA and MIA is based on the value of $\kappa_{\max}$.

In the state-of-the-art, the only assertion which could be done was that the optimal distinguisher is performing the best, i.e., it is better than all others. However, in general, it was difficult to formal and numerical comparison between distinguishers was not possible. We enable that. More precisely, we rate distinguishers according to whether they match for minimum or maximal values of κ. Then, we classify them when there is one distinguisher better than another one over the full range of $\sigma > 0$. Finally, we show that depending on the values of $\kappa_{\min}$ and $\kappa_{\max}$, some distinguishers might be better than others.

5.5 Conclusion

In this chapter, we have mathematically proven that only two parameters, the confusion coefficient and the SNR, determine the side-channel distinguishing efficiency for monobit leakages. Both of them are easy to compute because the confusion coefficient can be calculated with the knowledge of the operating substitution box and the SNR can be measured offline.

Our work is useful to predict how fast a distinguisher will succeed to recover the secret key. Long and painful simulations can be advantageously replaced by the computation of the success exponent using closed-form expressions.

This chapter also consolidates the state of the art about the classical distinguishers, especially for MIA and KSA. We have derived the success exponent for these two distinguishers as a function of the confusion coefficient and the standard deviation of the noise.

Chapter 6

When MIA is a Maximum Likelihood and better than CPA

6. WHEN MIA IS A MAXIMUM LIKELIHOOD AND BETTER THAN CPA

This chapter covers the work presented at ArticCrypt 2016.

Contents

6.1 Introduction

Many embedded systems implement cryptographic algorithms, which use secret keys that must be protected against extraction. Side-channel analysis (SCA) is one effective threat: physical quantities, such as instant power or radiated electromagnetic field, leak outside the embedded system boundary and reveal information about internal data. SCA consists in exploiting the link between the *leakage* signal and key-dependent internal data called *sensitive variables*.

The cryptographic algorithm is generally public information, whereas the implementation details are kept secret. For high-end security products, the confidentiality of the design is mandated by certification schemes, such as the Common Criteria [22]. For instance, to comply with ALC_DVS (Life-Cycle support – Development Security) requirement, the developer must provide a documentation that describes "*all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE (target of evaluation) design and implementation in its development environment*" [22, clause 2.1 C at page 141]. In particular, an attacker does not have enough information to precisely model the leakage of the device. On commercial products certified at highest evaluation assurance levels (EAL4+ or EAL5+), the attacker cannot set specific secret key values hence cannot profile the leakage [1] Therefore, many side-channel attacks can only be performed online using some distinguisher.

Correlation Power Analysis (CPA) [12] is one common side-channel distinguisher. It is known [39, Theorem 5] that its *optimality* holds only for a specific noise model (Gaussian) and for a specific knowledge of the deterministic part of the leakage—namely it should be perfectly known up to an unknown scaling factor and an unknown offset.

Linear Regression Analysis (LRA) [28] has been proposed in the context where the leakage model is drifting apart from a Hamming weight model. Its parametric structure and ability to include several basis functions makes it a very powerful tool, that can adjust to a broad range of leakage models when the additive noise is Gaussian. Incidentally, CPA may be seen as a 2-dimensional LRA [39].

When both model and noise are partially or fully unknown, generic distinguishers have been proposed, such as Mutual Information Analysis (MIA) [34], Kolmogorov-Smirnov test [86, 90] or Cramér-von-Mises test [86, Sec. 3.3.]. Thorough investigations have been carried out (e.g., [17, 60, 88]) to identify strengths and weaknesses of various distinguishers in various scenarios, including empirical comparisons. In keeping with these results, we aim at showing some *mathematical justification* regarding MIA versus CPA and LRA. Our goal is thus to structure the field of attacks, by providing theoretical motivations why attacks strength may differ, irrespective of the particular traces datasets.

Contributions. In this chapter, we derive MIA anew as the distinguisher which maximizes the success rate when the exact probabilities are replaced by online estimations. In order to assess the practicability of this mathematical result, we show two scenarios where MIA can outperform its competitors CPA and LRA, which themselves do not estimate probabilities. In these scenarios, we challenge the two hypotheses needed for CPA to be optimal: additive Gaussian noise and perfect knowledge of the model up to an affine transformation. This is illustrated in Fig. 6.1.

Last, we extend the fast computation trick presented in [49] to MIA: the distinguisher is only computed from time to time based on histograms obtained by accumulation, where the accumulated histograms are shared for all the key guesses.

[1] Obviously, this hypothesis only holds provided the device manufacturer does not reuse the same cryptographic engine in an open platform, such as a JavaCard, where the user is able to use the cryptographic API at its will.

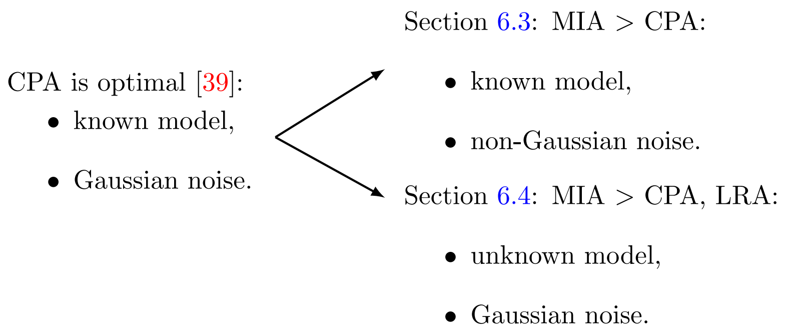


Figure 6.1: Illustration of two practical situations where MIA can defeat CPA

Organization. The remainder of this chapter is organized as follows. Section 6.2 provides notations, assumptions, and the rigorous mathematical derivation that MIA reduces to a maximum likelihood distinguisher, where exact leakage probabilities are replaced by online probabilities. Section 6.3 studies two examples where the attacker knows the one-bit model under non-Gaussian algorithmic noise, and for which MIA is shown to outperform CPA. Section 6.4 provides a scenario in which the leakage model is partially unknown under additive Gaussian noise, and where MIA outperforms CPA and LRA. Last, in Section 6.5, we propose a fast MIA computation deduced from our mathematical rewriting allowing to factor several computations. Section 6.6 concludes.

6.2 Optimality of Mutual Information Analysis

6.2.1 Notations and Assumptions

We assume that the attacker has at his disposal $\widetilde{q}$ independent *online* leakage measurements[1] $\widetilde{\mathbf{x}} = (\widetilde{x}_1, \ldots, \widetilde{x}_{\widetilde{q}})$[2] for some sequence of independent and uniformly distributed text n-bit words $\widetilde{\mathbf{t}} = (\widetilde{t}_1, \ldots, \widetilde{t}_{\widetilde{q}})$ (random but known). The n-bit secret key k^* is fixed but unknown.

We do not make any precise assumption on the leakage model—in particular the attacker is *not* able to estimate the actual probability density in a profiling phase. Instead we choose an algorithmic-specific function f and a device-specific function φ to compute, *for each key hypothesis* k, sensitive values $\widetilde{\mathbf{y}} = (\widetilde{y}_1, \ldots, \widetilde{y}_{\widetilde{q}})$ by the formula

$$\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}})), \tag{6.1}$$

[1] We comply with the usual notations of [30] where offline quantities are indicated with a hat, whereas online quantities are indicated with a tilde. In this chapter, there is no profiling phase hence no offline quantities.

[2] We use bold letters to indicate vectors while scalars are presented using small italic letters.

that is, $\widetilde{y}_i = \varphi(f(k, \widetilde{t}_i))$ for all $i = 1, \ldots, \widetilde{q}$. In practice, a suitable choice of φ should be optimized depending on some leakage model but in what follows, f and φ can be taken arbitrarily such that they fulfill the following *Markov condition*.

Assumption 6.1 (Markov condition). *The leakage $\widetilde{\mathbf{x}}$ depends on the actual secret key k^* only through the computed model $\widetilde{\mathbf{y}} = \varphi(f(k^*, \widetilde{\mathbf{t}}))$.*

Thus, while the conditional distribution $\mathbb{P}_k(\widetilde{\mathbf{x}}|\widetilde{\mathbf{t}})$ depends on the value k of the secret key, the expression $\mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}})$ depends on k only through $\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}}))$. If we let $\mathbb{P}_k(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}})$ be the joint probability distribution of $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{t}}$ when $k^* = k$, one has the Fisher factorization [19]

$$\mathbb{P}_k(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \mathbb{P}(\widetilde{\mathbf{t}})\mathbb{P}_k(\widetilde{\mathbf{x}}|\widetilde{\mathbf{t}}) = \mathbb{P}(\widetilde{\mathbf{t}})\mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) \qquad \text{where } \widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}})). \tag{6.2}$$

In the latter expression we have $\mathbb{P}(\widetilde{\mathbf{t}}) = 2^{-\widetilde{q}n}$ since all text n-bit words are assumed independent and identically distributed (i.i.d.) and uniformly distributed.

In the case of an additive noise model, we simply have $\widetilde{\mathbf{x}} = \widetilde{\mathbf{y}} + \widetilde{\mathbf{n}}$ where $\widetilde{\mathbf{n}}$ is the noise vector, and the Markov condition is obviously satisfied. In general, in order to fulfill the Markov condition the attacker needs some knowledge on the actual leakage model. We give two examples regarding the Markov condition:

Example 6.1. If leakage x_i is linked to t_i and k^* through the relationship $x_i = w_H(k^* \oplus t_i) + n_i$ for all $i = 1, \ldots, \widetilde{q}$, where w_H is the Hamming weight and n_i is the noise (independent of t_i), then both models $y_i = k \oplus t_i$ and $y_i = w_H(k \oplus t_i)$ satisfy the Markov condition.

In order to uniquely distinguish the correct key, some conditions on the expressions of y are required. Specifically, let us denote by y_k the function $t \mapsto y_k(t) = y(k, t)$, and let $\mathcal{B}$ the set of bijections on the leakage space $\mathcal{X}$. We have:

$$\text{if } \forall k, \exists k' \neq k, \quad \exists \beta \in \mathcal{B} \text{ s.t. } y_{k'} = \beta \circ y_k, \quad \text{then the distinguisher features a } \textit{tie}, \tag{6.3}$$

$$\text{if } \forall k, \forall k' \neq k, \quad \exists \beta \in \mathcal{B} \text{ s.t. } y_{k'} = \beta \circ y_k, \quad \text{then the distinguisher is } \textit{not sound}. \tag{6.4}$$

Indeed, in Eq. (6.3), there is no way for the distinguisher to tell k^* from k', and in Eq. (6.4), the distinguisher yields the same value for all the key guesses [1]. Sections 6.3 and 6.4 give other, more sophisticated, examples that satisfy the Markov condition.

Example 6.2. In the same scenario as in Example 6.1, consider the bit-dropping strategy (called `7LSB` in [34] and used in [72, 89]). Then e.g., $y_i = (k \oplus t_i)[1:7]$ (the first seven bit components) does *not* satisfy the Markov condition. Note that the leakage model in this example intentionally discards some information, hence may not be satisfactory [72].

[1] We refer the interested reader to the work done in [91, Sec. 3]. We note that $y_i = k \oplus t_i$ does not lead to a sound distinguisher, as for all k', $x \mapsto x \oplus k'$ is bijective, and maps y_k to $y_{k \oplus k'}$. On the contrary, there is no bijection β such that for all t, $w_H(k \oplus t) = \beta(w_H(k \oplus k' \oplus t))$. So the choice $y_i = w_H(k \oplus t_i)$ is sound.

Let $\widetilde{k}$ be the key estimate that maximizes a distinguisher $\mathcal{D}$ given $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{t}}$, i.e.,

$$\widetilde{k} = \arg \max_{k \in \mathcal{K}} \ \mathcal{D}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) \tag{6.5}$$

where $\mathcal{K}$ is the key space.

We also assume that leakage values are quantized[1] in a suitable finite set $\mathcal{X}$. Letting $\mathcal{Y}$ denote the discrete sensitive value space, we have $\widetilde{\mathbf{x}} \in \mathcal{X}^{\widetilde{q}}$ and $\widetilde{\mathbf{y}} \in \mathcal{Y}^{\widetilde{q}}$. The actual probability densities being unknown, the attacker estimates them online, during the attack, from the available data in the sequences $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ (*via* $\widetilde{\mathbf{t}}$), by counting all instances of possible values of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$:

$$\widetilde{\mathbb{P}}(x) = \frac{1}{\widetilde{q}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x}, \tag{6.6}$$

$$\widetilde{\mathbb{P}}(y) = \frac{1}{\widetilde{q}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{y}_i = y}, \tag{6.7}$$

$$\widetilde{\mathbb{P}}(x, y) = \frac{1}{\widetilde{q}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{y}_i = y}, \tag{6.8}$$

$$\widetilde{\mathbb{P}}(x|y) = \frac{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{y}_i = y}}{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{y}_i = \widetilde{y}}} = \frac{\widetilde{\mathbb{P}}(x, y)}{\widetilde{\mathbb{P}}(y)}, \tag{6.9}$$

where $\mathbb{1}_A$ denotes the indicator function of A: $\mathbb{1}_A = 1$ if A is true and $= 0$ otherwise.

Definition 6.1 (Empirical Mutual Information)**.** The empirical mutual information is defined as

$$\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \widetilde{\mathbb{P}}(x, y) \ \log_2 \frac{\widetilde{\mathbb{P}}(x, y)}{\widetilde{\mathbb{P}}(x) \widetilde{\mathbb{P}}(y)}, \tag{6.10}$$

which can also be written as

$$\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \widetilde{H}(\widetilde{\mathbf{x}}) - \widetilde{H}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}), \tag{6.11}$$

where the empirical entropies are defined as

$$\widetilde{H}(\widetilde{\mathbf{x}}) = \sum_{x \in \mathcal{X}} \widetilde{\mathbb{P}}(x) \ \log_2 \frac{1}{\widetilde{\mathbb{P}}(x)} \tag{6.12}$$

and

$$\widetilde{H}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \widetilde{\mathbb{P}}(x, y) \ \log_2 \frac{1}{\widetilde{\mathbb{P}}(x|y)}. \tag{6.13}$$

These quantities are functions of the sequences $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ since $\widetilde{\mathbb{P}}(x, y)$ is a function of $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$. They also depend on the key guessed value k, *via* the expression of $\widetilde{\mathbf{y}}$.

[1] Some side-channels are discrete by nature, such as the timing measurements (measured in units of clock period). In addition, oscilloscopes or data acquisition appliances rely on ADCs (Analog to Digital Converters), which usually sample a continuous signal into a sequence of integers, most of the time represented on 8 bits (hence $\mathcal{X} = \mathbb{F}_2^8$).

6.2.2 Mathematical Derivation

In this subsection, we show that MIA coincides with the maximum likelihood expression where leakage probabilities $\mathbb{P}$ are replaced by online estimated probabilities $\widetilde{\mathbb{P}}$.

Definition 6.2 (Success Rate [82, Sec. 3.1])**.** The *success rate* (averaged over all possible secret key values) is defined as:

$$\mathrm{SR} = \frac{1}{2^n} \sum_{k=0}^{2^n-1} \mathbb{P}_k(\widetilde{k} = k). \tag{6.14}$$

Here we follow a frequentist approach. An equivalent alternative Bayesian approach would be to assume a uniform prior key distribution [39].

Theorem 6.1 (Maximum Likelihood [20])**.** *Let* $\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}}))$*. The* optimal key estimate *that maximizes the success rate* (6.14) *is:*

$$\widetilde{k} = \arg\max_k \ \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}). \tag{6.15}$$

Proof. We give here a formal proof, which nicely relates to Definition 6.2. Straightforward computation yields:

$$\begin{aligned}
\mathrm{SR} &= \frac{1}{2^n} \sum_{k=1}^{2^n} \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{t}}} \mathbb{P}_k(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) \ \mathbb{1}_{k=\widetilde{k}} && (6.16)\\
&= \frac{1}{2^n} \sum_{k=1}^{2^n} \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{t}}} \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}}))) \ \mathbb{P}(\widetilde{\mathbf{t}}) \ \mathbb{1}_{k=\widetilde{k}} \quad \text{(by (6.2) \& Assumption 6.1)} && (6.17)\\
&= \frac{1}{2^{n(\widetilde{q}+1)}} \sum_{k=1}^{2^n} \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{t}}} \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}}))) \ \mathbb{1}_{k=\widetilde{k}} && (6.18)\\
&= \frac{1}{2^{n(\widetilde{q}+1)}} \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{t}}} \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}} = \varphi(f(k = \widetilde{k}, \widetilde{\mathbf{t}}))). && (6.19)
\end{aligned}$$

Thus, for each given sequences $\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}$ maximizing the success rate amounts to choosing $k = \widetilde{k}$ so as to maximize $\mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}} = \varphi(f(k = \widetilde{k}, \widetilde{\mathbf{t}})))$:

$$\widetilde{k} = \arg\max_k \ \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}). \tag{6.20}$$

□

When no profiling is possible the conditional distribution

$$\mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \prod_{i=1}^{\widetilde{q}} \mathbb{P}(\widetilde{x}_i|\widetilde{y}_i) \tag{6.21}$$

is unknown to the attacker. Therefore, Theorem 6.1 is no longer practical and we require a *universal*[1] version of it.

Definition 6.3 (Universal Maximum Likelihood)**.** Let $\widetilde{\mathbf{y}} = \varphi(f(k, \widetilde{\mathbf{t}}))$. The *universal* maximum likelihood (UML) key estimate is defined by

$$\widetilde{k} = \arg\max_k \widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}), \tag{6.22}$$

where

$$\widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \prod_{i=1}^{\widetilde{q}} \widetilde{\mathbb{P}}(\widetilde{x}_i|\widetilde{y}_i). \tag{6.23}$$

Here $\widetilde{\mathbb{P}}$, defined in Equations (6.9), (6.8), (6.7) and (6.6), is estimated directly from the available data, that is, from the actual values in the sequences $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$.

Theorem 6.2 (UML is MIA)**.** *The universal maximum likelihood key estimate is equivalent to the mutual information analysis (MIA) [34]:*

$$\widetilde{k} = \arg\max_k \widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \arg\max_k \widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}), \tag{6.24}$$

where $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ is the universal mutual information (definition 6.1).

Proof. Rearrange the likelihood product according to values taken by the $\widetilde{x}_i$ and $\widetilde{y}_i$'s:

$$\widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \prod_{i=1}^{\widetilde{q}} \widetilde{\mathbb{P}}(\widetilde{x}_i|\widetilde{y}_i) = \prod_{x\in\mathcal{X}, y\in\mathcal{Y}} \widetilde{\mathbb{P}}(x|y)^{\widetilde{n}_{x,y}} \tag{6.25}$$

where $\widetilde{n}_{x,y}$ is the number of components $(\widetilde{x}_i, \widetilde{y}_i)$ equal to (x, y), i.e.,

$$\widetilde{n}_{x,y} = \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{y}_i = y} = \widetilde{q}\, \widetilde{\mathbb{P}}(x, y). \tag{6.26}$$

The second inequality in Eqn. (6.25) is based on a counting argument: some events collide, i.e., we have $(x_i, y_i) = (x_{i'}, y_{i'})$ for $i \neq i'$. The exponent $\widetilde{n}_{x,y}$ is meant to enumerate all such possible collisions. This gives

$$\widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}}) = \prod_{x\in\mathcal{X}, y\in\mathcal{Y}} \widetilde{\mathbb{P}}(x|y)^{\widetilde{q}\,\widetilde{\mathbb{P}}(x,y)} = 2^{-\widetilde{q}\widetilde{H}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}})}, \tag{6.27}$$

(see Definition 6.1). Therefore, maximizing $\widetilde{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}})$ amounts to minimizing the empirical conditional entropy $\widetilde{H}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}})$. Since $\widetilde{H}(\widetilde{\mathbf{x}})$ is key-independent, this in turn amounts to maximizing the empirical mutual information $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \widetilde{H}(\widetilde{\mathbf{x}}) - \widetilde{H}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{y}})$. □

[1] *Universal*, in the information theoretic sense of the word, means: computed from the available data without prior information.

From Theorem 6.2 we can conclude that MIA is "optimal" as a universal maximum likelihood estimation. This constitutes a rigorous proof that mutual information is a relevant tool for key recovery when the leakage is unknown (in the case where the model satisfies the Markov condition) as was already hinted in [34, 66, 72, 88].

Corollary 6.1. *MIA coincides with the ML distinguisher as $\widetilde{q} \to \infty$.*

Proof. By the law of large numbers, the online probability $\widetilde{\mathbb{P}}$ converges almost surely to the exact probability of the leakage as $\widetilde{q} \to \infty$. For any fixed values of $\widetilde{x} \in \mathcal{X}, \widetilde{y} \in \mathcal{Y}$,

$$\widetilde{\mathbb{P}}(\widetilde{x}|\widetilde{y}) \underset{\widetilde{q}\to\infty}{\longrightarrow} \mathbb{P}(\widetilde{x}|\widetilde{y}) \qquad a.s.$$

Thus in the limit, MIA coincides with the maximum likelihood rule. □

Remark 6.1. It is well known [66] that if the mapping $\widetilde{t} \mapsto \widetilde{y} = \varphi(f(k,\widetilde{t}))$ is one-to-one (for all values of k), then MIA cannot distinguish the correct key. This is also clear from Eq. (6.4) in footnote 1: given two different keys k, k', there is a bijection between y_k and $y_{k'}$, which is simply $\beta = y_{k'} \circ y_k^{-1}$. In our present setting this is easily seen by noting that when $\widetilde{y} = \varphi(f(k,\widetilde{t}))$,

$$\widetilde{\mathbb{P}}(x|y) = \frac{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{y}_i = y}}{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{y}_i = y}} = \frac{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{t}_i = t}}{\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{t}_i = t}} \tag{6.28}$$

is independent of the value k. Note that this is true for any fixed number of measurements $\widetilde{q}$ during the attack.

6.2.3 MIA Faster Than ML Distinguisher

Now that we have shown that the *Universal* Maximum Likelihood distinguisher is strictly equivalent to the MIA distinguisher, we show that the use of the MIA Distinguisher is cheaper in terms of calculations than the ML distinguisher. Both distinguishers require the knowledge of $\widetilde{\mathbb{P}}$, the online estimation of the leakage probability. However, the summation is not exactly the same:

- the ML distinguisher consists in a sum of $\widetilde{q}$ logarithms, whereas
- the MIA involves a sum over $|\mathcal{X}| \times |\mathcal{Y}|$ logarithms[1].

This means that computing a ML requires $\widetilde{q}$ logarithm computations while computing a MIA requires $|\mathcal{X}| \times |\mathcal{Y}|$ logarithm computations. As long as $|\mathcal{X}| \times |\mathcal{Y}|$ is smaller than $\widetilde{q}$, which is verified for practical signal-to-noise values, the MIA is faster than the ML in terms of logarithm computations. Furthermore, in section 6.5.2, we present a fast algorithm to compute MIA; it takes advantage of precomputations, which are similar to that already presented in [49].

[1] In practice, logarithms require a high computational power, hence the number of calls to this function shall be minimized.

6.3 Non-Gaussian Noise Challenge

In this section, we show two examples where MIA outperforms CPA due to non-Gaussian noise. The first example presented in subsection 6.3.1 is an academic (albeit artificial) example built in order to have the success rate of CPA collapse. The second example in subsection 6.3.2 is more practical.

6.3.1 Pedagogical Case-study

We consider a setup where the variables are $X = Y + N$, with $Y = \varphi(f(k^*, T))$, where $Y \in \{\pm 1\}$, and $N \sim \mathcal{U}(\{\pm\sigma\})$ (meaning that N takes values $-\sigma$ and $+\sigma$ randomly, with probabilities $\frac{1}{2}$ and $\frac{1}{2}$), where σ is an integer. Specifically, we assume that k^*, $t \in \mathbb{F}_2^n$, with $n = 4$, and that $f : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^m$ is a (truncated version) of the SERPENT Sbox[1] fed by the XOR of the two inputs (key and plaintext nibbles) and $\varphi = w_H$ is the Hamming weight (which reduces to the identity $\mathbb{F}_2 \to \mathbb{F}_2$ if $m = 1$ bit).

The optimal distinguisher (Theorem 6.1) in this scenario has the following closed-form expression:

$$\mathcal{D}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_k \mathbb{P}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{t}}, k) = \arg\max_k \frac{1}{2^{\widetilde{q}}} \prod_{i=1}^{\widetilde{q}} \delta(\widetilde{x}_i, \widetilde{t}_i, k), \tag{6.29}$$

where $\delta : \mathbb{F}_2^m \times \mathbb{F}_2^n \times \mathbb{F}_2^n \to \{0, 1\}$ is defined as:

$$\delta(x, t, k) = \begin{cases} 1 & \text{if } x - \varphi(f(k, t)) = -\sigma, \\ 1 & \text{if } x - \varphi(f(k, t)) = +\sigma, \\ 0 & \text{otherwise.} \end{cases}$$

The evaluation of this quantity requires the knowledge of σ, which by definition is an unknown quantity related to the noise. Our simulations have been carried out as follows.

1. Generate two large uniformly distributed random vectors $\widetilde{\mathbf{t}}$ and $\widetilde{\mathbf{n}}$ of length $\widetilde{q}$;

2. Deliver the pair of vectors $(\widetilde{\mathbf{t}}, \widetilde{\mathbf{x}} = \varphi(f(k^*, \widetilde{\mathbf{t}})) + \widetilde{\mathbf{n}})$ to the attacker;

3. Estimate averages and PMFs (probability mass functions) of this data for $\widetilde{q}_{\mathsf{step}}$ $(= 1)$, then for $2\widetilde{q}_{\mathsf{step}}$, $3\widetilde{q}_{\mathsf{step}}$ and so on;

[1] The least significant bit S_0 of the PRESENT Sbox S is not suitable because one has $\forall z \in \mathbb{F}_2^4$, $S_0(z) = S_0(z \oplus \texttt{0x9}) = \neg S_0(z \oplus \texttt{0x1}) = \neg S_0(z \oplus \texttt{0x8})$. As in Eq. (6.3) of footnote 1, *ties* occur: it is not possible to distinguish k^*, $k^* \oplus \texttt{0x9}$, $k^* \oplus \texttt{0x1}$, $k^* \oplus \texttt{0x8}$ (the corresponding bijections are respectively $x \mapsto x$ and $x \mapsto 1 - x$). Therefore, we consider component 1 instead of 0, which does not satisfy such relationships.

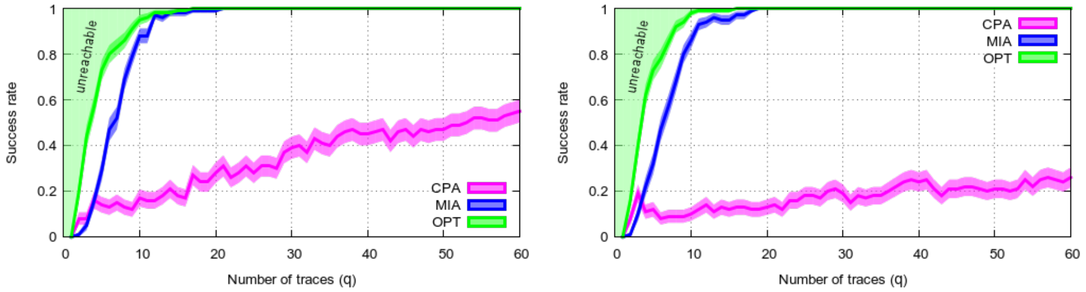


Figure 6.2: Success rate for $\sigma = 2$ (left) and $\sigma = 4$ (right), when $Y \sim \mathcal{U}(\{\pm 1\})$ and $N \sim \mathcal{U}(\{\pm\sigma\})$

4. At each multiple of $\widetilde{q}_{\mathsf{step}}$, carry out CPA and MIA.

The attacks are reproduced 100 times to allow for narrow error bars on the estimated success rate.

Remark 6.2. We do not consider *linear regression analysis* because the model is not parametric. The only unknown parameter is related to the noisy part of the leakage, not its deterministic part.

Simulation results are given in Fig. 6.2 for $\sigma = 2$ and $\sigma = 4$. The success rate of the "optimal" distinguisher (the maximum likelihood distinguisher of Theorem 6.1 – see Eqn. (6.29)) is drawn in order to visualize the limit between feasible (below) and unfeasible (above) attacks. It can be seen that MIA is almost as successful as the maximum likelihood distinguisher, despite the knowledge of the value of σ is not required for the MIA. In addition, one can see that the CPA performs worse, and all the worst as σ increases. In this case, the CPA is not the optimal distinguisher (as e.g., underlined in [39, Theorem 5]) since the noise is not Gaussian (but discrete).

Remark 6.3. Another attack strategy for the leakage model presented in this subsection would simply be to filter out the noise. One could for instance dispose of all traces where the leakage is negative. The remaining traces (half of them) contain a constant noise $N = +\sigma > 1$, hence the signal Y can be read out without noise. Such attack, known as the *subset attack* [62, Sec. 5.2], is not far from the optimal one (Eqn. (6.29)). It actually does coincide with the optimal attack if the attacker recovers Y from both subsets $\{i/X_i > 0\}$ and $\{i/X_i < 0\}$. Still it can noted that MIA is very close to being optimal for this scenario.

Asymptotics. We can estimate the theoretical quantities for CPA and MIA as follows. We have $\mathsf{Var}(Y) = 1$ and $\mathsf{Var}(N) = \sigma^2$, hence a signal to noise ratio $\mathrm{SNR} = 1/\sigma^2$. In addition, X can only take four values: $\pm 1 \pm \sigma$. Since $\mathbb{E}(XY) = \mathbb{E}(X^2) + \mathbb{E}(YN) = Var(X) + \mathbb{E}(Y)\mathbb{E}(N) = 1 + 0 \times 0 = 1$, the correlation is simply $\rho(X,Y) = 1/\sigma$, which vanishes as σ increases.

However, for $\sigma > 1$, the mutual information $\mathsf{I}(X,Y) = 1$ bit. Indeed, $\mathsf{H}(X) = -\sum_{x\in\{\pm1\pm\sigma\}} \mathbb{P}(X = x)\log_2 \mathbb{P}(X = x) = -\sum_{x\in\{\pm1\pm\sigma\}} \frac{1}{4}\log_2 \frac{1}{4} = \log_2 4 = 2$ bit, $\mathsf{H}(X|y = \pm1) = \log_2 2 = 1$ bit, so $\mathsf{I}(X,Y) = \log_2 4 - \sum_{y\in\{\pm1\}} \mathbb{P}(X = x)\log_2 2 = \log_2 4 - \log_2 2 = 1$ bit, irrespective of $\sigma \in \mathbb{N}$.

The important fact is that the mutual information does not depend on the value of σ. Accordingly, it can be seen from Fig. 6.2 that the success rate of the MIA is not affected by the noise variance. This explains why MIA will outperform the CPA for large enough σ.

6.3.2 Application to Bitslice PRESENT

Bitslicing algorithms is a common practice. This holds both for standard [70] (e.g., AES) and lightweight [56] (PRESENT, Piccolo) block ciphers. Here the distinguishers must be single-bit: $Y \in \{\pm1\}$. However, compared to the case of Sec. 6.3.1, the noise takes now more than two values: On an 8-bit computer, the 7 other bits will leak independently. They are, however, not concerned by the attack, and constitute *algorithmic noise* N which follows a binomial law $\alpha \times \mathcal{B}(7, \frac{1}{2})$, where α is a scaling factor.

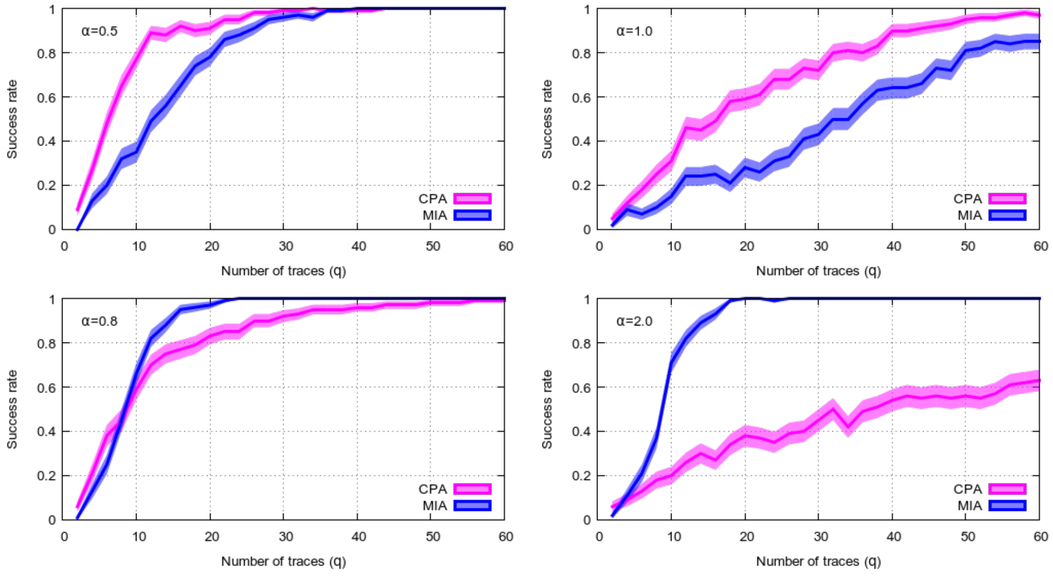


Figure 6.3: Success rate for the attack of a bitsliced algorithm on an 8-bit processor, where 7 bits make up algorithmic noise, and have weight 0.5, 1.0 (top) and 0.8 and 2.0 (bottom).

Simulation results for various values of α are in Fig. 6.3. Interestingly, MIA is efficient for the cases where the leakage $Y \sim \mathcal{U}(\{\pm1\})$ is not altered by the addition of noise: For $\alpha = 0.8$ and $\alpha = 2.0$, it is still possible to tell unambiguously from X what is the value of Y. On the contrary, when $\alpha = 0.5$ or $\alpha = 1.0$, the function $(Y, N) \mapsto X = Y + N$ is not one-to-one. For instance, in the case $\alpha = 1.0$, the value $X = 2$ can result as well from $Y = -1$ and $N = 3$, or $Y = +1$ and $N = 2$. (see Fig. 6.4).

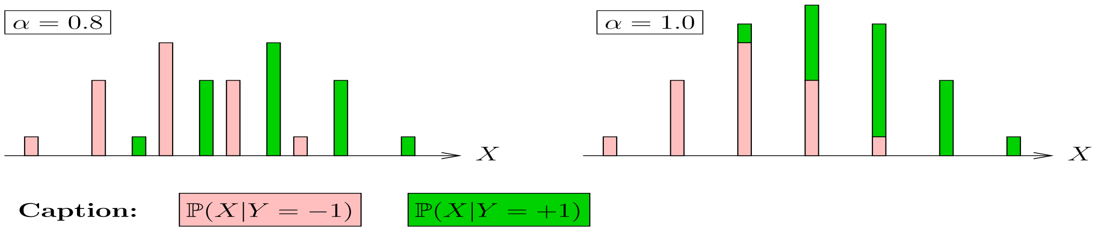


Figure 6.4: Illustration bijectivity (left) vs. non-injectivity (right) of the leakage function.

6.4 Partially Unknown Model Challenge

Veyrat-Charvillon and Standaert [86, section 4] have already noticed that MIA can outperform CPA if the model is drifted too far away from the real leakage. However, LRA is able to make up for the model drift of [86] (which considered unevenly weighted bits). In this section, we challenge CPA and LRA with a partially unknown model. We show that, in our example, MIA has a much better success rate than both CPA and LRA.

For our empirical study we used the following setup:

$$X = \psi(Y(k^*)) + N, \qquad Y(k^*) = w_H(\mathrm{S_{box}}(k^* \oplus T)),$$

where $\mathrm{S_{box}}$ is the AES substitution box, ψ is the non-linear function given by:

x	0	1	2	3	4	5	6	7	8
$\psi(x)$	$+1$	$+2$	$+3$	$+4$	0	-4	-3	-2	-1

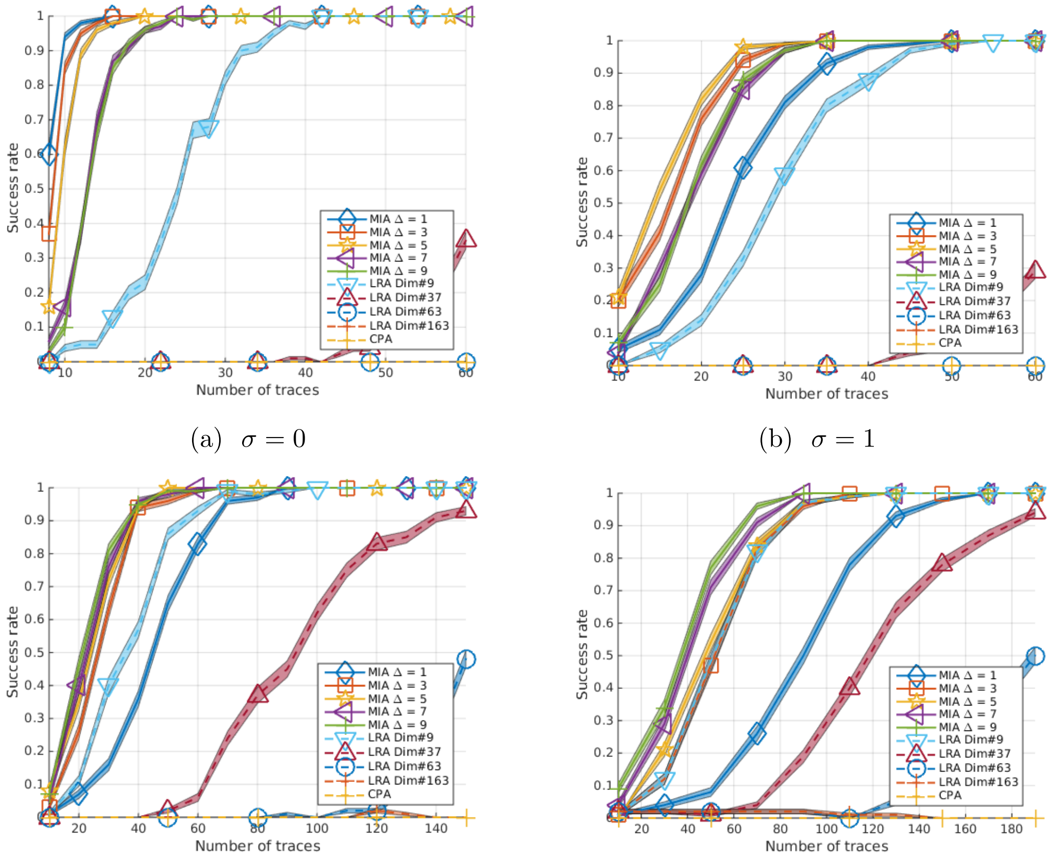
which is unknown to the attacker, and N is a centered Gaussian noise with unknown standard deviation σ. The non-linearity of ψ is motivated by [60], where it is discussed that a linear model favors CPA over MIA.

The leakage is continuous due to the Gaussian noise. In order to discretize the leakage to obtain discrete probabilities, we used the *binning* method. We conducted MIA with several different binning sizes:

$$B = \{[(i-1) \times \Delta x, i \times \Delta x[, \ i \in \mathbb{Z}\} \qquad \text{for } \Delta x = \{1, 3, 5, 7, 9\}. \tag{6.30}$$

In this chapter, we do not try to establish any specific result about binning, but content ourselves to present empirical results obtained with different bin sizes.

We have carried out LRA for the standard basis in dimension $d = 9$ and higher dimensions $d = \{37, 93, 163\}$. More precisely, for $d = 9$ we have $\widetilde{\mathbf{y}}'(k) = (\vec{1}, \widetilde{\mathbf{y}}_1(k), \widetilde{\mathbf{y}}_2(k), \ldots, \widetilde{\mathbf{y}}_8(k))$ with $\widetilde{\mathbf{y}}_j(k) = [\mathrm{Sbox}(k \oplus T)]_j$ where $[\cdot]_j : \mathbb{F}_2^n \to \mathbb{F}_2$ is the projection mapping onto the j^{th} bit. For $d = 37$ the attacker additionally takes into consideration the products between all possible $\widetilde{\mathbf{y}}_j$ $(1 \leq j \leq 8)$, i.e., $\widetilde{\mathbf{y}}_1 \cdot \widetilde{\mathbf{y}}_2, \widetilde{\mathbf{y}}_1 \cdot \widetilde{\mathbf{y}}_3, \widetilde{\mathbf{y}}_1 \cdot \widetilde{\mathbf{y}}_4$ and so on. Consequently, $d = 93$ considers additionally the product between 3 $\widetilde{\mathbf{y}}$'s and $d = 163$ includes also all possible product combinations with 4 columns. See [37] for a detailed description on the selection of basis functions.



(a) $\sigma = 0$

(b) $\sigma = 1$

(c) $\sigma = 2$

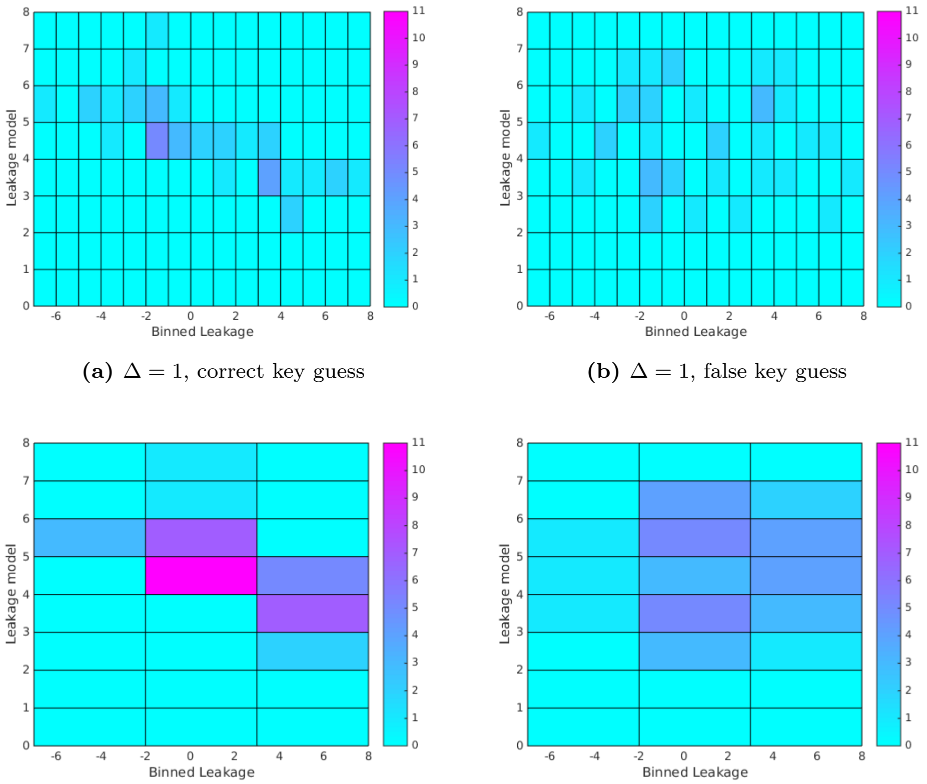
(d) $\sigma = 3$

Figure 6.5: Success rate for $\sigma \in \{0, 1, 2, 3\}$ when the model is unknown

Fig. 6.5 shows the success rate using 100 independent experiments. Perhaps surprisingly, MIA turns out to be more efficient than LRA. Quite naturally, MIA and LRA become closer as the the variance of the independent measurement noise N increases. It can be seen that LRA using higher dimension requires a sufficient number of traces for estimation (for $d = 37$ around

100, $d = 93$ around 150, and $d = 137$ failed below 200 traces). Consequently, in this scenario using high dimensions is not appropriate, even if the high dimension in question might fit the unknown function ψ.

One reason why MIA outperforms CPA and LRA in this scenario is that the function ψ was chosen to have a null covariance. Moreover, one can observe that the most efficient binning size depends on the noise variance and thus on the scattering of the leakage. As σ grows larger values of Δ should be chosen. This is contrary to the suggestions made in [34], which proposes to estimate the probability distributions as good as possible and thus to consider as many bins as there are distinct values in the traces. In our experiments, when noise is absent ($\sigma = 0$) the optimal binning size is $\Delta = 1$ which is equivalent to the step size of Y, while for $\sigma = 2$ the optimal binning is $\Delta = 5$ (see Fig. 6.5(c)).



(a) $\Delta = 1$, correct key guess

(b) $\Delta = 1$, false key guess

(c) $\Delta = 5$, correct key guess

(d) $\Delta = 5$, false key guess

Figure 6.6: Estimated $\widetilde{\mathbb{P}}(X|Y)$ using 40 traces for $\sigma = 2$ (see Fig. 6.5(c))

It can be seen that using 40 traces the success rate of MIA with $\Delta = 5$ reaches 90%, whereas using $\Delta = 1$ it is only about 30%. To understand this phenomenon, Fig. 6.6 displays the estimated $\widetilde{\mathbb{P}}(x|y)$ in a 3D histogram for the correct key and one false key hypothesis, such that MIA is able to reveal the correct key using $\Delta = 5$ but fails for $\Delta = 1$. Clearly, the distinguishability between the correct and false key is much higher in case of $\Delta = 5$ than for $\Delta = 1$.

More precisely, as the leakage is dispersed by the noise the population of bins of the false key becomes similar to the the ones of the correct key when considering smaller binning size (compare Fig. 6.6a and 6.6b). In contrast, the difference is more visible when the leakage is quantified into larger bins (compare Fig. 6.6c and 6.6d). Therefore, even if the estimation of $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ using $\widetilde{\mathbb{P}}(x|y)$ for larger Δ is more coarse and thus looses some information, the distinguishing ability to reveal the correct key is enhanced.

6.5 Fast Computations

In this section, we explain how we compute CPA and MIA in a faster way. We first show an algorithm for CPA, then we move to MIA.

6.5.1 Fast computation of CPA

We recall here the definition of empirical CPA:

$$\begin{aligned}\rho(X, Y(k)) &= \frac{\frac{1}{m}\sum_{i=1}^m x_i y_i(k) - \left(\frac{1}{m}\sum_{i=1}^m x_i\right)\left(\frac{1}{m}\sum_{i=1}^m y_i(k)\right)}{\sqrt{\frac{1}{m}\sum_{i=1}^m x_i^2 - \left(\frac{1}{m}\sum_{i=1}^m x_i\right)^2}\sqrt{\frac{1}{m}\sum_{i=1}^m y_i^2(k) - \left(\frac{1}{m}\sum_{i=1}^m y_i(k)\right)^2}} \\ &= \frac{m\sum_{i=1}^m x_i y_i(k) - \left(\sum_{i=1}^m x_i\right)\left(\sum_{i=1}^m y_i(k)\right)}{\sqrt{m\sum_{i=1}^m x_i^2 - \left(\sum_{i=1}^m x_i\right)^2}\sqrt{m\sum_{i=1}^m y_i^2(k) - \left(\sum_{i=1}^m y_i(k)\right)^2}},\end{aligned} \tag{6.31}$$

where $y_i(k) = \varphi(f(k, t_i))$. For the fast computation the following accumulators are required:

- $\mathsf{sx}[t] = \sum_{i/t_i=t} x_i$, the sum of leakages for a common t;
- $\mathsf{sx2}[t] = \sum_{i/t_i=t} x_i^2$, the sum of leakage squares for a common t;
- $\mathsf{qt}[t] = \sum_{i/t_i=t} 1$, the number of t which occurred.

We detail the various terms in the two next paragraphs.

6.5.1.1 Averages

First, we simply have:

$$\sum_{i=1}^{m} x_i = \sum_t \mathsf{sx}[t],$$

which is key independent. Second:

$$\begin{aligned}\sum_{i=1}^{m} y_i &= \sum_t \sum_{i/t_i=t} y_i \\ &= \sum_t \sum_{i/t_i=t} \varphi(f(k,t_i)) \\ &= \sum_t \varphi(f(k,t_i))\mathsf{qt}[t],\end{aligned}$$

which, quite surprisingly, is key independent.

6.5.1.2 Scalar product

The scalar product can be written the following way:

$$\begin{aligned}\sum_{i=1}^{m} x_i y_i &= \sum_y \sum_{i/y_i=y} x_i y_i \\ &= \sum_y y \sum_{i/y_i=y} x_i \\ &= \sum_y y \sum_{i/\varphi(f(k,t_i))=y} x_i \\ &= \sum_y y\ \mathsf{sx}'[y],\end{aligned}$$

where $\mathsf{sx}'[y] = \sum_{t/\varphi(f(k,t))=y} \mathsf{sx}[t]$. This optimization is certainly useful for long traces, because it minimizes the number of multiplications (precisely, only 2^m multiplications are done). But, we need 2^m temporary accumulators to save the $\mathsf{sx}'[y]$.

However, in monosample traces, we can also use this more simple computation:

$$\begin{aligned}\sum_{i=1}^{m} x_i y_i &= \sum_t \sum_{i/t_i=t} x_i \varphi(f(k,t_i)) \\ &= \sum_t \sum_{i/t_i=t} \varphi(f(k,t_i)) x_i \\ &= \sum_t \varphi(f(k,t_i)) \sum_{i/t_i=t} x_i \\ &= \sum_t \varphi(f(k,t_i))\mathsf{sx}[t].\end{aligned}$$

6.5.2 Fast computation of MIA

We setup a structure for the PMF, namely an array of hash tables, denoted as $\mathsf{PMF}[t][x]$, where t and x live in the sets $\mathbb{F}_2^n$ and $\mathsf{Im}(\varphi \circ f) + \mathsf{supp}(N)$. So, let us say we have accumulated $\widetilde{q}$ leakage pairs (t, x).

At this stage, we have the joint probability given by

$$\widetilde{\mathbb{P}}(t, x) = \frac{1}{\widetilde{q}} \mathsf{PMF}[t][x].$$

Now, when using MIA as a distinguisher, we need to compute $\mathsf{PMF}[y][x]$, where $y \in \mathbb{F}_2^n$ (and we expect the $(t, k) \mapsto y = \varphi(f(k, t))$ function to be *non-injective* [91], which is the case in the previous sections). The value $\mathsf{PMF}[y][x]$ implicitly depends upon a key guess k, as: $\mathsf{PMF}[y][x] = \mathsf{PMF}[y = \varphi(f(k, t))][x]$. Now, instead of computing $\widetilde{\mathbb{P}}(x, y)$ through $\mathsf{PMF}[y][x]$ explicitly for each key guess, we are able to reformulate

$$\widetilde{\mathbb{P}}(y, x) = \sum_t \widetilde{\mathbb{P}}(t, y, x) = \sum_{t/\varphi(f(k,t))=y} \widetilde{\mathbb{P}}(t, x) = \frac{1}{\widetilde{q}} \sum_{t/\varphi(f(k,t))=y} \mathsf{PMF}[t][x].$$

Thus, we can reuse the tabulated $\mathsf{PMF}[t][x]$ for each key guess, which requires thus much less computations as a straightforward implementation.

Recall the expression of the estimated mutual information:

$$\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \sum_{x,y} \widetilde{\mathbb{P}}(x, y) \log_2 \frac{\widetilde{\mathbb{P}}(x, y)}{\widetilde{\mathbb{P}}(x)\, \widetilde{\mathbb{P}}(y)}.$$

The value for $\widetilde{\mathbb{P}}(x)$ is identical for all key hypotheses and thus can be factored out. Indeed, this quantity is a *scaling constant* which could be omitted. But for the sake of completeness, we have

$$\widetilde{\mathbb{P}}(x) = \sum_t \widetilde{\mathbb{P}}(t, x) = \frac{1}{\widetilde{q}} \sum_t \mathsf{PMF}[t][x].$$

Lastly, we need to evaluate $\widetilde{\mathbb{P}}(y)$. This is simply done as:

$$\widetilde{\mathbb{P}}(y) = \sum_x \widetilde{\mathbb{P}}(x, y),$$

Algorithm 2 illustrates the fast computation process for MIA while Algorithm 3 computes the success rate of MIA. It calls the function MIA-Distinguisher as a subroutine. This last function corresponds to the of the computation of fast MIA (Alg. 2). However, it is optimized this way:

Algorithm 2: Fast computation algorithm for MIA

```
input  : $\widetilde{\mathbf{x}}$ a set of $\widetilde{q}$ traces which take discrete values,
         $\widetilde{\mathbf{t}}$ a corresponding set of $\widetilde{q}$ plaintexts/ciphertexts
output : $(\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)))_{k \in \mathcal{K}}$

// From $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{t}}$, build a hash table $\mathsf{PMF}[t][x]$ (i.e., an histogram)
for $i \in \{1, \ldots, \widetilde{q}\}$ do
    $\mathsf{PMF}[\widetilde{t}_i][\widetilde{x}_i] \mathrel{+}= 1$
end

for $x \in \mathcal{X}$ do
    $\widetilde{\mathbb{P}}(x) = 0$                                  // $\widetilde{\mathbb{P}}(x)$ holds $m\widetilde{\mathbb{P}}(x)$, cf. Eqn. (6.6)
    for $t \in \mathbb{F}_2^n$ do
        $\widetilde{\mathbb{P}}(x) \mathrel{+}= \mathsf{PMF}[t][x]$
    end
end

for $k \in \mathcal{K}$ do // Key enumeration
    $\forall x \in \mathcal{X}, y \in \mathcal{Y}, \widetilde{\mathbb{P}}(x, y) = 0$           // $\widetilde{\mathbb{P}}(x,y)$ holds $m\widetilde{\mathbb{P}}(x,y)$, cf. Eqn. (6.8)
    for $t \in \mathbb{F}_2^n$ do
        for $x \in \mathcal{X}$ do
            $\widetilde{\mathbb{P}}(x, \varphi(f(k,t))) \mathrel{+}= \mathsf{PMF}[t][x]$           // $y = \varphi(f(k,t))$, cf. Eqn. (6.1)
        end
    end
    $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)) = 0$
    for $y \in \mathcal{Y}$ do
        $\widetilde{\mathbb{P}}(y) = 0$                              // $\widetilde{\mathbb{P}}(y)$ holds $m\widetilde{\mathbb{P}}(y)$, cf. Eqn. (6.7)
        for $x \in \mathcal{X}$ do
            $\widetilde{\mathbb{P}}(y) \mathrel{+}= \widetilde{\mathbb{P}}(x, y)$
        end
        for $x \in \mathcal{X}$ do
            // Nota bene: $(\widetilde{\mathbb{P}}(x) = 0 \vee \widetilde{\mathbb{P}}(y) = 0) \implies \widetilde{\mathbb{P}}(x, y) = 0$
            if $\widetilde{\mathbb{P}}(x) \neq 0$ and $\widetilde{\mathbb{P}}(y) \neq 0$ then
                $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)) \mathrel{+}= \frac{\widetilde{\mathbb{P}}(x,y)}{m} \log_2 \left( \frac{m\widetilde{\mathbb{P}}(x,y)}{\widetilde{\mathbb{P}}(x)\widetilde{\mathbb{P}}(y)} \right)$
            end
        end
    end
end

return $(\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)))_{k \in \mathcal{K}}$                    // As in Eqn. (6.10)
```

Algorithm 3: Computation of the MIA success rate

```
input  :
    • $\widetilde{\mathbf{x}}$ a set of $\widetilde{q}$ traces which take discrete values,
    • $\widetilde{\mathbf{t}}$ a corresponding set of $\widetilde{q}$ plaintexts/ciphertexts,
    • $k^* \in \mathcal{K}$, the correct key,
    • $\widetilde{q}_{\mathsf{step}}$, typically of the order of $\widetilde{q}/100$ (number of times the success rate is computed),
    • $M$, the number of experiments
output : $\widehat{\mathrm{SR}}_{k^*}$, the empirical rate of MIA (computed as per Eqn. (6.10))
$\widehat{\mathrm{SR}}_{k^*} = \{0, \ldots, 0\}$                // Initialization of $m/m_{\mathsf{step}}$ values to zero
foreach experiment $\in \{1, \ldots, M\}$ do
    PMF$[t][x] = 0, \forall t \in \mathbb{F}_2^n, x \in \mathcal{X}$
    foreach step $\in \{1, \ldots, \widetilde{q}/\widetilde{q}_{\mathsf{step}}\}$ do
        for $i \in \{1 + (\mathsf{step} - 1) \times \widetilde{q}_{step}, \ldots, \mathsf{step} \times \widetilde{q}_{step}\}$ do
            PMF$[t_i][x_i]$ += $1$
        end
        for $k \in \mathcal{K}$ do // Key enumeration
            $+ +$ score$_k$ $=$ MIA-DISTINGUISHER(PMF, $k$) $+ +$
            // See MIA-Distinguisher
            score$_k$ $=$ MIA-DISTINGUISHER(PMF, $k$)          // Function at page 116
        end
        if arg max$_{k \in \mathcal{K}}$ score$_k$ $= k^*$ then
            $\widehat{\mathrm{SR}}_{k^*}$[step] += $1/M$
        end
    end
end
return $\widehat{\mathrm{SR}}_{k^*}$                          // The empirical Success Rate
```

- The values of $\widetilde{\mathbb{P}}(x)$ and $\widetilde{q}$ do not impact the result, hence they are not computed. This means that lines 4–8 of Alg. 2 are not part of Alg. MIA-Distinguisher, and that in the accumulation of $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$ at line 15 of Function MIA-Distinguisher, the terms $\widetilde{\mathbb{P}}(x)$ and $\widetilde{q}$ (present at line 25 of Alg. 2), are simply dropped.

- In the line 17 of MIA-Distinguisher, we subtract the term which corresponds to the denominator $\widetilde{\mathbb{P}}(y)$ of line 25 of Alg. 2. Notice that all the parameters of the logarithms are now integers. We can thus tabulate the logarithms in a table $\texttt{log}[i]$, for $i = 1, \ldots, \widetilde{q}$. Incidentally, the basis of the logarithm can be arbitrarily chosen. If $\widetilde{q}$ is really large, say larger than 10 millions, then $\texttt{log}[i]$ can be precomputed from all $i < 10^6$, and evaluated otherwise, since anyhow the call of `log` for large values is restricted to the case of $\widetilde{\mathbb{P}}(y)$ which is expected to be way larger than any $\widetilde{\mathbb{P}}(x, y)$.

6.5.3 Standard computation algorithm for MIA

The standard computation for MIA unfolds as in Alg. 4. This algorithm outputs exactly the same as Alg. 2 but is slower for two reasons:

1. All the $\widetilde{q}$ samples are scanned for each key hypothesis;

2. Probability mass functions are normalized. Now, divisions are costly, and also they require a conversion from integer to floating point numbers;

6.6 Conclusion

We derived MIA anew as the distinguisher which maximizes the success rate when the exact probabilities are replaced by online estimations. This suggests that MIA is an interesting alternative when the attacker is not able to exactly determine the link between the measured leakage and the leakage model. This situation can either result from an unknown deterministic part or from an unknown noise distribution. We have proved that, if the number of traces is greater than the number of possible values of x and y, the MIA is faster in terms of logarithm computations.

We have presented two practical case-studies in which MIA can indeed be more efficient than CPA or LRA. The first scenario is for non-Gaussian noise but known deterministic leakage model. The second scenario is for Gaussian noise with unknown deterministic leakage model,

Function MIA-Distinguisher

input : $\mathsf{PMF}[t][x] \in \mathbb{N}^{\mathbb{F}_2^n \times \mathcal{X}}$, a non-normalized bi-dimensional histogram,
$k \in \mathcal{K}$, a key guess

output : A score, affinely proportional to MIA (computed as per Eqn. (6.10)), with the same (irrelevant) affine scaling factors for all the keys

```
$\forall x \in \mathcal{X}, y \in \mathcal{Y}$, $\widetilde{\mathbb{P}}(x,y) = 0$        // $\widetilde{\mathbb{P}}(x,y)$ holds $m\widetilde{\mathbb{P}}(x,y)$, cf. Eqn. (6.8)
for $t \in \mathbb{F}_2^n$ do
    for $x \in \mathcal{X}$ do
        $\widetilde{\mathbb{P}}(x, \varphi(f(k,t)))$ += $\mathsf{PMF}[t][x]$        // $y = \varphi(f(k,t))$, cf. Eqn. (6.1)
    end
end
$\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)) = 0$        // Quantity actually affine with $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$
for $y \in \mathcal{Y}$ do
    $\widetilde{\mathbb{P}}(y) = 0$        // $\widetilde{\mathbb{P}}(y)$ holds $m\widetilde{\mathbb{P}}(y)$, cf. Eqn. (6.7)
    for $x \in \mathcal{X}$ do
        $\widetilde{\mathbb{P}}(y)$ += $\widetilde{\mathbb{P}}(x,y)$
    end
    for $x \in \mathcal{X}$ do
        if $\widetilde{\mathbb{P}}(x,y) \neq 0$ then        // $\widetilde{\mathbb{P}}(x,y) \neq 0 \implies (\widetilde{\mathbb{P}}(x) \neq 0 \wedge \widetilde{\mathbb{P}}(y) \neq 0)$
            $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$ += $\widetilde{\mathbb{P}}(x,y) \cdot \mathtt{log}[\widetilde{\mathbb{P}}(x,y)]$
        end
        $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$ −= $\widetilde{\mathbb{P}}(x) \cdot \mathtt{log}[\widetilde{\mathbb{P}}(y)]$
    end
end
return $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$
```

Algorithm 4: Standard computation algorithm for MIA

```
input  : $\widetilde{\mathbf{x}}$ a set of $m$ traces which take discrete values,
         $\widetilde{\mathbf{t}}$ a corresponding set of $m$ plaintexts/ciphertexts
output : $(\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)))_{k \in \mathcal{K}}$
$\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)) = 0, \forall x \in \mathcal{X}, k \in \mathcal{K}$
for $k \in \mathcal{K}$ do // Key enumeration
    // From $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$, build a hash table PMF$[y][x]$ (i.e., an histogram)
    for $i \in \{1, \ldots, m\}$ do
        PMF$[\varphi(f(k, t_i))][x_i]$ += 1                // $y = \varphi(f(k,t))$, cf. Eqn. (6.1)
    end
    $\forall x \in \mathcal{X}, y \in \mathcal{Y}, \widetilde{\mathbb{P}}(x, y) = 0$    // cf. Eqn. (6.8)
    for $y \in \mathcal{Y}$ do
        for $x \in \mathcal{X}$ do
            $\widetilde{\mathbb{P}}(x, y) = {}^1\!/\!_m$ PMF$[y][x]$
        end
    end
    for $y \in \mathcal{Y}$ do
        $\widetilde{\mathbb{P}}(y) = 0$                        // cf. Eqn. (6.7)
        for $x \in \mathcal{X}$ do
            $\widetilde{\mathbb{P}}(y)$ += $\widetilde{\mathbb{P}}(x, y)$
        end
    end
    for $x \in \mathcal{X}$ do
        $\widetilde{\mathbb{P}}(x) = 0$                        // cf. Eqn. (6.7)
        for $y \in \mathcal{Y}$ do
            $\widetilde{\mathbb{P}}(x)$ += $\widetilde{\mathbb{P}}(x, y)$
        end
    end
    for $y \in \mathcal{Y}$ do
        for $x \in \mathcal{X}$ do
            // Nota bene: $(\widetilde{\mathbb{P}}(x) = 0 \lor \widetilde{\mathbb{P}}(y) = 0) \implies \widetilde{\mathbb{P}}(x, y) = 0$
            if $\widetilde{\mathbb{P}}(x) \neq 0$ and $\widetilde{\mathbb{P}}(y) \neq 0$ then
                $\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k))$ += $\widetilde{\mathbb{P}}(x, y) \log_2 \left( \frac{\widetilde{\mathbb{P}}(x,y)}{\widetilde{\mathbb{P}}(x)\widetilde{\mathbb{P}}(y)} \right)$
            end
        end
    end
end
return $(\widetilde{I}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}(k)))_{k \in \mathcal{K}}$   // As in Eqn. (6.10)
```

where one leverages a challenging leakage function which results in failure for CPA, and in harsh regression using LRA. Incidentally, this example is in line with the work carried out by Whitnall and Oswald [88] where a notion of relative margin is used to compare attacks. Our findings go in the same direction using the success rate as a figure of merit to compare attacks.

Finally, we extended the computation trick given for CPA to MIA avoiding the histogram estimation of conditional probabilities for each sub key individually, improving the speed of the computation.

We note that all our results are φ-dependent. It seems obvious that the closer we are to the actual leakage, the better the success rate will be. An open question is to find an analytic way to determine the function model that will provide the highest success rate.

Last, we note that our analysis is monovariate: we consider a leakage which consists in only one value. A future work would be to extend our results to mutivariate attacks.

Another topic of research is to carry out practical examples where MIA beats CPA. An viable option would be the exploitation of some specific timing attacks where the behaviour of the processor changes at every start-up.

Part IV

Practical Issues With Timing Attacks

Introduction

The field of cryptography is currently very sensitive as it deals with data protection and safety. Thus, in order to assess the security of cryptographic devices, it is crucial to know and test their weaknesses. For example, the Advanced Encryption Standard (AES) [26] is renowned as trustworthy from a mathematical point of view—there is currently no realistic way to cryptanalyze the AES-128. However, it is possible to break the 128-bit secret key byte by byte using side-channel analysis (SCA). SCA exploits the physical fact that the secret key leaks some information out of the device boundary through various "side-channels" such as power consumption or *timing*—number of clock cycles to perform a given operation. These leakages, correctly analyzed by SCA, yield the secret key of a device.

A good side-channel attack needs a good leakage model. Timing, for example, can be modeled easily when the implementation is unbalanced: Several successful attacks [13, 14, 75, 76] exploit a timing leakage in the conditional extra-reductions of Montgomery modular multiplications. Some conditional operations can also happen in AES, e.g. in field operations, as for instance discussed in [27, Alg. 1].

Even when the code is balanced—a recommended secure coding practice—some residual unbalances in timing can result from the hardware which executes the code. Indeed, processors implement speed optimization mechanisms such as memory caching and out-of-order execution. As a consequence, it is not possible to predict with certainty how timing leaks information. The attacker is then led to make predictions about the way the device leaks.

In this part, we consider side-channel attacks that are performed in two phases:

1. a *profiling phase* where the attacker accumulates leakage from a device with a known secret key;

2. an *attacking phase* where the attacker accumulates leakage from the device with an unknown secret key.

This type of attack is known as a *template attack* [20]. It has been shown [20] to be very efficient under three conditions: (a) leakage samples are independent and identically distributed (i.i.d.); (b) the noise is additive white Gaussian; and (c) the secret key leaks byte by byte, which enables a divide-and-conquer approach. For some side-channels, such as power or electromagnetic radiations, condition (b) is met in practice. However, for timing attacks, the noise cannot be Gaussian as timing is discrete. Moreover, the noise source is non-additive in this case, since it arises from complex replacement policies in caches and processor-specific on-the-fly instructions reordering.

The first proposed profiled timing attack is the seminal timing attack of Kocher [44]. The same methodology can be used on AES, as noted by Bernstein in 2005 [5]. Further works used the same method [8, 69, 87]. To our best knowledge, all these works consist in profiling moments, such as the average timing under a given plaintext and key. However, it is known [20] that the best attacks should use maximum likelihood[1]

In this part, as illustrated in Tab. 6.1, we focus on a profiling where the distribution is characterized and used as such, and is not reduced to its moments. The attacker computes distributions using histogram methods. These distributions are then used to recover the correct secret key.

Table 6.1: State-of-the-art on profiled timing attacks

Profiling method	**Reference articles**
Moments	[5, 8, 69, 87]
Distributions	This part (Caution about *empty bins*)

The discrete nature of timing leakage leads to an *empty bin* issue which appears when a data value in the attacking phase has never been seen during the profiling phase. Based on profiling only, this data should have a zero probability, which can be devastating for the attack. One known workaround is to use kernel distribution methods [64] to estimate probabilities since the smoothing can be such that no empty bins remain. This method can however be seen as a postprocessing in existing information. This alters therefore the data. In addition, this method has very large computational complexity and its performance highly depends on *ad-hoc* choices of several parameters such as kernel type and bandwidth. Moreover the estimation via the kernel method depends on other parameters such as the choice of the kernel and the size of the

[1] We will explain in Subsec. 8.2.2 that in practice, maximum likelihood might not always perform better than moment-based distinguishers in ideal situations (no noise), because the learning stage for probability mass functions demands too many traces; besides an imperfect profiling is very detrimental to maximum likelihood distinguishers, and affects less the moment-based distinguishers. However, in non-ideal situations, e.g., in the presence of *random delay* kind of noise, maximum likelihood remains robust, where the model-based distinguishers collapse (since they are value-based).

kernel. In this part, we have decided to keep information as it comes as we focus on information theoretic distinguishers.

Contributions In this part, we show that even when all abovementioned requirements (a), (b), and (c) are not present, timing attacks with incomplete profiling can be achieved successfully by adapting the maximum likelihood distinguisher and keeping the histogram method for probabilities estimation. We build six different distinguishers, which are all good answers to the empty bin issue. For some of them, new histograms are built, such that the empty bin issue totally disappears. Furthermore, we compare these distinguishers and show which one of them is the best in every specific context. We underline that, in practice, for a moderate profiling with 256 000 offline measurements, the *soft drop* and the combined *offline-online profiling* approaches are clearly the two best strategies: the AES key is typically extracted with only about 2 000 online measurements, i.e., a complete break in about 0.2 ms. Finally, we provide some theoretical results proving how optimal some of the distinguishers can be.

Organization The part is organized according to the following structure. In Chapter 7, we first provide the mathematical tools to deal with the empty bins issue. Section 7.1 provides mathematical tools to understand distinguishers and notations. Section 7.2 introduces new distinguishers that are suitable in the context of empty bins. Section 7.3 provides simulations for these distinguishers. In Chapter 8 we focus on the timing leakages for a specific implementation. Section 8.1 investigates real attacks on an ARM processor. Interestingly, all proposed distinguishers work, albeit with very noticeably different performances. In section 8.2, some interpolations of the obtained results in the presence of external measurement noise are derived. Section 8.3 concludes for both chapters.

Chapter 7

Methods to Solve the Empty-Bin Issue

A part of the work of this chapter has been presented at HASP 2018.

Contents

7.1 Mathematical Derivations

The mathematical notations and assumptions presented here will also be used in Chapter 8.

7.1.1 Notations and Assumptions

We consider a side-channel attack with a profiling stage and use the following notations:

- during the profiling phase, a vector $\widehat{\mathbf{t}}$ of $\widehat{q}$ text bytes is sent and the profiler garners a vector of $\widehat{\mathbf{x}}$ measurements;
- during the attacking phase, a vector $\widetilde{\mathbf{t}}$ of $\widetilde{q}$ text bytes is sent and the attacker gathers a vector $\widetilde{\mathbf{x}}$ of leakage measurements—also customarily known as *traces*;
- we use simplified notations $\mathbf{t}$, q and $\mathbf{x}$ when discussing either profiling data or attacking data;
- the probability of a vector $\mathbf{x}$ with i.i.d. components x_i is denoted by $\mathbb{P}(\mathbf{x}) = \prod_i \mathbb{P}(x_i)$;
- we define the following sets:
  1. $\widehat{\mathcal{X}}$, $\widehat{\mathcal{T}}$, $\widetilde{\mathcal{X}}$ and $\widetilde{\mathcal{T}}$ are the sets of possible values of components $\widehat{x}$, $\widehat{t}$, $\widetilde{x}$ and $\widetilde{t}$, respectively;
  2. $\mathcal{X} = \widehat{\mathcal{X}} \cup \widetilde{\mathcal{X}}$ and $\mathcal{T} = \widehat{\mathcal{T}} \cup \widetilde{\mathcal{T}}$;
  3. $\mathcal{K}$ is the set of all possible values for the key k.
- k and t are made of n bits (in particular, they are “bytes” when $n = 8$).

Here all sample components of one vector are i.i.d. and belong to some discrete set. Typically, $\mathcal{X}$ is a finite subset of $\mathbb{N}$ and $\mathcal{T}$ is equal to $\{0,1\}^n$.

In the profiling stage, the secret key $\widehat{k}^*$ is known and variable. In the attacking phase, the secret key $\widetilde{k}^*$ is unknown but fixed. Further, we assume that x_i depends only on t_i and k^* for all $i = 1, 2, \ldots, q$, in the form:

$$x_i = \psi(t_i \oplus k^*) \qquad (i = 1, 2, \ldots, q) \tag{7.1}$$

where $\oplus$ is the XOR (exclusive or) operator and ψ is an unknown function which may contain noise, masking and other hidden parameters[1].

Furthermore, in this part, we use of the notation $n_{x,t}$ to denote the number of occurrences of (x, t). Thus we can write

$$\begin{aligned} \widehat{n}_{x,t} &= \textstyle\sum_{i=1}^{\widehat{q}} \mathbb{1}_{\widehat{x}_i = x, \widehat{t}_i = t} & \widehat{n}_x &= \textstyle\sum_{i=1}^{\widehat{q}} \mathbb{1}_{\widehat{x}_i = x}, \\ \widetilde{n}_{x,t} &= \textstyle\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{t}_i = t} & \widetilde{n}_x &= \textstyle\sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x}. \end{aligned}$$

where $\mathbb{1}_A = 1$ if A is true, $= 0$ otherwise.

Definition 7.1 (Probabilities)**.** We define three[2] different types of probabilities $\mathbb{P}$, $\widehat{\mathbb{P}}$ and $\widetilde{\mathbb{P}}$. $\mathbb{P}$ is the actual (real) underlying probability distribution, but it is generally not available and has to be estimated by either $\widehat{\mathbb{P}}$ or $\widetilde{\mathbb{P}}$.

- $\widehat{\mathbb{P}}$ is computed using the profiling data:

$$\widehat{\mathbb{P}}(x, t) = \frac{1}{\widehat{q}} \sum_{i=1}^{\widehat{q}} \mathbb{1}_{\widehat{x}_i = x, \widehat{t}_i = t} = \frac{\widehat{n}_{x,t}}{\widehat{q}}, \tag{7.2}$$

$$\widehat{\mathbb{P}}(x) = \frac{1}{\widehat{q}} \sum_{i=1}^{\widehat{q}} \mathbb{1}_{\widehat{x}_i = x} = \frac{\widehat{n}_x}{\widehat{q}}. \tag{7.3}$$

- $\widetilde{\mathbb{P}}$ is computed using the attacking data:

$$\widetilde{\mathbb{P}}(x, t) = \frac{1}{\widetilde{q}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x, \widetilde{t}_i = t} = \frac{\widetilde{n}_{x,t}}{\widetilde{q}}, \tag{7.4}$$

$$\widetilde{\mathbb{P}}(x) = \frac{1}{\widetilde{q}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widetilde{x}_i = x} = \frac{\widetilde{n}_x}{\widetilde{q}}. \tag{7.5}$$

[1] The AES meets the secret and the text byte through a xor (`SubBytes`) executed in a fixed number of clock cycles. However, the rest of the AES consists in table look-ups and other miscellaneous operations which are difficult to model and need different amounts of time to execute, hence the use of unknown function ψ.

[2] For the sake of evading the empty bin issue, we will also introduce yet another notation "$\mathbb{P}_\alpha$" in section 7.2.1 (Equation (7.15)).

In practice, as the secret key leaks through the function via a XOR (Equation (7.1)), we shall often consider $\mathbb{P}(x, t \oplus k)$.

For a fair comparison between distinguishers, Standaert et al. [82] have put forward the *success rate* as a measure of efficiency of a given distinguisher.

Definition 7.2 (Success Rate)**.** The success rate SR is probability, averaged over all possible keys, of obtaining the correct key.

$$\mathsf{SR} = \frac{1}{2^n} \sum_{k^*=0}^{2^n-1} \mathbb{P}_{k^*}(\widetilde{k} = k^*), \tag{7.6}$$

where $\widetilde{k}$ is the key guess obtained by the distinguisher during the attack.

It has been proven [39, Theorem 1, equation (3)] that for equiprobable keys the optimal distinguisher maximizes likelihood:

$$\mathcal{D}_{\mathsf{Optimal}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg \max_{k \in \mathcal{K}} \mathbb{P}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k). \tag{7.7}$$

In real life, however, the attacker does not know the leakage model perfectly and thus $\mathbb{P}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k)$ is not available. In order to get an estimation of $\mathbb{P}$, we use the profiling data to build $\widehat{\mathbb{P}}$ defined in Equation (7.2). This is the classical *template attack*. The distinguisher becomes

$$\mathcal{D}_{\mathsf{Template}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg \max_{k \in \mathcal{K}} \widehat{\mathbb{P}}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k). \tag{7.8}$$

This distinguisher *is no longer optimal* as it does not use the real distribution $\mathbb{P}$. However, if profiling tends to exhaustivity, $\widehat{\mathbb{P}}$ and $\mathbb{P}$ will be very close since by the law of large numbers,

$$\forall x, t \quad \widehat{\mathbb{P}}(x, t) \underset{\widehat{q} \to \infty}{\longrightarrow} \mathbb{P}(x, t).$$

Moreover, we notice that non-optimality is not the only issue with template attacks in the context of discrete leakage. The attacker also faces the problem that the attack is ill-formed.

In practice, it is convenient to use the logarithm $\arg \max_{k \in \mathcal{K}} \log \widehat{\mathbb{P}}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k)$. In fact, since the samples are i.i.d., we have

$$\mathbb{P}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k) = \prod_{i=1}^{\widetilde{q}} \mathbb{P}(\widetilde{x}_i | \widetilde{t}_i \oplus k) \quad \text{and} \quad \widehat{\mathbb{P}}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k) = \prod_{i=1}^{\widetilde{q}} \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k).$$

Therefore, the attacker computes

$$\mathcal{D}_{\mathsf{Template}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg \max_{k \in \mathcal{K}} \sum_{i=1}^{\widetilde{q}} \log \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) \tag{7.9}$$

where the logarithm is used to transform products into sums for a more reliable computation. However, we would like to avoid empty bins for which $\widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) = 0$; otherwise, Equation (7.9) would not be well defined.

7.1.2 About Empty Bins

The empty bin issue appears when there exists $i \in \{1, \dots, \widetilde{q}\}$ and $k \in \mathcal{K}$ such that $\widetilde{\mathbb{P}}(\widetilde{x}_i|\widetilde{t}_i \oplus k) > 0$ and $\widehat{\mathbb{P}}(\widetilde{x}_i|\widetilde{t}_i \oplus k) = 0$. This may even happen for the *correct* key hypothesis, leading to a wrong key guess during the attack.

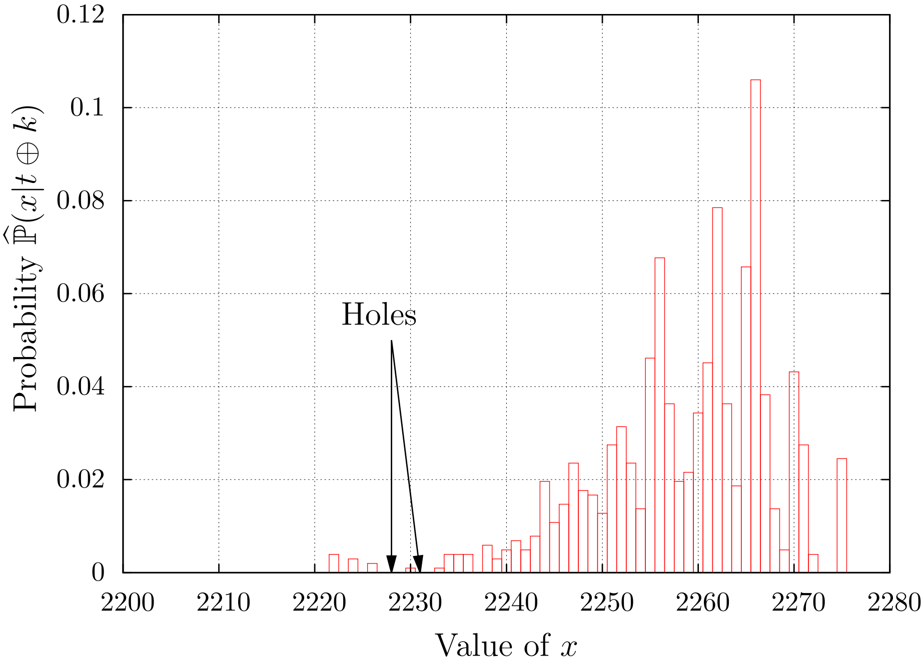


Figure 7.1: Empirical probability $\widehat{\mathbb{P}}(x|t \oplus k)$ for $t = 0$ and $k = 67$ and $\widehat{q} = 2\,560\,000$

Figures 7.1 and 7.2 show how empty bins can look like after a profiling phase[1]. We notice that some parts of the histograms are left blank, some of them indicated by arrows noticed as "holes" in the figures. These timing values x are possible "empty bins". When such a hole is called during the attack, meaning that the attacker gets a trace with corresponding with a hole, we call this an *empty bin*. Notice that no additional "binning" is needed as in the case of continuous distributions. The figures also show that the noise is not Gaussian as can be observed from the shape of the distribution.

The shortcoming of empty bins can be seen when evaluating the likelihood. The attacker encounters a zero probability, which makes the product vanish for the probability of a given key guess, even if many traces are used. As we wrote earlier, the empty bin may appear even for the correct key guess in template attacks, leading to a null success rate if not taken into account and

[1] Figures obtained with the STM Discovery Board presented in Section 8.1. The unit of x is the "clock cycle".

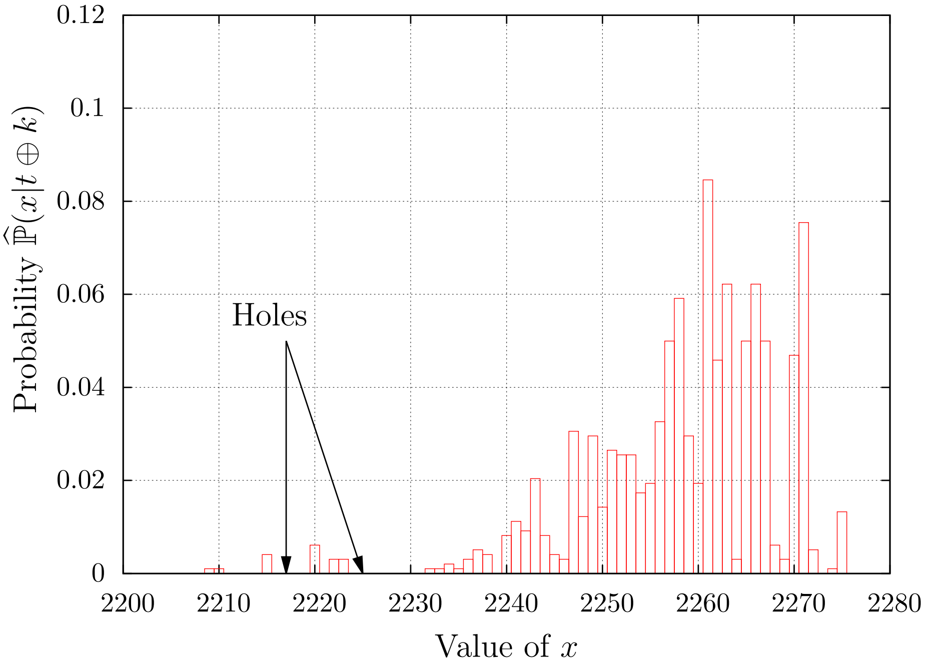


Figure 7.2: Empirical probability $\widehat{\mathbb{P}}(x|t \oplus k)$ for $t = 0$ and $k = 149$ and $\widehat{q} = 2\,560\,000$

not well treated. As an example, the number of empty bins for the practical example presented Section 8.1 for the *correct key guess* is around 500 for a poor learning phase and around 50 for a good learning phase. This multiplication by zero is not inherent to the attack; it is rather a profiling artifact. In fact, with more profiling traces, the empty bin would likely be populated. Thus, the empty bin issue is a mere side-effect of insufficient profiling, which results in an attack failure if it is encountered in the computation of the likelihood of the correct key.

7.2 Distinguishers which Tolerate Empty Bins

7.2.1 Building Distributions or Models

Before presenting the novel distinguishers in Subsection 7.2.2, we need to define yet another other type of distribution known as a Dirichlet *a posteriori* in a Bayesian approach.

The Dirichlet A Posteriori

In order to avoid zero probabilities, we use a method based on Dirichlet Prior calculations [33, Section 1]. This method leads to a new distribution denoted by $\overline{\mathbb{P}}_\alpha$, where $\alpha > 0$ is a user-defined parameter whose value (typically $= 1$) will be discussed next.

Let $\mathcal{X}$ be the set of possible values for x and $\mathcal{T}$ be the set of possible values for t. For any x, we set $p_{x,t} = \mathbb{P}(x,t)$ their joint probability and $\mathbf{p} = (p_{x,t})_{x,t}$. Prior to obtaining any trace, $p_{x,t}$ is completely unknown and we consider a Bayesian approach to estimate $p_{x,t}$.

1. We consider the following *a priori*: without further information, we suppose that for all x, t,

$$\bar{\mathbb{P}}_\alpha(x,t) = \frac{\alpha_{x,t}}{\sum_{x',t'} \alpha_{x',t'}},$$

where $\alpha_{x,t} > 0$ is an *a priori* parameter. To simplify, we may choose $\alpha_{x,t} = \alpha$ constant for all x, t. Let us suppose that $\mathbf{p}$ follows a Dirichlet (prior) distribution, whose probability density function is

$$f(\mathbf{p}) = \frac{\Gamma(\sum_{x,t} \alpha_{x,t})}{\prod_{x,t} \Gamma(\alpha_{x,t})} \prod_{x,t} p_{x,t}^{\alpha_{x,t}-1}, \tag{7.10}$$

where Γ is the Gamma function defined for $x > 0$ as

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} \, \mathrm{d}t. \tag{7.11}$$

The Dirichlet distribution can also be written as

$$f(\mathbf{p}) = \mathcal{N}_\alpha \prod_{x,t} p_{x,t}^{\alpha_{x,t}-1}, \tag{7.12}$$

where $\mathcal{N}_\alpha = \frac{\Gamma(\sum_{x,t} \alpha_{x,t})}{\prod_{x,t} \Gamma(\alpha_{x,t})}$ is a normalization factor. Notice that the prior distribution is *uniform* when $\alpha_{x,t} = \alpha = 1$ for all x, t.

2. Then suppose we know $\widehat{\mathbf{x}}$, $\widetilde{\mathbf{x}}$, $\widehat{\mathbf{t}}$ and $\widetilde{\mathbf{t}}$. We can now compute the *a posteriori* probability

$$\mathbb{P}(x, t | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \int f(\mathbf{p}, x, t | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) \, \mathrm{d}p.$$

By Bayes' rule,

$$f(\mathbf{p}, x, t | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \mathbb{P}(x, t | \mathbf{p}, \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) f(\mathbf{p} | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}).$$

As components x_i and t_i are i.i.d., we can write

$$\begin{aligned} f(\mathbf{p}, x, t | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) &= \mathbb{P}(x, t | \mathbf{p}) \cdot f(\mathbf{p} | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}, t) \\ &= p_{x,t} \cdot f(\mathbf{p} | \widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) \end{aligned}$$

Again by Bayes' rule,

$$\begin{aligned} f(\mathbf{p}|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) &= \frac{\mathbb{P}(\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}, \widehat{\mathbf{t}}|\mathbf{p})f(\mathbf{p})}{\mathbb{P}(\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}, \widehat{\mathbf{t}})} \\ &= \frac{\prod_{x',t' \in \mathcal{X}\times\mathcal{T}} p_{x',t'}^{\widehat{n}_{x',t'}+\widetilde{n}_{x',t'}(k)}}{\mathbb{P}(\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}, \widehat{\mathbf{t}})} f(\mathbf{p}) \\ &= \frac{\mathcal{N}_\alpha}{\mathbb{P}(\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}, \widehat{\mathbf{t}})} \prod_{x',t' \in \mathcal{X}\times\mathcal{T}} p_{x',t'}^{\widehat{n}_{x',t'}+\widetilde{n}_{x',t'}+\alpha_{x',t'}-1}. \end{aligned}$$

We recognize another Dirichlet distribution with parameters $\widehat{n}_{x',t'} + \widetilde{n}_{x',t'} + \alpha_{x',t'}$. Let $\mathcal{N}_{\alpha'} = \frac{\Gamma(\sum_{x',t'} \alpha_{x',t'}+\widetilde{n}_{x',t'}+\alpha_{x',t'})}{\prod_{x,t} \Gamma(\alpha_{x,t}+\widetilde{n}_{x',t'}+\alpha_{x',t'})}$ be the new normalization constant for this distribution. We, finally, obtain

$$f(\mathbf{p}, x, t|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = p_{x,t} \cdot \mathcal{N}_{\alpha'} \prod_{x',t' \in \mathcal{X}\times\mathcal{T}} p_{x',t'}^{\widehat{n}_{x',t'}+\widetilde{n}_{x',t'}+\alpha_{x',t'}-1}. \tag{7.13}$$

Therefore,

$$\mathbb{P}(x, t|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \int p_{x,t} \cdot \mathcal{N}_{\alpha'} \prod_{x',t' \in \mathcal{X}\times\mathcal{T}} p_{x',t'}^{\widehat{n}_{x',t'}+\widetilde{n}_{x',t'}+\alpha_{x',t'}-1} \, \mathrm{d}p.$$

which is known as the Dirichlet *a posteriori.*

3. The integral can be easily expressed in terms of the Gamma function:

$$\mathbb{P}(x, t|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \frac{\Gamma(\sum_{x',t'} \alpha_{x,t} + \widehat{n}_{x',t'} + \widetilde{n}_{x',t'})}{\prod_{x',t'} \Gamma(\alpha_{x,t} + \widehat{n}_{x',t'} + \widetilde{n}_{x',t'})} \times \frac{\prod_{x',t'} \Gamma(\alpha_{x,t} + \widehat{n}_{x',t'} + \widetilde{n}_{x',t'} + \delta_{x,t})}{\Gamma(\sum_{x',t'} \alpha_{x,t} + \widehat{n}_{x',t'} + \widetilde{n}_{x',t'} + \delta_{x,t})}$$

which simplifies to

$$\mathbb{P}(x, t|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \frac{\widehat{n}_{x,t} + \widetilde{n}_{x,t} + \alpha_{x,t}}{\widehat{q} + \widetilde{q} + \sum_{x',t'} \alpha_{x',t'}}.$$

This new distribution will now be noted:

$$\overline{\mathbb{P}}_\alpha(x, t) = \mathbb{P}(x, t|\widehat{\mathbf{x}}, \widetilde{\mathbf{x}}, \widehat{\mathbf{t}}, \widetilde{\mathbf{t}}) = \frac{\widehat{n}_{x,t} + \widetilde{n}_{x,t} + \alpha_{x,t}}{\widehat{q} + \widetilde{q} + \sum_{x',t'} \alpha_{x',t'}}. \tag{7.14}$$

It is important to notice that for all $(x, t) \in \mathcal{X} \times \mathcal{T}$, one has $\overline{\mathbb{P}}_\alpha(x, t) > 0$. In other words, $\overline{\mathbb{P}}_\alpha$ has **no empty bin issue**.

4. With $\overline{\mathbb{P}}_\alpha(x, t)$ we can calculate

$$\begin{aligned} \overline{\mathbb{P}}_\alpha(t) = \sum_x \overline{\mathbb{P}}_\alpha(x, t) &= \sum_x \frac{\widehat{n}_{x,t} + \widetilde{n}_{x,t} + \alpha_{x,t}}{\widehat{q} + \widetilde{q} + \sum_{x',t'} \alpha_{x',t'}} \\ &= \frac{\widehat{n}_t + \widetilde{n}_t + \sum_t \alpha_{x,t}}{\widehat{q} + \widetilde{q} + \sum_{x',t'} \alpha_{x',t'}} = \frac{\widehat{n}_t + \widetilde{n}_t + \alpha_t}{\widehat{q} + \widetilde{q} + \sum_{x'} \alpha_{x'}}, \end{aligned}$$

where $\alpha_t = \sum_x \alpha_{x,t}$. The resulting conditional probability[1] is

$$\bar{\mathbb{P}}_\alpha(x|t) = \frac{\bar{\mathbb{P}}_\alpha(x,t)}{\bar{\mathbb{P}}_\alpha(t)} = \frac{\widehat{n}_{x,t} + \widetilde{n}_{x,t} + \alpha_{x,t}}{\widehat{n}_t + \widetilde{n}_t + \alpha_t}. \tag{7.15}$$

The Learned MIA Model

When $\widehat{q}$ is small, the model cannot be profiled accurately, and $\widehat{\mathbb{P}}$ is a bad approximation of $\mathbb{P}$. However, these profiled values $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{t}}$ can still be useful, yet they require a more robust distinguisher.

Distinguishers that compute models using profiling have already been proposed. For example, [61] computes a correlation on moments. However, correlations analysis may be sensitive to model errors [85]. Mutual Information Analysis (MIA) yields a distinguisher that can be robust when models are not perfectly known [85, Section 4], but it requires at least a vague estimation of the leakage model.

Since our function ψ is unknown, we can create a first-order model $\widehat{\psi}$ with the profiled data as

$$\widehat{\psi}(t \oplus \widehat{k}^*) = \mathsf{Step}\Big(\frac{1}{n_t} \sum_{i \text{ s.t. } \widehat{t}_i = t} \widehat{x}_i\Big) \qquad (\forall t \in \mathcal{T}). \tag{7.16}$$

The Step function is a function that ensures the non-injectivity of the model. The simplest way to define Step is the following:

$$\mathsf{Step}(x) = \frac{\lfloor d \cdot x \rfloor}{d} \qquad (x \in \mathbb{R})$$

where $d > 0$—the greater d, the smaller the step size. This parameter d has to be small enough in order to make the model non-injective [34, Sec. 4.1]. In our case, we choose, for all our experiments, $d = 1$. With such a model, it is possible to compute a MIA, which successfully distinguishes the correct key.

7.2.2 Robust distinguishers

In this subsection, we present six distinguishers that tackle null probabilities. Some of these solutions seem quite obvious while others are deduced from the notions presented in the preceding Subsection 7.2.1.

[1] We should normally have used the notation $\widetilde{\widetilde{\mathbb{P}}}_\alpha$ instead of $\bar{\mathbb{P}}_\alpha$, but we found this too heavy and confusing; hence the use of $\bar{\mathbb{P}}_\alpha$.

❶ Hard Drop Distinguisher

The first naive method consists in removing all the traces which, for any key guess, have a zero probability.

Definition 7.3 (Hard Drop Distinguisher)**.** The hard drop distinguisher is defined as followed:

$$\mathcal{D}_{\mathsf{Hard}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \log \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k), \tag{7.17}$$

where set $\mathcal{I}$ is defined as

$$\mathcal{I} = \left\{ i \in \{1, \ldots, \widetilde{q}\} \mid \forall k \in \mathcal{K},\ \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) > 0 \right\}. \tag{7.18}$$

Recall that $\widehat{\mathbb{P}}$, defined in Equation (7.2), is an empirical histogram estimated on profiled data $\widehat{\mathbf{x}}$ (along with corresponding texts $\widehat{\mathbf{t}}$).

The Hard Drop Distinguisher, as the name indicates, drops some data. In very noisy cases, it may even drop most of the data.

❷ Soft Drop Distinguisher

The second possibility is to drop values only for some keys. However, it has to be done carefully because dropping a value in a product implicitly implies a probability value of one. For this reason, instead of removing the trace, we replace the zero probability by a constant which is smaller than the smallest probability.

Definition 7.4 (Soft Drop Distinguisher)**.** We define the Soft Drop Distinguisher as

$$\mathcal{D}_{\mathsf{Soft}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_{k \in \mathcal{K}} \sum_{i \text{ s.t. } \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) > 0} \log \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) + \sum_{i \text{ s.t.} \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i, k) = 0} \log \gamma, \tag{7.19}$$

where $\gamma \in \mathbb{R}_+^*$ is a constant such that $\forall i, k \in \{1, \ldots, \widetilde{q}\} \times \mathcal{K}, \quad \gamma \leq \widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k)$. This means that we penalize data with zero probability. The smaller γ, the harder the penalty.

The choice of parameter γ is thus important in order to get a fair result for the distinguisher. If we choose $\gamma \geq \frac{1}{\widehat{q}}$, the penalty may be greater than the smallest strictly positive probability. This case would mean that the penalty is less important than some licit probabilities. On the other hand, choosing γ smaller than $\frac{1}{\widehat{q}}$ means a very strong penalty. In this case, the limit when $\gamma \to 0$ is a distinguisher for which only the number of empty bins is really matters. This leads to the *Empty Bin Distinguisher* presented next in Definition 7.8.

❸ The Dirichlet Prior Distinguisher

The Dirichlet Prior Distinguisher uses the Dirichlet *a posteriori* distributions presented in Subsection 7.2.1.

Definition 7.5 (The Dirichlet Distinguisher)**.** We define the Dirichlet Distinguisher as:

$$\mathcal{D}_{\mathsf{Dirichlet}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_{k \in \mathcal{K}} \overline{\mathbb{P}}_{\alpha}(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k). \tag{7.20}$$

Remark 7.1. As can be seen in the construction of the Dirichlet *a posteriori*, the Dirichlet distinguisher is α-dependent. It is important to evaluate the influence of α over the success rate. In practice, $\alpha = 1$ seems a natural choice since the corresponding prior is uniform, which minimizes the impact of the *a priori*. In contrast, another value of α like $1/2$ can be interpreted as an *a priori* bin count. We may also consider scenarios where $\alpha \approx 0$ to have the least possible impact to the modified values of the histogram.

❹ Offline-Online Profiling

The Dirichlet Prior Distinguisher is set by α. As we discussed in Remark 7.1, we can choose any α so long as it is strictly positive (the Dirichlet distribution would not be defined if $\alpha = 0$). However, it is interesting to study its asymptotical behavior as α vanishes:

$$\lim_{\alpha \to 0} \overline{\mathbb{P}}_{\alpha}(x|t) = \frac{\widehat{n}_{x,t} + \widetilde{n}_{x,t}}{\widehat{n}_t + \widetilde{n}_t}.$$

This distribution can be denoted as $\overline{\mathbb{P}}_0(x|t)$ and resembles a profiling stage that would start offline and continue online.

Definition 7.6 (Offline-Online Profiling)**.** The Offline-Online Profiled (OOP) distinguisher is defined as:

$$\mathcal{D}_{\mathsf{OOP}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_{k \in \mathcal{K}} \overline{\mathbb{P}}_0(\widetilde{\mathbf{x}} | \widetilde{\mathbf{t}} \oplus k) \tag{7.21}$$

The OOP distinguisher seems easier than the Dirichlet prior distinguisher since α is no longer in use. Of course, it also solves the empty bin issue since for all $(x, t) \in \mathcal{X} \times \mathcal{T}$, one has $\overline{\mathbb{P}}_0(x, t) > 0$.

❺ Learned MIA Distinguisher

The Learned MIA Distinguisher is constructed with the profiled model function $\widehat{\psi}$ presented in Eqn. (7.16) of Subsection 7.2.1.

Definition 7.7 (The Learned MIA Distinguisher)**.**
The Learned MIA Distinguisher is defined as:

$$\mathcal{D}_{\mathsf{MIA\_Learned}} = \arg \max_{k \in \mathcal{K}} \widetilde{\mathrm{I}} \left(\widetilde{\mathbf{x}}; \widehat{\psi}(\widetilde{\mathbf{t}} \oplus k) \right), \tag{7.22}$$

where $\widetilde{\mathrm{I}}$ is the empirical mutual information [34].

❻ Empty Bin Distinguisher

The empty bin Distinguisher is yet another intuitive solution based on the idea that instead of avoiding null probabilities, we may take only these into account. It is the key guess with the least number of null probabilities that "should" be the correct key.

Definition 7.8. The Empty Bin Distinguisher is defined as:

$$\mathcal{D}_{\mathsf{Empty\_Bin}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg \min_{k \in \mathcal{K}} \sum_{i=1}^{\widetilde{q}} \mathbb{1}_{\widehat{\mathbb{P}}(\widetilde{x}_i | \widetilde{t}_i \oplus k) = 0}. \tag{7.23}$$

The Empty Bin Distinguisher assumed that missing data contain more information than actual (measured) data. More precisely, a drop should normally not happen unless the guessed key is wrong; hence, the key guess with the least drops should be the correct key. Obviously, this distinguisher is not effective anymore if no drop occurs for at least two key guesses.

Further Remarks All these distinguishers use a profiling phase. Before comparing them, we would like to make *a priori* discussion about their respective efficiency. As the Hard Drop Distinguisher does not take into account some data, we may suppose that it will be the one with the least success rate for a given number of traces. The OOP Distinguisher takes into account two types of data: profiling and attacking data. Therefore, it should be more efficient than other distinguishers. Lastly, we build the Learned MIA Distinguisher in order to prevent model errors, such as inaccurate profiling. In that case, we suppose that Learned MIA should work better with few data during the profiling stage.

7.3 Simulated Results

In this section, we present the results obtained on a simulated model. With these results, we can give a comparison of the proposed distinguishers.

7.3.1 Presentation of the Simulated Model

The simulated model is built as follows:

$$\begin{aligned} x_i &= \mathrm{H_w}(\mathrm{S_{box}}(t_i \oplus k^*)) + u_i \\ &= \varphi(t_i \oplus k^*) + u_i = y_i(k^*) + u_i, \end{aligned} \tag{7.24}$$

where u_i is a discrete uniformly distributed noise $u_i \sim \mathcal{U}(-\sigma, \sigma)$, $\mathrm{S_{box}}$ is the AES substitution box function, and $\mathrm{H_w}$ is the Hamming weight of a byte.

This very simple leakage is used to compare distinguishers in the case the attacker has no information about the model.

Remark 7.2 (Optimal Distinguisher). The optimal distinguisher (7.7) can be easily calculated if the model is perfectly known, as

$$\mathcal{D}_{\mathsf{Optimal}}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{t}}) = \arg\max_{k \in \mathcal{K}} \prod_{i=1}^{\widetilde{q}} \delta_\sigma(\widetilde{x}_i - \mathrm{H_w}(\mathrm{S_{box}}(\widetilde{t}_i \oplus k))), \tag{7.25}$$

where δ_σ is defined such that $\delta_\sigma(x) = 1$ if $|x| \leq \sigma$ and 0 otherwise. In Figures 7.3, 7.4 and 7.5, we include the optimal distinguisher for reference, to show how far the other curves are from the fundamental limit of performance.

By construction, the leakage simulation (7.24) generates some traces with zero probability, but notice that there is no i such that $\mathbb{P}(x_i|t_i, k) = 0$ for the correct key guess. This academic example is useful to compare the distinguishers defined in Section 7.2.

7.3.2 Attack Results

We computed the success rates (7.6) of the various attacks (namely attacks ❶, ❷, ❹, ❺ and ❻ — attack ❸ being less efficient than its limit ❹) for for $\sigma = 24$, $n = 4$ bits, and $\widehat{q}$ ranging from small to high values.

The only difference between Figures 7.3, 7.4, and 7.5, is that we have increased the number of data during the profiling stage. When profiling is bad (Figure 7.3), the best distinguisher is the Offline-Online profiling distinguisher, while the Learned MIA Distinguisher is not as good as was expected. When $\widehat{q} = 1\,600$ (Figure 7.4), all distinguishers improve. Finally, when profiling is good ($\widehat{q} = 4\,000$, Figure 7.5), the best distinguisher is now the Empty Bin distinguisher, followed by the Soft Drop distinguisher and the Offline-Online profiling.

Remark 7.3. In this very special case, we can show that the Empty Bin Distinguisher can accurately approximate the Optimal Distinguisher. Indeed, the actual probability is such that

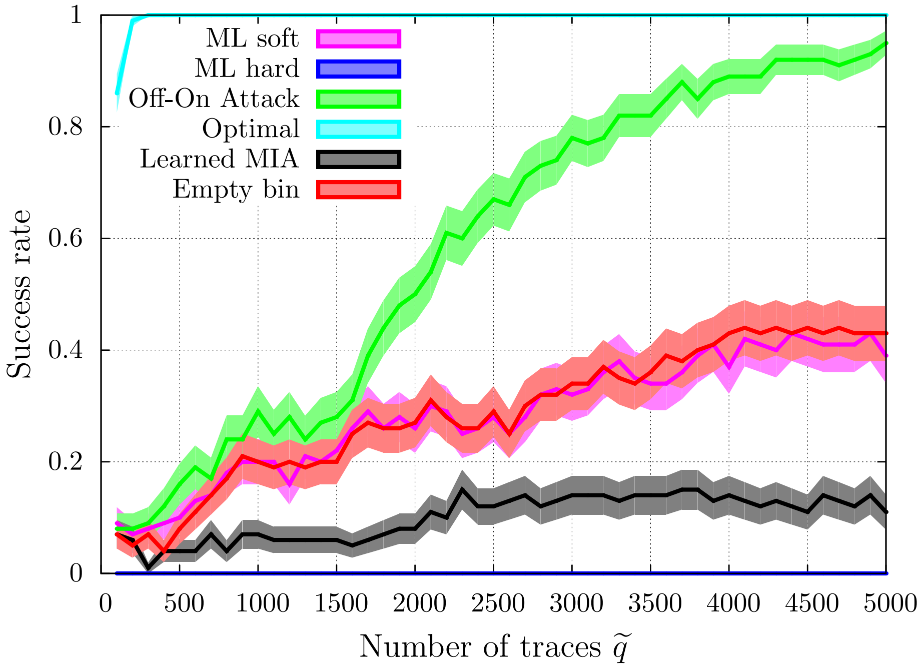


Figure 7.3: SR for $\widehat{q} = 320$ and $\sigma = 24$ on synthetic measurements

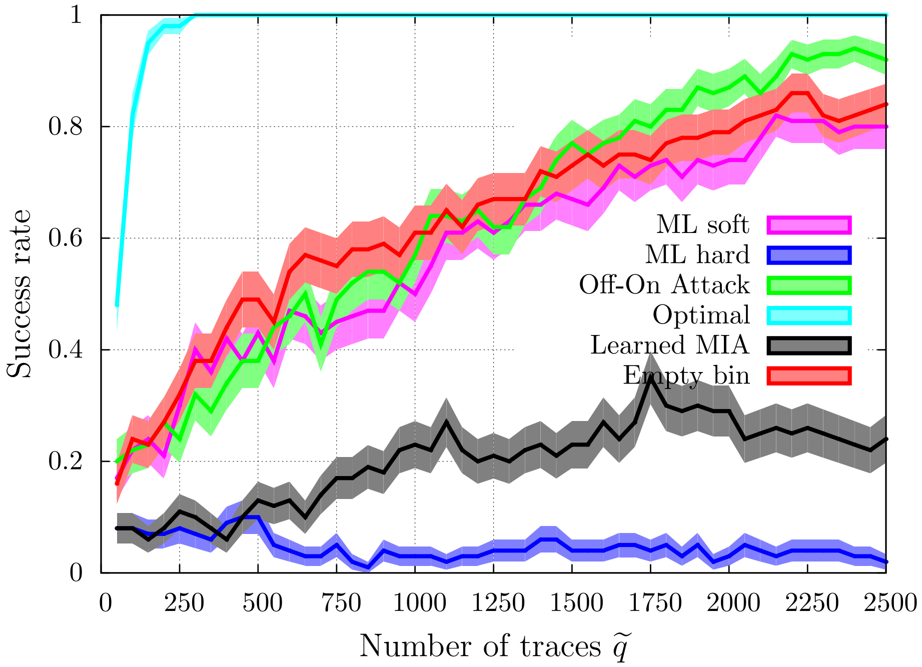


Figure 7.4: SR for $\widehat{q} = 1\,600$ and $\sigma = 24$ on synthetic measurements

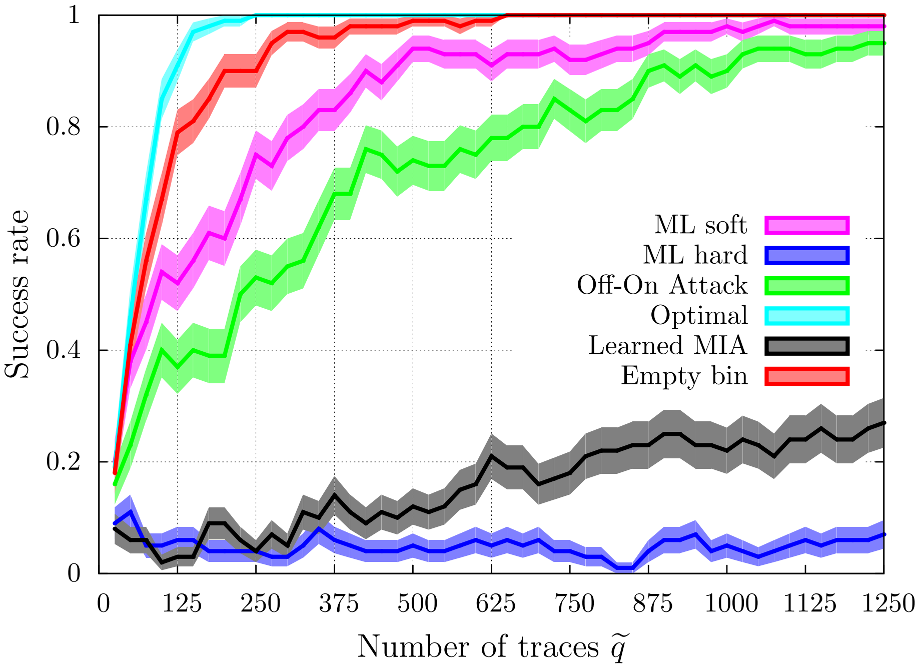


Figure 7.5: SR for $\widehat{q} = 4\ 000$ and $\sigma = 24$ on synthetic measurements

for all $(x, t) \in \mathcal{X} \times \mathcal{T}$,

$$\mathbb{P}(x|y(k)) = \begin{cases} \frac{1}{2\sigma+1} & \text{if } -\sigma \leq x - \varphi(t \oplus k) \leq \sigma, \\ 0 & \text{otherwise,} \end{cases} \tag{7.26}$$

which is constant if x is in the appropriate interval. For the Empty Bin Distinguisher,

$$\widehat{\mathbb{P}}(x|y(k)) > 0 \implies \mathbb{P}(x|y(k)) = \frac{1}{2\sigma + 1}$$

due to the leakage model. Therefore, we can predict that at least $\widehat{q} = (2\sigma+1)|\mathcal{Y}|\frac{1}{\min\ \mathbb{P}(y)} = 3\ 920$ profiling traces are needed to make sure that the Empty Bin Distinguisher becomes as efficient as the Optimal Distinguisher. As profiling consists in random draws *with replacement*, the $\mathcal{D}_{\mathsf{Empty\underline{B}in}}$ distinguisher is found very close to the $\mathcal{D}_{\mathsf{Optimal}}$ distinguisher with $\widehat{q} = 4\ 000$ profiling traces.

Chapter 8

Obtain a Leakage Model with Timing Attacks

Contents

8.1 Results on Real Devices

We have chosen to carry out a timing attack on an STM32F4 discovery board [59]. One interesting aspect is that we do not make any assumption on the model. In real life, the leakage model happens to be much more complex than the one employed in simulations (e.g., Equation (7.24)). As will be seen, in practice empty bins appear even for the correct key guess and for a "good" profiling phase. This observation differs from the ideal case of our simulations carried out in the preceding Section 7.3.

8.1.1 The ARM processor

We used a STM32F4 discovery board by STMicroelectronics[1]. It contains an STM32F407VGT6 microcontroller, which has an ARM cortex-M4 MCU with 1 MB flash memory for instructions and data, and a 192 KB Random Access Memory (RAM). The RAM is divided into three sections: one of 16 KB, another one of 112 KB, and the last one consisting of 64 KB Core Coupled Memory (CCM). The CCM has a zero flash wait state and is often used to store critical data such as data from the operating system. Since the RAM is divided into three regions, the users are unable to make use of the 192 KB RAM as a continuous memory block.

[1] We emphasize that the attacks we present are not due to a flaw in ARM or STMicroelectronics processors. Instead, as we will discuss next, the CCM feature of STM32F4 processors allows to protect the implementation against timing attacks by granting a constant execution time.

STM32F4 microcontrollers contain a proprietary prefetch module (Adaptive Real-Time memory accelerator - ART accelerator). ART accelerator contains an instruction cache which has 64 lines and a data cache which contains 8 lines. The line size of both instruction cache and data cache is 128-bits. The precise details about ART accelerator (cache replacement policy and cache associativity) are not mentioned as the module is an intellectual property of STMicroelectronics

The STM32F407VGT6 microcontroller does not have either a CPU cycle counter or a performance register to measure a cycle accurate time. However, the Data Watchpoint and Trace (DWT) unit has a cycle accurate 32 bit counter (DWTC̲YCCNT register), which can be used for measuring the duration of critical operations. When processor runs at 168 MHz, the DWTC̲YCCNT register will overflow at every 25.5 seconds providing enough time window to measure the encryption / decryption time for an adversary to measure the elapsed time without timer overflowing. In practice, we collected timing data repeatedly within the ARM, and then dump it as large data buffers sporadically. This modus operandi allowed us to reach about 10 000 measurements per second.

8.1.2 Weaknesses - Non Constant AES Time

We use OpenSSL (version 1.0.2) AES as the cryptographic library, where the $\mathrm{S}_{\mathrm{box}}$ function is implemented with large 1 KB T-boxes (see [63, Sec. 5.2.1, page 18]). Interestingly, the OpenSSL code (copied in Appendix C.1) does not contain any conditional statement, hence can be considered constant-time by a code review. However, once programmed on the STM32F4 processor, one notices that the execution duration depends on the inputs. The AES timing acquisition is illustrated in Figure 8.1. Before each encryption, we reset DWTC̲YCCNT register. This yields the exact timing of the AES execution (which is about 2 600 clock cycles in average — recall Figure 7.1 and 7.2). In a real attack, an attacker would measure a noisy timing using an external "chronometer". However, our attack models the best case for an attacker; hence, bounds the security of the analyzed implementation. In particular, we underline that our measurement methodology is fully *non invasive*: the timing measurement is performed in parallel to the AES computation, thereby keeping the victim circuit run at full speed, without interference.

Time deviations for different configurations of Instruction Cache (IC) and Data Cache (DC) are shown in Figure 8.1. We observe a huge time difference when data cache is turned Off / On. When DC is turned off, there is no timing leakage as AES is constant time. Yet, when DC is

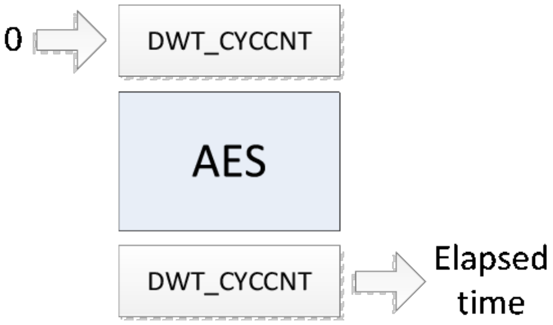


Figure 8.1: Measuring elapsed time for AES encryption

turned on, AES is not time constant. This non-constant time on AES leads to the following conclusions:

- This is a weakness for the security of the processor as two different plaintext lead to two different time clock to compute AES.
- Following Figure 8.1, it seems the enabling or not Instruction Cache, does not modify the behaviour of the leakages.
- Data presented Figure 8.1 are obtained using a fixed key and varying one byte of the plaintext.

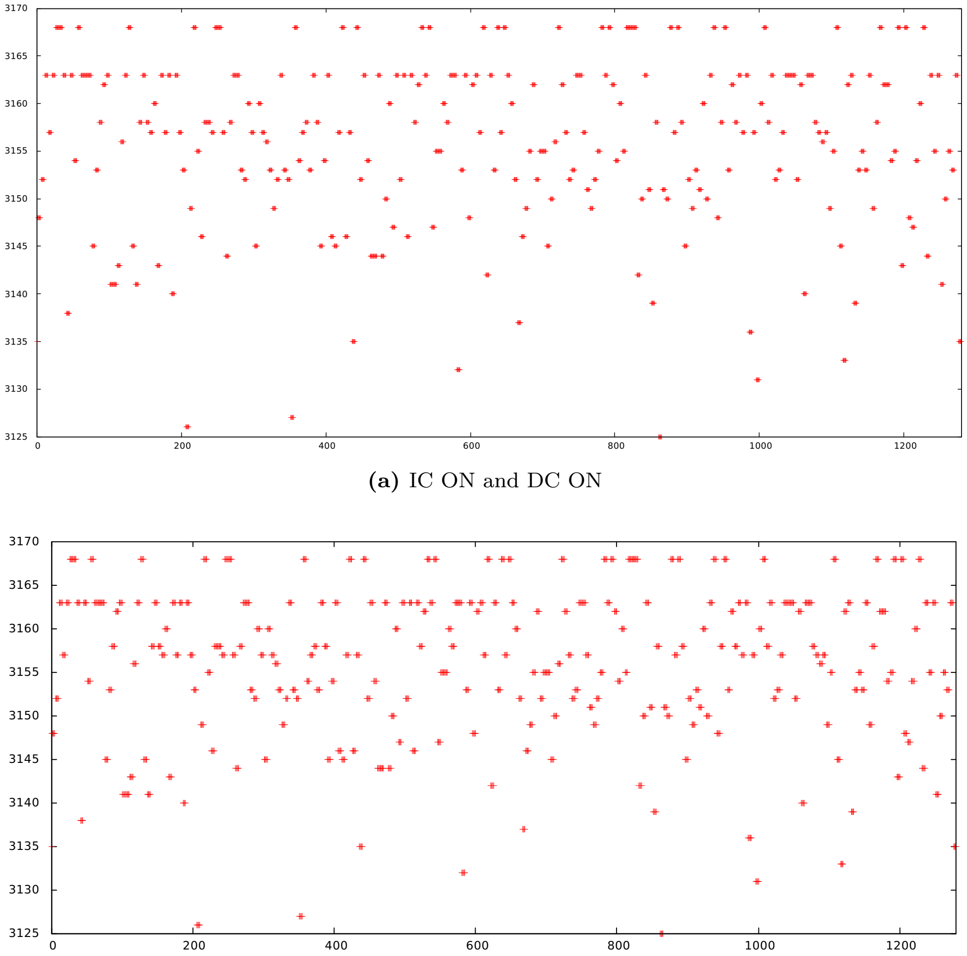
Figure 8.1 instructs us that caches shall be disabled to reduce the leakage in timing. However, we emphasize that such decision has a strongly negative impact on the AES performance: with DC off, the overall AES execution time is about 27% longer.

Therefore, in a realistic context, we shall assume that both DC and IC are enabled, which we will do in the sequel (see next Sec. 8.2 for some indications how well attacks perform when caches are disabled).

8.1.3 Characterizing the leakages for Data Cache On

As seen earlier, when the Data Cache in enabled, the AES computation is not time constant. This can be due to the T-boxes called during the computation. Indeed, calling a value in a table also stores this in the Data Cache. If this value is called within the eight next calls, the load will be faster. In Appendix C.1, we have copied the OpenSSL source code for the AES encryption with a 128 bits key. In this code, we notice that there are 160 calls to the T-boxes.

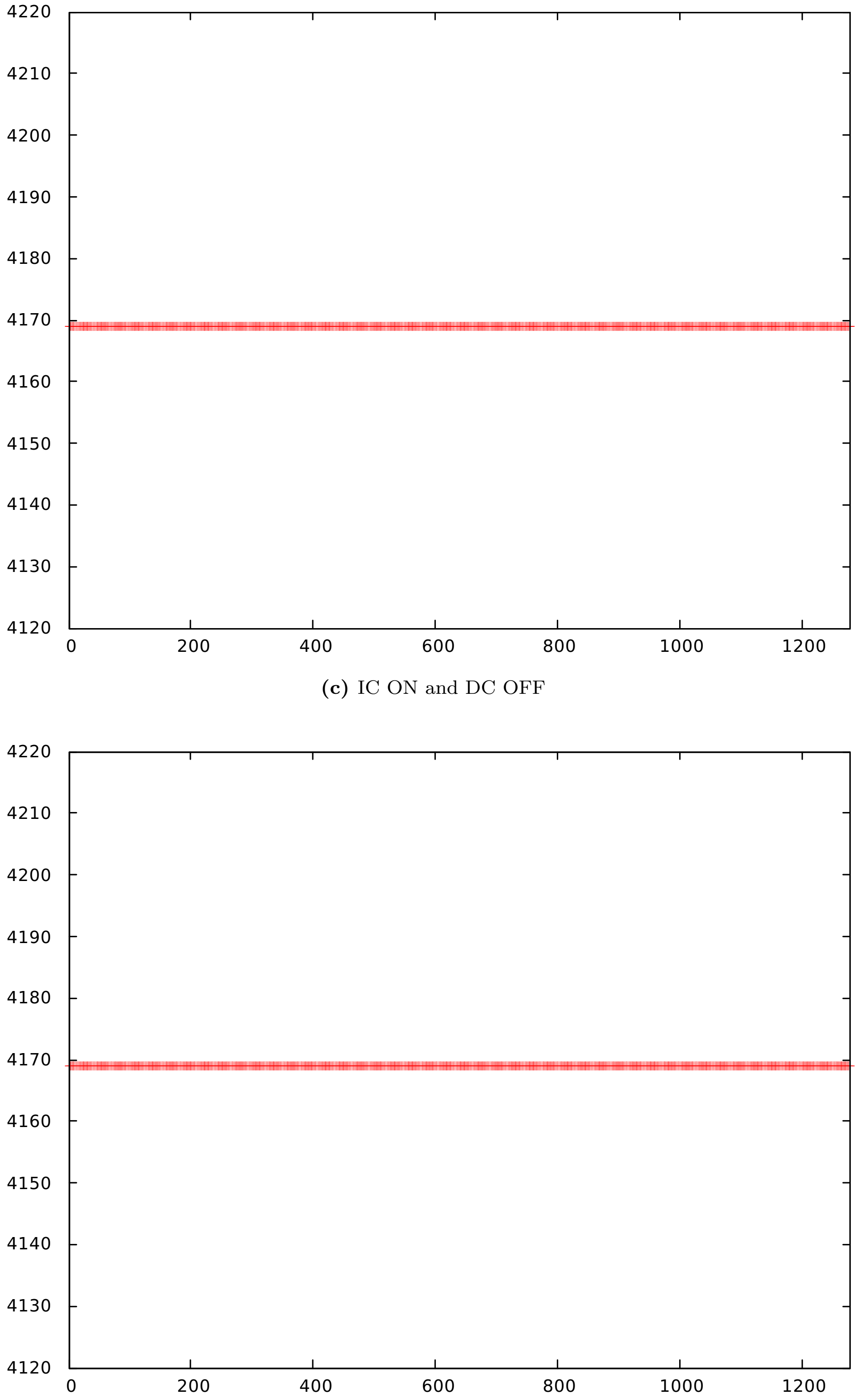
In order find a model of the leakage, we inferred the cache policy of STM32F4 ARM microcontrollers based on a thorough study of their timing response to some adaptively constructed



(a) IC ON and DC ON

(b) IC OFF and DC ON

Figure 8.2: Time deviations for different configurations of Instruction Cache (IC) and Data Cache (DC).



(c) IC ON and DC OFF

(d) IC OFF and DC OFF

Figure 8.1: Time deviations for different configurations of Instruction Cache (IC) and Data Cache (DC).

requests. We discovered that it is actually a FIFO (First-In, First Out) cache. If one requests a particular table lookup within last eight cache accesses, then the access is a hit (if not, it is a miss).

In case of a hit, the time to access such register is 5 or 6 clock cycles faster than a miss. To show this behaviour, we have done a very simple experiment:

- We generate a table of length 256;
- We generate 16 random values between 0x00 and 0xff;
- We call 16 elements of the table corresponding to the 16 values generated previously;
- We measure the time to call these 16 elements of the table.

We have plotted in Figure 8.2 the histogram of the clock cycles. the negative number in the x axis is due to the fact that we have set the 0 at the maximum value of the clock cycles, which is the obtained value for not hit at all[1]. We notice that when a hit occurs, the time is faster by 5 or 6 clock cycles. For two hits, there are three possible values: 10, 11 or 12 clock cycles.

Figure 8.2 has to be compared with a full AES encryption timing in order to see if this model is relevant. Therefore, we have plotted in Figure 8.3 the histogram for a full AES encryption. Once more, the 0 in the x axis is set to the maximum.

Very interestingly, we can observe in this figure high density levels corresponding to the hits:

1. One hit at -5 and -6;
2. Two hits at -10 and -11;
3. Three hits at -15 and -16.

Below -16 clock cycles, the hits are lost into the noise.

The comparaison of these two figures show that the FIFO model for table hits is correct, but does not explain all the time leakage due to the cache policy of the processor.

8.1.4 Attack Results

As already noticed above, the leakage model is mostly unknown. We only suppose that the text byte is mixed with the key through a XOR operation. As a consequence, the optimal

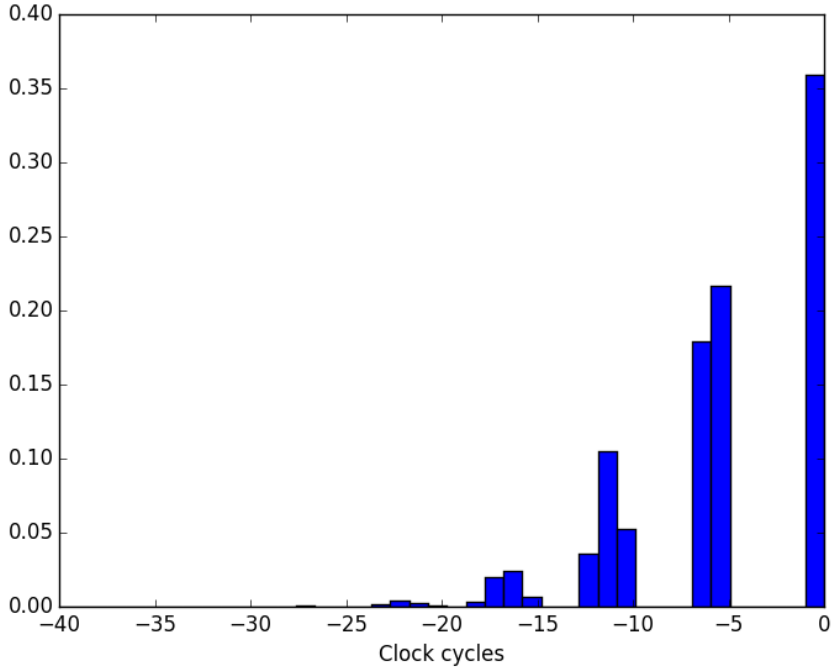


Figure 8.2: Distribution of the clock cycles for a simple example

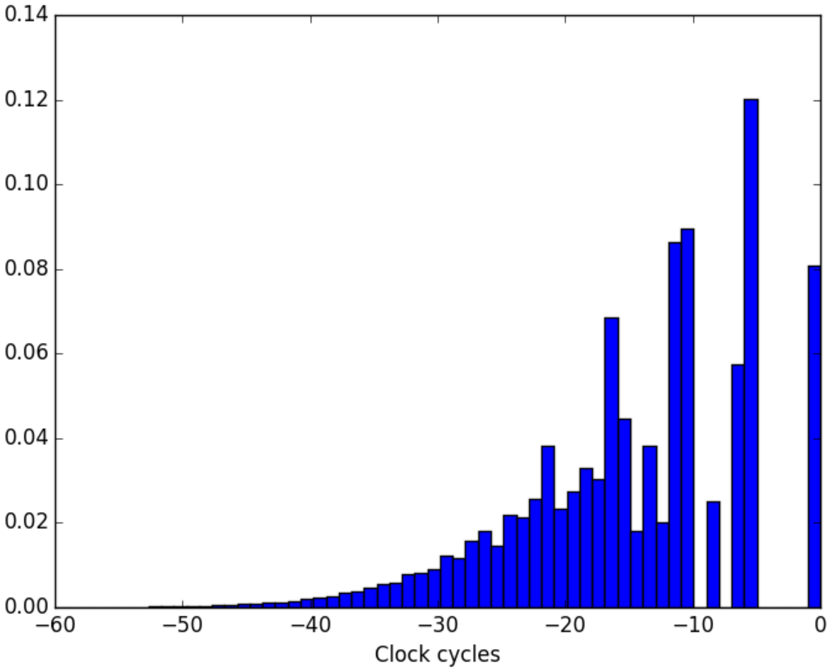


Figure 8.3: Distribution of the clock cycles for a full AES encryption

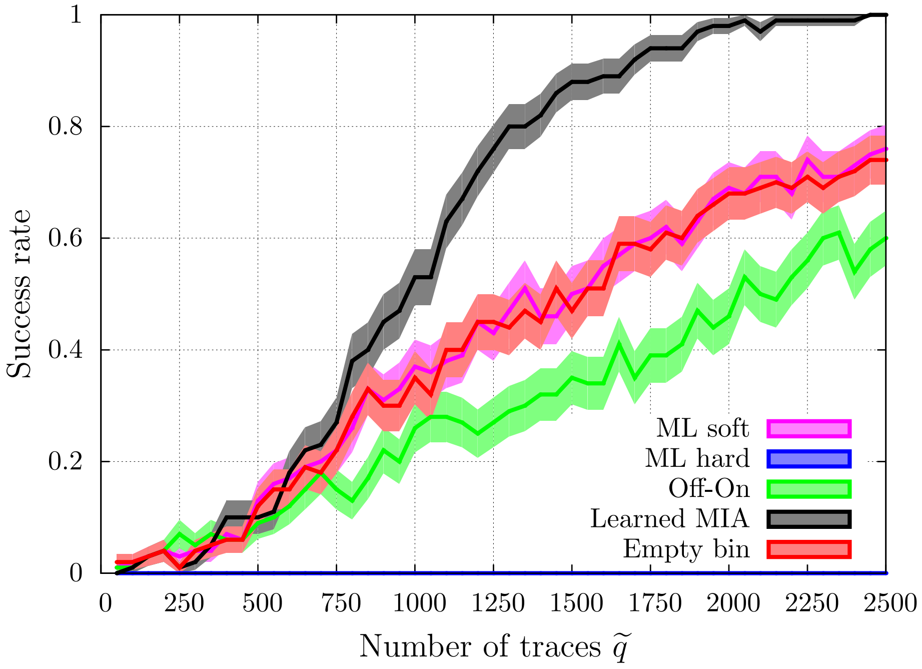


Figure 8.4: SR for $\widehat{q} = 25\ 600$ on real-world measurements

distinguisher (giving the limit of performance) is not known. The SNR of the leakage is $\mathsf{Var}(\mathbb{E}(x|t))/\mathbb{E}(\mathsf{Var}(x|t)) = 0.4$.

In Figure 8.4, we notice that Learned MIA is the best distinguisher in the case of poor profiling. The Hard Drop Distinguisher is not succeeding at all since it drops about 90% of the data.

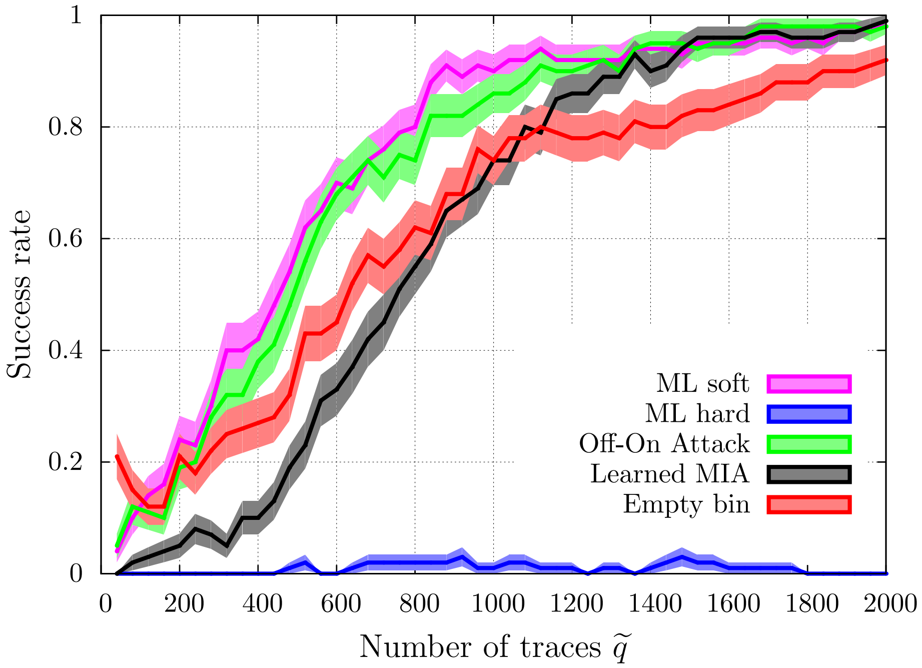


Figure 8.5: SR for $\widehat{q} = 256\ 000$ on real-world measurements

Figure 8.5 presents the success rate for a better profiling stage. We notice the following interesting improvements:

- The Learned MIA distinguisher is only slightly better than in Figure 8.4. To reach 80% success rate, 1 100 traces are needed as compared to 1 250 traces previously.
- The Soft Drop and Offline-Online distinguishers are the best distinguishers in this scenario, with a small advantage for the Soft Drop distinguisher.
- The Hard Drop distinguisher remains unsuccessful.

[1] This is a voluntary choice as we only focus on the gap between two picks of distribution. The absolute value has no real sense since we are comparing two computations that are not the same.

We notice that the Soft Drop Distinguisher has been established using the γ parameter defined in Equation 7.19 such that $\gamma = 1/\widetilde{q}$.

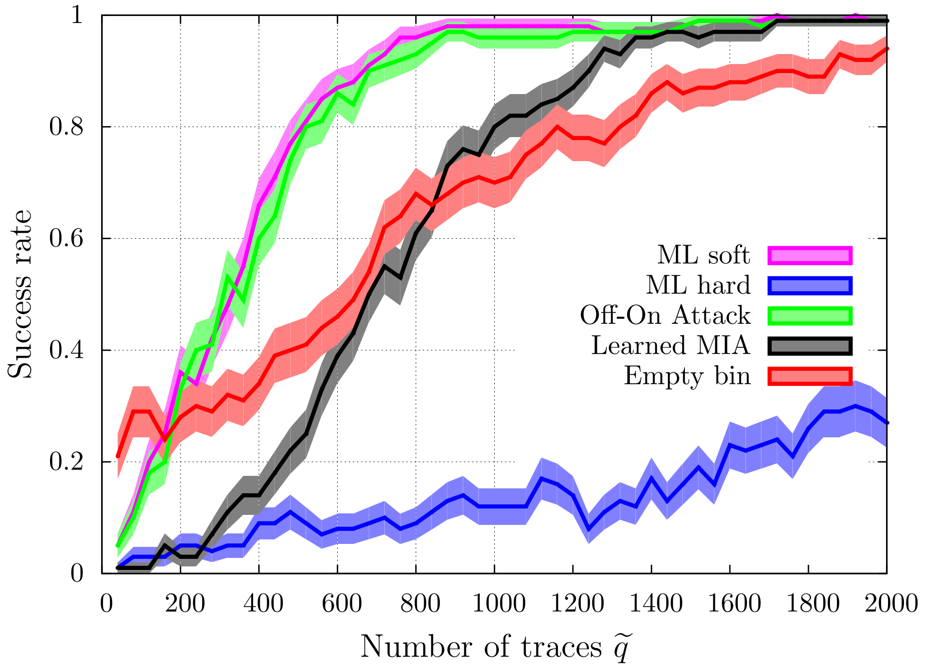


Figure 8.6: SR for $\widehat{q} = 2\ 560\ 000$ on real-world measurements

Figure 8.6 is the continuation of Figure 8.5 with much more traces in the profiling stage. The resulting profiling is very good and one may consider that the approximation of $\mathbb{P}$ is tight. In this case, Soft Drop and OOP Distinguishers are both very successful, which seems natural regarding the fact that $\widehat{\mathbb{P}}$ has converged to the actual probability $\mathbb{P}$. For this attack, we recall that the timing of 10 000 traces can be acquired in one second. Therefore, the attack is successfully in about 0.2 second using Soft Drop or OOP distinguishers.

As a conclusion to this study on the STM32F4 discovery board, we have learned the following comparisons between the proposed distinguishers:

- when the profiling stage is poor, the best distinguisher is the Learn MIA Distinguisher;
- when there is enough data in the profiling stage, the best distinguisher is the Soft Drop Distinguisher, closely followed by the OOP Distinguisher;

- the Empty Bin Distinguisher converges to the optimal success rate, but is not as efficient as previously in Section 7.3. This can be explained by the fact that we skip a lot of data in the computation;
- the Hard Drop Distinguisher is the slowest to converge to 100% success rate.

Remark 8.1. When comparing Figures 8.5 and 8.6, we notice that the Empty Bin distinguisher does not improve as the number of profiling traces increases. An explanation that there is no more empty bins to be filled between these two situations; then only a more precise estimation of the probability would make the difference.

Remark 8.2. As discussed in Definition 7.4, the value of γ is important. We have run the same experience as in Figure 8.5 with $\gamma = \frac{1}{\widehat{q} \times 10^{10}}$. The results, we obtained, are presented in Figure 8.7. When comparing this figure with Figure 8.5, we notice that the performance of the Soft Drop Distinguisher has dropped and is now much closer to that of the Empty Bin Distinguisher, as we had forecast.

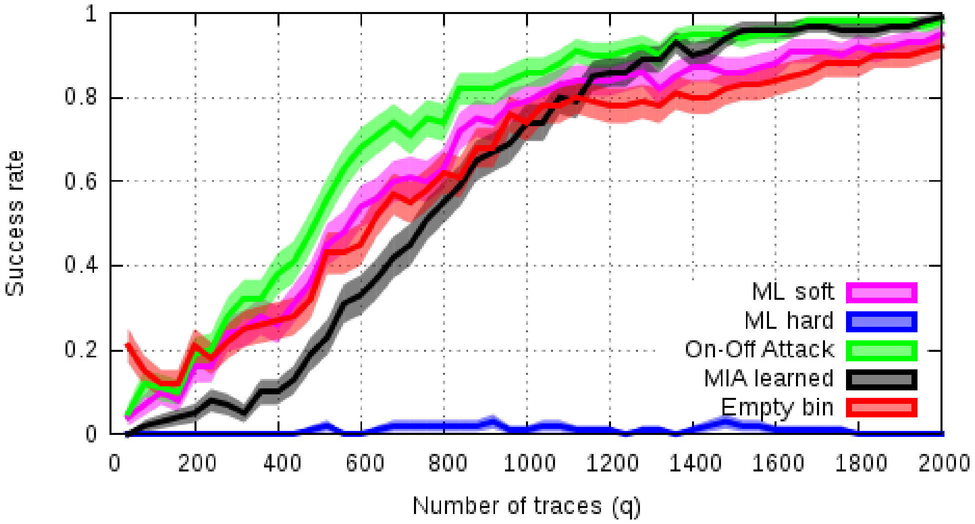


Figure 8.7: SR for $\widehat{q} = 256\ 000$ with $\gamma = \frac{1}{\widehat{q} \times 10^{10}}$.

8.1.5 Nature of Empty Bins

Defined in 7.1.2, Empty Bins can appear under two circumstances. The first possibility is insufficient profiling: some rare occurrences are not encountered by lack of training measurements. The second possibility is what we call *Structural Empty Bins*. They are present whatever the profiling under fixed key and do not depend on the number of traces $\widehat{q}$ in the profiling stage. In order to decide for the reason of Empty Bins, we have drawn the number of empty bins for a given key according to the number of traces in the profiling stage.

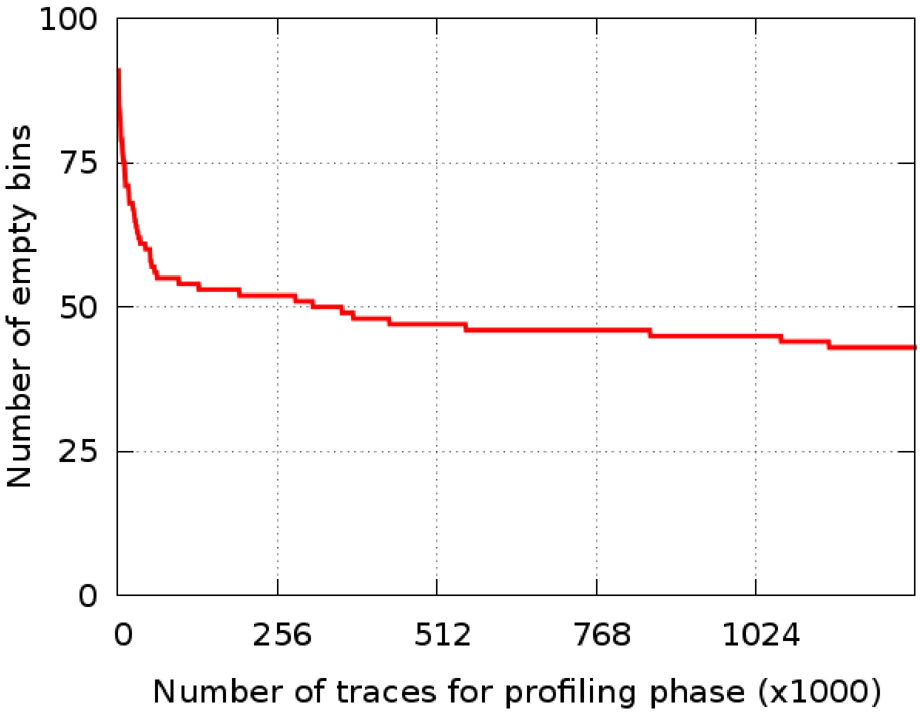


Figure 8.8: Empirical number of empty bins

Figure 8.8 presents this study obtained with the STMicroelectronics Discovery Board. We considered $\widehat{q} = 1\,280\,000$, and define the number of empty bins as:

$$\left|\left\{x \in \{\min_{q=1}^{\widehat{q}} \widehat{x}_q, \ldots, \max_{q=1}^{\widehat{q}} \widehat{x}_q\}, \text{ such that } \nexists q,\ \widehat{x}_q = x\right\}\right|.$$

We can see that the number of empty bins decreases, but never reaches 0. At the beginning, the high number of empty bins is due to both poor profiling and structural empty bins. With a good profiling, we only keep the structural empty bins.

8.1.6 Study on the Mean-Square Error

An interesting point noticed in Figures 8.4, 8.5, and 8.6 is that the Learned MIA distinguisher is working better than the Soft Drop Distinguisher for a poor learning phase (i.e., $\widehat{q} = 25\ 600$). However, with a better learning phase (i.e., $\widehat{q} = 256\ 000$ and $\widehat{q} = 2\ 560\ 000$), the Soft Drop Distinguisher has a much better success rate. In order to understand why the Learned MIA

Distinguisher does not improve that much with a better learning phase, we have computed the Mean-Square Error of these two distinguishers for the three learning phases (i.e., $\widehat{q} \in \{25\,600, 256\,000, 2\,560\,000\}$).

Definition 8.1 (MSE, Bias and Variance)**.** Let us consider a random variable X and its expectation $\theta = \mathbb{E}[X]$. An estimator of the random variable is noted $\bar{X}$. The MSE is defined as follows:

$$\mathsf{MSE} = \mathbb{E}\left[(\bar{X} - \theta)^2\right].$$

The bias of the estimator is the expectation of the difference between the estimator and the mean of the random variable:

$$\mathsf{Bias} = \mathbb{E}\left[\bar{X} - \theta\right].$$

At last, the variance of the estimator is:

$$\mathsf{Variance} = \mathbb{E}\left[\bar{X}^2\right] - \mathbb{E}\left[\bar{X}\right]^2$$

From these definitions, we have the following relation between MSE, bias and variance:

$$\mathsf{MSE} = \mathsf{Bias}^2 + \mathsf{Variance} \tag{8.1}$$

The Mean-Square Error (MSE) is computed using the following method:

1. For the secret key k^*, we calculate the value of the distinguisher i.e. the value of $\widehat{\mathbb{P}}(\widetilde{\mathbf{x}}|\widetilde{\mathbf{t}} \oplus k^*)$ for the Soft Drop and $I(\widetilde{\mathbf{x}}; \widehat{\varphi}(\widetilde{\mathbf{t}} \oplus k^*)$ for the Learned MIA. We compute this value for different number of traces $\widetilde{q}$. This gives an estimation of the normalized distinguisher for the correct key.

2. The most accurate estimation is obtained for the highest value of $\widetilde{q}$. Therefore, taking the average over a large number of experiences for this highest value of $\widetilde{q}$ gives a good estimation of the Expectation of the estimator.

3. Then we calculate, for every value of $\widetilde{q}$ the bias and the variance of the estimator, and the Average MSE is obtained using the formula: $\mathsf{MSE} = \mathsf{Bias}^2 + \mathsf{Variance}$.

We have plotted in Figures 8.9 and 8.10 the Average MSE for the two distinguishers. In order to be more relevant, we have plotted the logarithm of the MSE. Furthermore, we have chosen to plot the MSE separately as the distinguishers are not comparable.

The MSE for the Learned MIA Distinguisher stays almost constant with the improvement of the learning phase whereas the MSE of the Soft Drop Distinguisher is much smaller. This means that a better learning phase gives a much better estimator of the distinguisher.

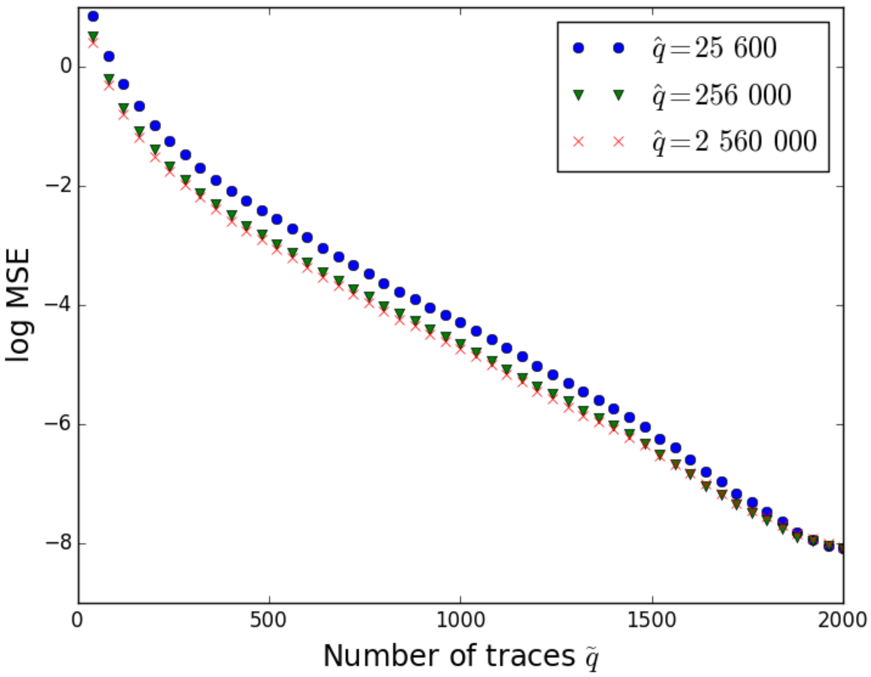


Figure 8.9: Average MSE for the Learned MIA Distinguisher

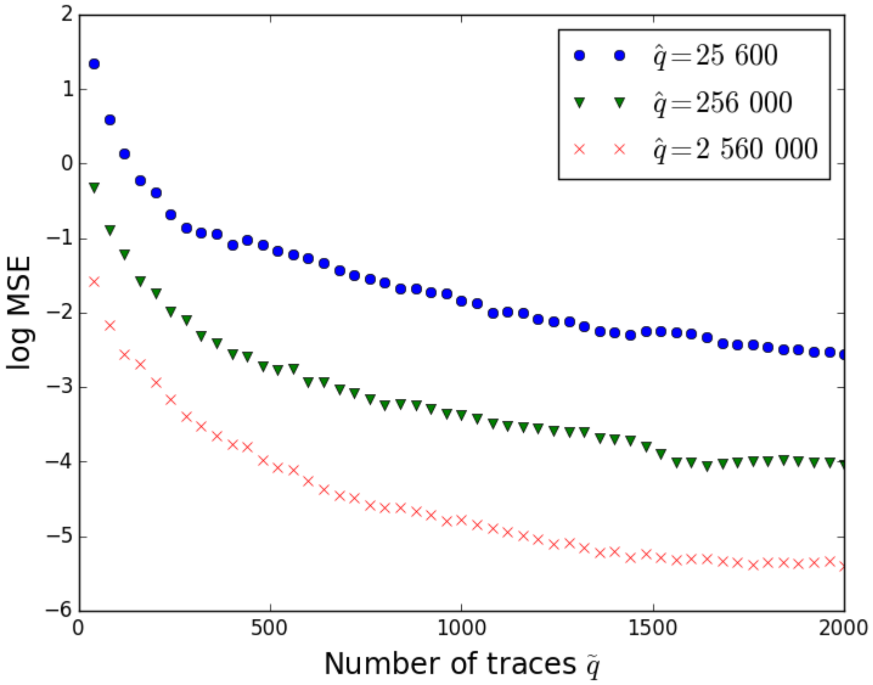


Figure 8.10: Average MSE for the Soft Drop Distinguisher

To understand more deeply this MSE, we separate bias and variance for these two distinguishers. The results are computed Figure 8.11 for the Learned MIA Distinguisher and Figure 8.12 for the Soft Drop Distinguisher.

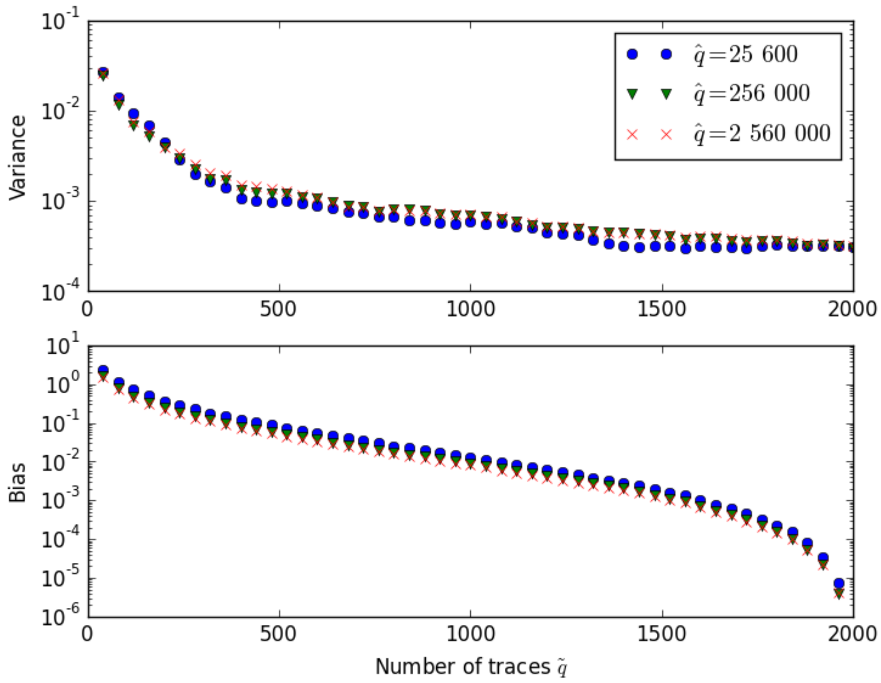


Figure 8.11: Variance and bias of the Learned MIA Distinguisher

We notice the following aspects:

- For the Soft Drop Distinguisher, the bias is almost equal to zero. In fact, the MSE is the variance.
- For the Learned MIA Distinguisher, it is mainly the opposite: the biggest part of the MSE is the bias.

To conclude with the MSE, the Soft Drop Distinguisher improves because the estimator has a much smaller variance with a better learning phase. Meanwhile, the Learned MIA Distinguisher does not improve because it is a biased estimator and a better learning phase does not reduce this bias.

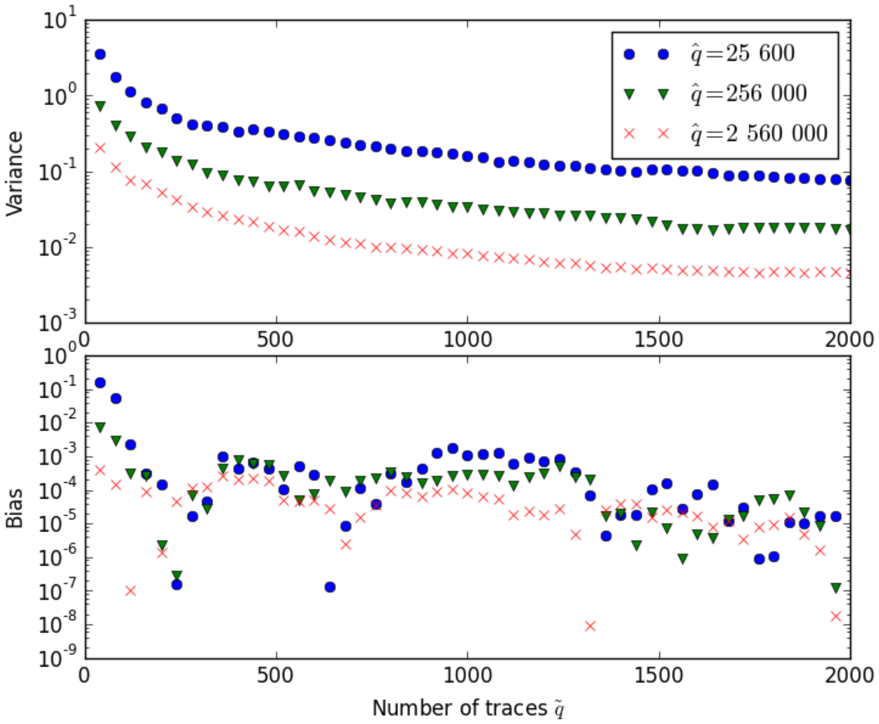


Figure 8.12: Variance and bias of the Soft Drop Distinguisher

8.2 Success Rate in Presence of External Noise

The measurement setup used in simulation (Sec. 7.3) and on real-world traces (Sec. 8.1) is ideal. Indeed, the only considered noise is said *algorithmic*, i.e., it consists in the varying timing which arise from the parts of the algorithm not under study. In this section, we analyse the effect of noise external to the monitored cryptographic algorithm. Subsection 8.2.1 discusses in general terms the effect of noise addition, and subsection 8.2.2 details quantitatively how distribution-based distinguishers cope efficiently with noise (while moment-based distinguishers fail to resist noise).

8.2.1 Effect of Measurement Noise

However, in practice, timing measurements contain a noisy part. Let us give three examples:

1. Measure of a difference of timing between request and response from the AES (over a network of unknown latency);

2. Use of a side-channel signal (such as the power or the electromagnetic field) to observe the AES computation; the beginning and the end of an AES are easy to identify, as they consist in sixteen consecutive operations (namely sixteen XOR making up the AddRoundKey operations). As these patterns have a remarkable signature, they can be extracted with great accuracy thanks to a mere cross-correlation. Still, the AES itself might not be executed in constant time, hence some alignments issues;

3. Use of a cache attack, which would disclose that the program flows entered and exited the AES function. However, the timing for access to cache is non deterministic.

Let us denote the variance of the added noise as σ^2.

Now, it is known that any additive distinguishers (which is the case of our distinguishers), the number of traces to recover the secret for a given success rate is inversely proportional to the inverse of the signal-to-noise ratio (see e.g., [36, Corollary 2]).

As a direct consequence, we can predict the complexity of the attacks when IC and DC are disabled. It can be seen in Figure 8.1 that the timing variation is about divided by three (from ≈ 20 to ≈ 8) when the DC is disabled. Therefore, the number of required traces to recover the key is about multiplied by three.

In addition, we can approximate the required number of traces to extract the key in presence of external noise of standard deviation σ. In our case-study of OpenSSL AES on ARM, the algorithmic noise has standard deviation about 20 clock cycles (see Figure 7.1 and 7.2).

So, if the external noise has standard deviation $\sigma < 20$, the impact is small. But when $\sigma/20 > 1$, the influence of the external noise becomes preponderant. As the algorithmic noise and the external noise are independent, the number of traces required to extract the key will actually grow linearly with σ as soon as $\sigma/20 \gg 1$.

8.2.2 Comparison with Existing Methods in the Presence of Noise

In this subsection, we aim at comparing our distribution-based method with the existing methods (moment-based method mentioned in Tab. 6.1). In particular, we focus on the representative Bernstein correlation [5] with a learned model [the timing expectation for each value of the target AES byte], that we refer to as "CPA". This "CPA" between timing measurements and the learned average of timing per byte of the key does not suffer from the empty bin issue. We start by a comparison with little external noise. In this case, we have plotted in Figure 8.13 the success rate for both the soft drop distinguisher and the CPA. The x axis represents the number of traces for the profiling phase while the y axis is the number of traces needed during the attack to reach 80% of success rate. We notice that the CPA performs better than the soft drop method, for any profiling (even when learning with several million of traces). This can be due to bias between the profiled distribution and the attack distribution.

However, in a practical case, we encounter noisy timing leakages. In order to compare our methods with the existing methods (such as CPA) in the presence of external noise, we plotted Figure 8.14. In this figure, we took a good profiling phase ($\widehat{q} = 3 \times 10^6$), i.e., profiling is performed on sufficiently enough traces. This figure is obtained for a noisy timing, that is the nominal time to compute AES (as in Subsec. 7.1.2), where the noise follows the following law:

$$\begin{cases} 0 & \text{added time with probability 50\%,} \\ \text{T} & \text{added } (T \in \mathbb{N}\text{, a number of clock periods), with probability 50\%.} \end{cases} \tag{8.2}$$

This models the interruption of the CPU from a peripheral when AES is baremetal, or a descheduling of the AES process during one *time slot* on systems with an operating system (OS). Indeed, such events have the consequence, when they occur, to add a long period of time (often as long or even longer than the duration of the AES) to the encryption time, so that the interruption can be served, or so that the OS re-schedules the AES process. We notice that, in such case, it is more interesting to compute one of our methods, rather than previous existing

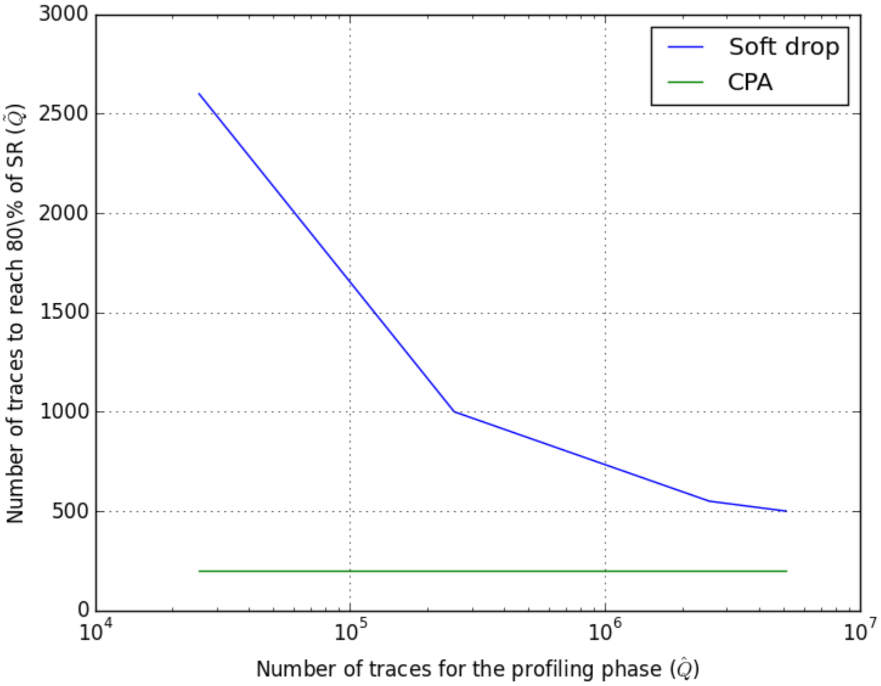
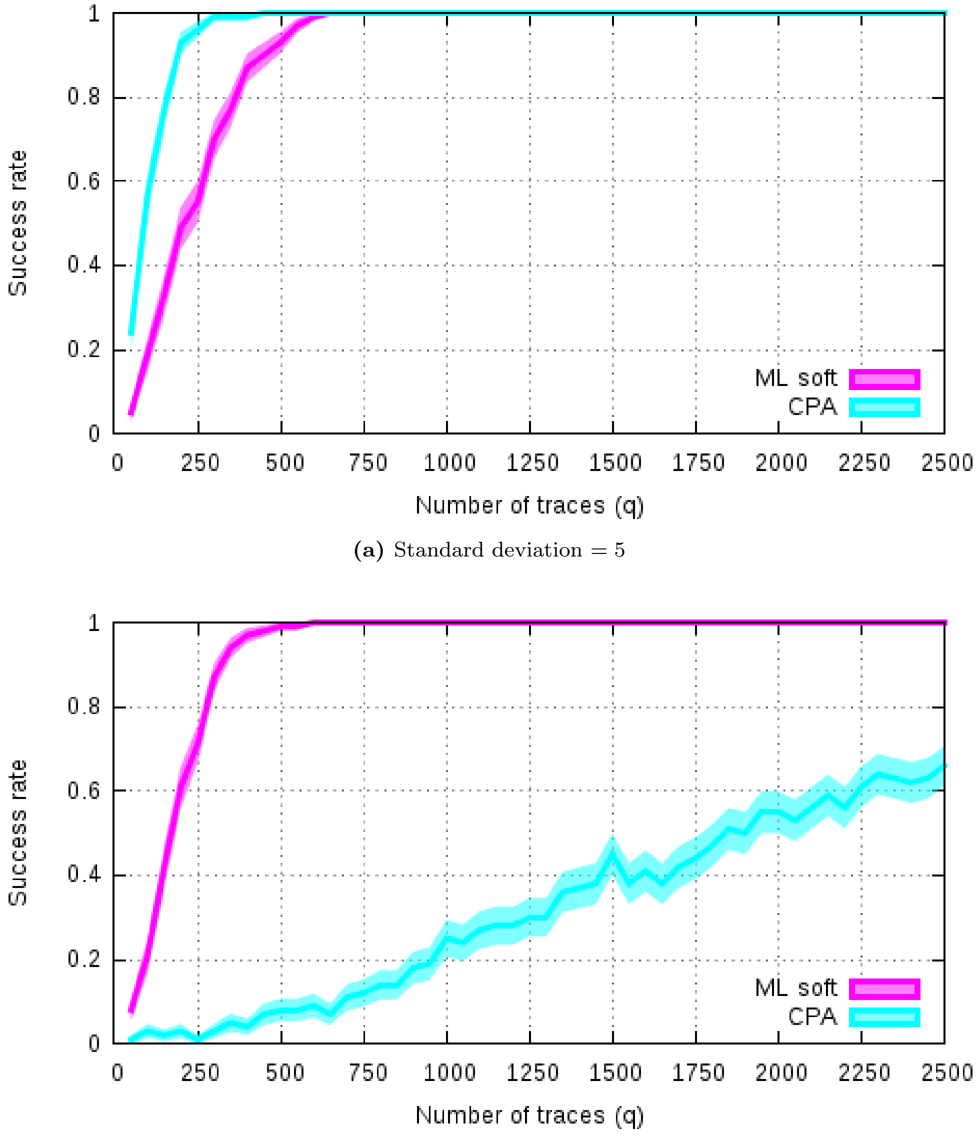


Figure 8.13: Comparison between CPA and soft drop distinguisher at 80% of success rate



(a) Standard deviation = 5

(b) Standard deviation = 50

Figure 8.14: Success rate for soft drop versus CPA for small noise and noise of standard deviation $T = 50$ (recall (8.2))

methods such as CPA. Indeed, distribution-based profiling is more accurate than CPA estimation with noisy signals. For instance, the results from Hassan Aly and Mohammed ElGayyar [1] show that 2^{22} encryptions are required for a key extraction on a more recent processors (Pentium Dual-Core and Core 2 Duo), which is significantly more than that used by Bernstein CPA in his original attack [6]. The authors of this paper remark incidentally that the best method is not to use correlation with the *means* of each class, but with the *minimum* value in each class. This confirms that the complexity of the distributions are better suited for distinguishing that simply the average per class. This justifies that our study focuses on distribution-based distinguishers (more robust to binary noise situations encountered while measuring durations) rather than moment-based distinguishers (recall Tab. 6.1).

8.3 Conclusion and Perspectives

We have derived several "information-theoretic" distinguishers as possible solutions to the empty bin issue. Some of them, like the Dirichlet Prior and the Offline-Online distinguishers, required the computation of novel distributions. We have shown in particular that the empty bins, previously believed to be an annoyance and dropped accordingly, can turn out to be valuable assets for the attacker as long as they are treated carefully. In all the part, real timing data are used, making the results very practical.

We have also compared the various distinguishers under two frameworks: a simulated test with synthetic leakage and real-world timing attacks. In both cases, we noticed that the outcome of the attacks depends on the quality of the profiling stage. A good profiling improves the results, where the best distinguisher seems to be the Soft Drop Distinguisher. A poor profiling makes the traditional distinguishers break down. More sophisticated solutions like Offline-Online Profiling and Learned MIA distinguishers are very useful in this case. A possible way to investigate more on this aspect is to use more powerfull statistical tools in order to extract the most precise model for the Learned MIA Distinguisher.

The interesting aspect on the studied timing attack is that one does not have to make any assumption on the leakage model. In addition to this, the main advantage of the new distinguishers is that the empty bin issue is completely solved. We also introduced distinguishers which can jointly exploit offline and online side-channel measurements. As an interesting perspective, our approach could advantageously be analyzed using the "perceived information" metric recently introduced by Standaert et al. in [71, Eqn. (1)].

Another perspective would be to compare our information-theoretic attacks with attacks based on machine learning techniques. Surprisingly and contrary to results reported in other papers, our preliminary results show that SCA based on support vector machines [40] has poor performance, even when profiling with very few traces ($\widehat{q}$ is small), which may be due to the univariate nature of the leakage.

An interesting observation is that writing cryptographic code robust to timing attacks is challenging. While the OpenSSL code for AES has no obvious flaw (such as unbalanced branches which depend on sensitive data), the timing of AES is data-dependent, due to microarchitectural features of the studied ARM core. There seem to exist two classes of solutions against timing attacks: The first aims at randomizing the execution timing, as studied for instance in [6]. Such an implementation can still be attacked with high-order distinguishers, albeit with more traces than without any protection. The second would attempt to balance the timing, yet this requires some hardware support such as the CCM feature of the STM32F4 processors.

Part V

Conclusion and Perpectives

Chapter 9

Conclusion

9.1 Conclusion

The title of my Ph.D. is "Towards a Better Formalization of the Side-Channel Threat". When I was recruited by Olivier Rioul in 2015 to start my Ph.D., Annelie Heuser was finishing writing her manuscript. During her thesis, she specially focused on the study of distinguishers. I could therefore use her fresh results to work on my thesis.

During my thesis, I focused on the similarities between a side-channel leakage model and a communication channel. The main result of these three years is based on a mathematical link between the success rate of an attack and the signal-to-noise ratio of the leakage. To obtain this result, I have used information theoretic tools. More specifically, I focused on the Mutual Information between the sensitive variable and the measured traces. When the noise is Gaussian, the Mutual Information is a function of the SNR. However, the main difficulty is that the leakage is not independent. This means that the Mutual Information is more complicated than Shannon's formula. I have therefore investigated on the estimation of Mutual Information for this particularity of side-channel analysis. This is indeed a particularity since in the literature in communication theory, most of the channels are considered with independent random variables.

According to me, this result will give a good bound to the efficiency of a model. Indeed, in side-channel analysis, the SNR is model dependant. With this knowledge of the SNR, we have an upper-bound on the success rate. This gives a first idea of the level of security of a chip without making simulations to calculate the success rate.

It is possible to extend this result to protected implementations but the formula of the mutual information is much more difficult to calculate. A possibility would therefore be to consider the approximation of Independence even if the bound is therefore looser.

On a more personal point-of-view, this thesis has been for me a very intense period. When I arrived at Télécom in 2015, I had never heard of cryptography. Side-channel analysis has therefore been for me a very good mean to enter this particular world. Side-channel analysis is a mixture between cryptography, electronics, mathematics and physics. This is therefore a very beautiful topic of research and I am confident that several talented researchers will invest time in this field.

More generally, this three years and a half Ph.D. has made of me a more mature man. When I started in 2015, I had no idea of what I could do after my thesis. During these three years I have learned how to be rigorous and how to self-criticize my work. I believe that I will be very well prepared for my future work.

9.2 Further Perspectives

Side Channel analysis has a very large spectrum of applications and studies. This topic is quite new since the first literature on this subject appeared in the 1990s.

The next challenges for the coming years will be focused on the Artificial Intelligence. Indeed, neural networks will be soon able to extract leakage models and furthermore, be able to recover secret keys. On the future of cryptography, post-quantum is a big topic of research. To prepare this quantum revolution, designers are already rethinking asymmetric algorithms (such as RSA) to be secured against quantum cryptanalysis. These new algorithms will be of course subject to side-channel analysis and it will be interesting to study them on different architectures.

On the study of distinguishers, another interesting topic of research will be the success rate under a wrong leakage model. In communication theory, this is called *mismatch decoding.* I mentioned this issue in the appendix but I believe that there is a lot to study in order to predict the success rate of an attack under this supposition.

Part VI

Appendix.

Appendix A

Appendix on the Shannon Bound

A.1 Proof of Lemma 4.1

Let $\mathbf{t} \in \mathfrak{T}$ and τ be the considered permutation. We have

$$\begin{aligned}
H(\mathbf{X} \mid \mathbf{T} = \mathbf{t}) &= -\sum_{\mathbf{x}} \mathbb{P}(\mathbf{x} \mid \mathbf{t}) \log_2 \mathbb{P}(\mathbf{x} \mid \mathbf{t})) \\
&= -\sum_{\mathbf{x}} \left[\sum_k \mathbb{P}(k) \mathbb{P}(\mathbf{x} \mid \mathbf{t}, k) \right] \log_2 \left(\sum_k \mathbb{P}(k) \mathbb{P}(\mathbf{x} \mid \mathbf{t}, k) \right) \\
&= -\sum_{\mathbf{x}} \left[\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_i \mid t_i, k) \right] \log_2 \left(\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_i \mid t_i, k) \right)
\end{aligned}$$

Re-arranging both products so that they are ordered in accordance with the permutation, we obtain

$$\begin{aligned}
H(\mathbf{X} \mid \mathbf{T} = \mathbf{t}) &= -\sum_{\mathbf{x}} \left[\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_{\tau(i)} \mid t_{\tau(i)}, k) \right] \log_2 \left(\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_{\tau(i)} \mid t_{\tau(i)}, k) \right) \\
&= -\sum_{\mathbf{x}} \left[\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_i \mid t_{\tau(i)}, k) \right] \log_2 \left(\sum_k \mathbb{P}(k) \prod_{i=1}^{q} \mathbb{P}(x_i \mid t_{\tau(i)}, k) \right) \\
&= H(\mathbf{X} \mid \mathbf{T} = \tau(\mathbf{t})) \qquad \square
\end{aligned}$$

A.2 Proof of (3.5)

We study the sign of the difference

$$\begin{aligned}
\Delta &= -\mathbb{E}_Y \log_2 \mathbb{E}_X[\exp(f(X,Y))] + \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y f(X,Y))]; \\
&= -\log_2 \exp \mathbb{E}_Y \log_2 \mathbb{E}_{X'}[\exp(f(X',Y))] + \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y \log_2 \exp f(X,Y))]; \\
&= \log_2 \mathbb{E}_X \frac{\exp(\mathbb{E}_Y \log_2 \exp f(X,Y))}{\exp \mathbb{E}_Y \log_2 \mathbb{E}_{X'}[\exp(f(X',Y))]}; \\
&= \log_2 \mathbb{E}_X \exp \mathbb{E}_Y \left[\log_2 \exp f(X,Y) - \log_2 \mathbb{E}_{X'}[\exp(f(X',Y))] \right]; \\
&= \log_2 \mathbb{E}_X \exp \mathbb{E}_Y \left[\log_2 \frac{\exp f(X,Y)}{\mathbb{E}_{X'}[\exp(f(X',Y))]} \right].
\end{aligned}$$

Since the log function is concave:

$$\begin{aligned}
\Delta &\leq \log_2 \mathbb{E}_X \exp \log_2 \mathbb{E}_Y \left[\frac{\exp f(X,Y)}{\mathbb{E}_{X'}[\exp(f(X',Y))]} \right]; \\
&= \log_2 \mathbb{E}_X \mathbb{E}_Y \left[\frac{\exp f(X,Y)}{\mathbb{E}_{X'}[\exp(f(X',Y))]} \right]; \\
&= \log_2 \mathbb{E}_Y \left[\frac{\mathbb{E}_X \exp f(X,Y)}{\mathbb{E}_{X'} \exp(f(X',Y))} \right]; \\
&= \log_2 \mathbb{E}_Y \left[1 \right]; \\
&= 0. \qquad \square
\end{aligned}$$

A.3 Proof of Corollary 3.1

In Lemma 3.5, we have proven that:

$$\mathbb{E}_Y \log_2 \mathbb{E}_X[\exp(f(X,Y))] \geq \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y f(X,Y))]$$

or

$$\mathbb{E}_Y \log_2 \mathbb{E}_X[\exp(f(X,Y))] \geq \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y \log \exp f(X,Y))].$$

Setting $g(x,y) = \exp(f(x,y))$, we have:

$$\mathbb{E}_Y \log_2 \mathbb{E}_X[g(X,Y)] \geq \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y \log g(X,Y))].$$

Hence,

$$\begin{aligned}\exp \mathbb{E}_Y \log_2 \mathbb{E}_X[g(X,Y)] &\geq \exp \log_2 \mathbb{E}_X[\exp(\mathbb{E}_Y \log g(X,Y))] \\ &\geq \mathbb{E}_X[\exp(\mathbb{E}_Y \log g(X,Y))] \qquad \square\end{aligned}$$

A.4 Alternative Proof of (3.5) and Further Comments

Consider, for any random vector $\mathbf{Y}'$,

$$\begin{aligned}\Delta &= I(\mathbf{X};\mathbf{Y}) + \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp\left(\mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}\right) \\ &= \mathbb{E}_{\mathbf{Y}}\mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}{\mathbb{P}(\mathbf{X})} + \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp\left(\mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}\right) \\ &= \mathbb{E}_{\mathbf{Y}} \log \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}{\mathbb{P}(\mathbf{X})} + \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp\left(\mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}\right) \\ &= \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}{\mathbb{P}(\mathbf{X})} + \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp\left(\mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}\right) \\ &= \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})\mathbb{P}(\mathbf{X} \mid \mathbf{Y})} \\ &= \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})}\end{aligned}$$

By the concavity of the log function,

$$\begin{aligned}\Delta &\leq \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp \log \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})} \\ &= \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \frac{\mathbb{E}_{\mathbf{Y}'}\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})} \\ &= \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \frac{\mathbb{P}(\mathbf{X}')}{\mathbb{P}(\mathbf{X})}\end{aligned}$$

where the $\mathbf{X}'$ distribution is given by $\mathbb{P}(\mathbf{x}') = \mathbb{E}_{\mathbf{Y}'}\mathbb{P}(\mathbf{x} \mid \mathbf{Y}')$. It is important to note that this derivation can be applied for any random vector $\mathbf{Y}'$. The derivations made in Section 3.2 were made for $\mathbf{Y}'$ following the same distribution as $\mathbf{Y}$. In this case $\mathbb{P}(\mathbf{X}') = \mathbb{P}(\mathbf{X})$ and

$$\Delta \leq \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \frac{\mathbb{P}(\mathbf{X})}{\mathbb{P}(\mathbf{X})} = 0$$

which proves inequality (3.5). □

Another choice is to take an i.i.d. vector $\mathbf{Y}'$ having the same marginals as $\mathbf{Y}$. Then

$$\Delta = \mathbb{E}_{\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})}$$

and by Corollary 3.1,

$$\begin{aligned}
\Delta &\leq \mathbb{E}_{\mathbf{Y}} \log \exp \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \mathbb{E}_{\mathbf{Y}'} \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})} \\
&= \mathbb{E}_{\mathbf{Y}} \mathbb{E}_{\mathbf{X}|\mathbf{Y}} \log \frac{\mathbb{E}_{\mathbf{Y}'}\mathbb{P}(\mathbf{X} \mid \mathbf{Y}')}{\mathbb{P}(\mathbf{X})} \\
&= \mathbb{E}_{\mathbf{X},\mathbf{Y}} \log \frac{\prod_i \mathbb{P}(X_i)}{\mathbb{P}(\mathbf{X})} \times \frac{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})}{\mathbb{P}(\mathbf{X} \mid \mathbf{Y})} \\
&= I(\mathbf{X};\mathbf{Y}) - qI(X;Y)
\end{aligned}$$

which is to be compared to Lemma 3.3. This proves that if applying our second bound with such an i.i.d. distribution $\mathbf{Y}'$ would lead to a bound that would be worse than the first upper bound (3.4).

A.5 A Discussion about Masking

In chapter 3, we supposed that there was no masking in the AES protocols. Nowadays, as side-channel analysis becomes a real threat, designers have started to invent new types of security against side-channel analysis. Masking is a good way to improve the security of a chip. Indeed, the higher the order of masking is, the more difficult it is to break the security of an embedded system.

Even if the security of the chip increases, it is though possible to recover the secret key. Nicolas Bruneau et al. mathematically proved that the best possible attack in case of masking is the maximum likelihood [16]. We can also cite once again Duc's paper where he showed that the order of making exponentially increases the security of a chip in the sens that the number of traces needed to recover the secret key are much higher [29, Equation (10)].

According to the leakage model of [16], we obtain the following framework for the communication channel.

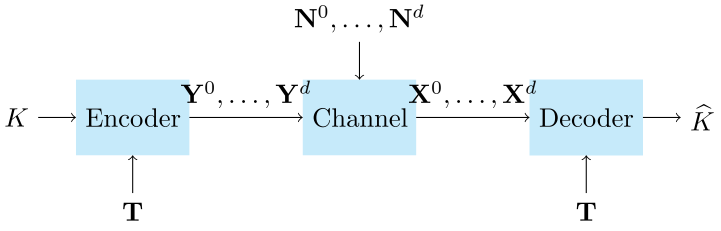

$\mathbf{N}^0, \ldots, \mathbf{N}^d$
K
Encoder
$\mathbf{Y}^0, \ldots, \mathbf{Y}^d$
Channel
$\mathbf{X}^0, \ldots, \mathbf{X}^d$
Decoder
$\widehat{K}$
$\mathbf{T}$
$\mathbf{T}$

Figure A.1: Representation of Side-Channel with masks

We consider that the masking order is d. In Figure A.1, the notations are the following:

- the d shares of sensitive variables are represented by $\mathbf{Y}^0, \ldots, \mathbf{Y}^d$.
- $\mathbf{N}^0, \ldots, \mathbf{N}^d$ are the d shares of additive noise.
- The d shares of the traces are $\mathbf{X}^0, \ldots, \mathbf{X}^d$.

The Markov chain with masking security, therefore becomes:

$$(K, \mathbf{T}) \longrightarrow (\mathbf{Y}^0, \ldots, \mathbf{Y}^d, \mathbf{T}) \longrightarrow (\mathbf{X}^0, \ldots, \mathbf{X}^d, \mathbf{T}). \longrightarrow \widehat{K} \tag{A.1}$$

This means that Lemma 3.2 can be adapted to masking and becomes

$$H(K) - (1 - \mathrm{P}_s) \log_2(2^n - 1) - H_2(\mathrm{P}_s) \leq I(\mathbf{X}^0 \ldots \mathbf{X}^d; K \mid \mathbf{T}). \tag{A.2}$$

This means that, one again, the Mutual Information between the traces and the key is relevant to calculate a bound on the success rate. However, estimating such Mutual Information is a open issue.

A.6 About Mismatched Decoding

The upper-bound obtained by Theorem 3.1 is always true for any distinguisher. Moreover, as we have based our calculations with the best possible case for the attacker. Indeed, the lower bound is obtained because we have supposed that the attacker knows the leakage model and therefore the distributions $\mathbb{P}$.

However, in many cases, the attacker only knows an estimation of the leakage distribution (noted $\widehat{\mathbb{P}}$) that may not be exactly equal to $\mathbb{P}$. In this case, our bound is still correct since the knowledge of $\mathbb{P}$ is the best possible case for the attacker. In this filed, François-Xavier Standaert proposed the notion of *Perceived Information* [30] as a metric to measure the impact of the estimation distribution $\widehat{\mathbb{P}}$ on the mutual information. In this section, we first re-write the Shannon channel coding theorem to show that it is possible to recover the secret key if perceived information is strictly positive (cf. Subsection A.6.1). Then we discuss about an information theoretic paper written by Neri Merhav and Amos Lapidoth in 1994 that deals with mismatch decoding [58](cf. Subsection A.6.2).

A.6.1 The Channel Coding Theorem with Divergence

In Information Theory, the Channel Coding Theorem written and proved by C.E. Shannon in [79] shows that, it is possible to send a message with an arbitrary small amount of error through a channel, as long as the rate of the message (i.e. the number of sent bits) is lower than the mutual information of this channel.

In his demonstration, Shannon supposed that the channel was perfectly known, meaning that the probability $\mathbb{P}(y|x)$ was known. In our case, we only have $\widehat{\mathbb{P}}(y|x)$ at our disposal. This means that even if Shannon's theory is true, we do not how far we can go with our estimation. Thus, we will re-write the proof of the channel-coding theorem with the consideration that the attacker makes an error of estimation.

Theorem 1.1 (Channel Coding with Divergence). *Let us consider a channel noted $(\mathcal{Y}, \mathbb{P}(y|x), \mathcal{X})$, where the decoder only knows $\widehat{\mathbb{P}}(y|x)$ an estimation of $\mathbb{P}(y|x)$. We suppose furthermore that the decoder perfectly knows the distribution of X (i.e. $\mathbb{P}(x)$).*

Let a message M to be sent taken uniformly in a set $\mathcal{M}$ and to be sent over a block of length q. The rate of the transmission is thus $R = \frac{\log_2(|\mathcal{M}|}{q}$. Let $\varepsilon > 0$, there exists a code $\mathcal{C}$ such that the probability of error is smaller than ε as long as $R < I(X;Y) - D(\mathbb{P}(y|x)||\widehat{\mathbb{P}}(y|x)) - \varepsilon$.

To prove Theorem 1.1, we first show the law of great numbers for a random variable taken under another distribution.

Lemma 1.1. *Let* $\mathbf{X}$ *a random i.i.d. vector of size q following the distribution $\mathbb{P}$ and $\widehat{\mathbb{P}}$ another distribution taking its values in the same set as* $\mathbf{X}$. *Then, we have:*

$$-\frac{1}{q}\log_2 \widehat{\mathbb{P}}(\mathbf{X}) \underset{q\to\infty}{\longrightarrow} H(X) + \mathrm{D}(\mathbb{P}||\widehat{\mathbb{P}})$$

This is a convergence in probability.

Proof for Lemma 1.1. We use the Bienaymé-Chebychev inequality to show the convergence. Let $\varepsilon > 0$, we know that:

$$\mathbb{P}\left[|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}(\mathbf{X}) - \mathbb{E}\left[-\frac{1}{n}\log_2 \widehat{\mathbb{P}}(\mathbf{X})\right]| \geq \varepsilon\right] \leq \frac{q\mathsf{Var}}{q^2\varepsilon^2}$$
$$\mathbb{P}\left[|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}(\mathbf{X}) - H(X) + \mathrm{D}(\mathbb{P}||\widehat{\mathbb{P}})| \geq \varepsilon\right] \leq \frac{\mathsf{Var}}{q\varepsilon^2}$$

Therefore, we have:

$$\mathbb{P}\left[|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}(\mathbf{X}) - H(X) + \mathrm{D}(\mathbb{P}||\widehat{\mathbb{P}})| \geq \varepsilon\right] \longrightarrow 0$$

As this is true for any $\varepsilon > 0$, this proves the lemma. □

We have now shown that the law of great numbers leads to a value which is $H(X) + \mathrm{D}(\mathbb{P}||\widehat{\mathbb{P}})$. In order to prove the Channel-Coding Theorem, C.E Shannon defined subsets of $\mathcal{X}^q \times \mathcal{Y}^q$ called typical sets. All the definitions of the typical sets, and their properties may be found in [23, Chapter 7]. We define here typical sets related to the estimation $\widehat{\mathbb{P}}$ of a random variable.

Definition 1.1 (Typical set related to $\widehat{\mathbb{P}}$)**.** Let $\mathbf{X}$ a random i.i.d. vector of size q following the distribution $\mathbb{P}$ and $\widehat{\mathbb{P}}$ the estimation of this distribution. Let $\varepsilon > 0$. The typical set related to $\widehat{\mathbb{P}}$ is defined as:

$$\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}} = \{\mathbf{x} \in \mathcal{X}^q \mid |-\frac{1}{q}\log_2 \widehat{\mathbb{P}}(\mathbf{x}) - H(X) - \mathrm{D}(\mathbb{P}||\widehat{\mathbb{P}})| \leq \varepsilon\}$$

For two random vectors $\mathbf{X}$ and $\mathbf{Y}$, we can also define the joint typical set.

Definition 1.2 (Joint Typical set related to $\widehat{\mathbb{P}}$)**.** Let $\mathbf{X}$ and $\mathbf{Y}$ two i.i.d. random vectors of length q each. $(\mathbf{X}, \mathbf{Y})$ follows the distribution $\mathbb{P}_{XY}$ and the estimation is $\widehat{\mathbb{P}}_{XY}$. The marginal distribution of $\mathbf{X}$ is $\mathbb{P}_X$ and the marginal distribution of $\mathbf{Y}$ is $\mathbb{P}_Y$. Let $\varepsilon > 0$. The joint typical set related to $\widehat{\mathbb{P}}$ is noted $\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}$. $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^q \times \mathcal{Y}^q$ belongs to $\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}$ if and only if the three conditions below are verified:

1. $|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}_X(\mathbf{x}) - H(X) - \mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X)| \leq \varepsilon$;
2. $|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}_Y(\mathbf{y}) - H(Y) - \mathrm{D}(\mathbb{P}_Y||\widehat{\mathbb{P}}_Y)| \leq \varepsilon$;
3. $|-\frac{1}{q}\log_2 \widehat{\mathbb{P}}_{XY}(\mathbf{x},\mathbf{y}) - H(X,Y) - \mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})| \leq \varepsilon$.

In [23], Cover and Thomas prove that the joint typical set has several very interesting properties. Here, with the joint typical set related to $\widehat{\mathbb{P}}$, we adapt these properties so they can fit with the estimation.

Lemma 1.2 (Properties of the joint typical set related to $\widehat{\mathbb{P}}$)**.** *Let* $(\mathbf{X},\mathbf{Y})$ *a random vector of length* q *drawn i.i.d. according to* $\mathbb{P}_{XY}$ *and estimated by* $\widehat{\mathbb{P}}_{XY}$*. Let* $\varepsilon > 0$*. We have:*

1. $\mathbb{P}\left[(\mathbf{X},\mathbf{Y}) \in \mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}\right] \longrightarrow 1$ *as* $q \to \infty$*;*
2. $|\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}| \leq 2^{q(H(X,Y)+\mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})+\varepsilon)}$
3. *For* $\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}$ *two independent random vectors such that* $\widetilde{\mathbf{X}} \sim \mathbb{P}_X$ *and* $\widetilde{\mathbf{Y}} \sim \mathbb{P}_Y$ *the marginals of* $\mathbb{P}_{XY}$*, we have:*

$$\mathbb{P}\left[(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}) \in \mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}\right] \leq 2^{-n(I(X;Y)-\mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})+\mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X)+\mathrm{D}(\mathbb{P}_Y||\widehat{\mathbb{P}}_Y)-3\varepsilon)}$$

The last property of the Lemma 1.2 seems quite heavy with the number of divergences. However, in our case, we suppose that the attacker knows the distribution of $\mathbf{Y}$, meaning that $\mathrm{D}(\mathbb{P}_y||\widehat{\mathbb{P}}_Y) = 0$. Furthermore, we have the relation between $\mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X)$ and $\mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})$:

$$\mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X) - \mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY}) = -\mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X}) \tag{A.3}$$

Proof of Lemma 1.2. We prove each term of the lemma one by one. Let $\varepsilon > 0$.

1. By the law of large numbers we know that there exist n_1, n_2, n_3 such that:

$$\mathbb{P}\left[|-\frac{1}{n_1}\log_2 \widehat{\mathbb{P}}_X(\mathbf{X}) - H(X) - \mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X)| > \varepsilon\right] \leq \frac{\varepsilon}{3}$$
$$\mathbb{P}\left[|-\frac{1}{n_2}\log_2 \widehat{\mathbb{P}}_Y(\mathbf{Y}) - H(Y) - \mathrm{D}(\mathbb{P}_Y||\widehat{\mathbb{P}}_Y)| > \varepsilon\right] \leq \frac{\varepsilon}{3}$$
$$\mathbb{P}\left[|-\frac{1}{n_3}\log_2 \widehat{\mathbb{P}}_{XY}(\mathbf{X},\mathbf{Y}) - H(X,Y) - \mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})| > \varepsilon\right] \leq \frac{\varepsilon}{3}$$

Taking $n = \max\{n_1, n_2, n_3\}$, and the union of the three probabilities, we obtain, for every $q \geq n$:

$$\mathbb{P}\left[(\mathbf{X},\mathbf{Y}) \notin \mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}\right] \leq \varepsilon$$

This proves the first part of the lemma.

2. We have the following inequality:

$$
\begin{aligned}
1 &= \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}^q\times\mathcal{Y}^q} \widehat{\mathbb{P}}_{XY}(\mathbf{x},\mathbf{y}) \\
&\geq \sum_{\mathbf{x},\mathbf{y}\in\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}} \widehat{\mathbb{P}}_{XY}(\mathbf{x},\mathbf{y}) \\
&\geq |\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}| 2^{-n(H(X,Y)+\mathrm{D}(\mathbb{P}_{XY}||\widehat{\mathbb{P}}_{XY})+\varepsilon)}
\end{aligned}
$$

This show the second part of the lemma.

3. We consider $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ defined in the lemma. We have:

$$
\begin{aligned}
\mathbb{P}\left[(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}})\in\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}\right] &= \sum_{\mathbf{x},\mathbf{y}\in\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}} \widehat{\mathbb{P}}_{X}(\mathbf{x})\widehat{\mathbb{P}}_{Y}(\mathbf{y}) \\
&\leq |\mathcal{A}^{\varepsilon}_{\widehat{\mathbb{P}}_{XY}}| 2^{-n(H(X)+\mathrm{D}(\mathbb{P}_X||\widehat{\mathbb{P}}_X)-\varepsilon)} 2^{-n(H(Y)+\mathrm{D}(\mathbb{P}_Y||\widehat{\mathbb{P}}_Y)-\varepsilon)}
\end{aligned}
$$

Putting this with the proof result of item 2, we obtain the inequality.

□

With all of these tools, we are now able to proof the Channel-Coding Theorem with divergence. This proof will be mainly inspired by the proof of the Coding-Channel Theorem written by Cover and Thomas in [23]. Our main contribution is to add the divergence into the proof and use typical sets including divergence to do so.

Proof of the Channel-Coding Theorem. To prove the Channel-Coding theorem, Shannon did not consider the probability of error of one particular code, but the average of the probability of error, taken over all the possible codebooks for a given message. Let us consider a rate R and the length q of the sent vector of the message. Shannon proved that the average probability of error over all the codebooks, is equal to the average probability of error over all the codebooks supposing that one particular index was sent. Let us consider a massage modeled by the random equiprobable variable M and the possible set of M is $\{1,\ldots,2^{qR}\}$. $\mathbf{Y}$ is therefore a function of M. Let us suppose that we send a 1 over the channel. Then, the average probability of error over all the codebooks knowing that a 1 is sent is noted $\mathbb{P}(\mathcal{E}|M=1)$.

We consider the following decoding scheme: $\mathbf{y}(1)$ is sent over the noisy channel and $\mathbf{x}$ is received. The decoder tests for which $i\in\{1,\ldots,2^{qR}\}$ the $\mathbf{y}(i)$ and $\mathbf{x}$ are jointly typical with probability $\widehat{\mathbb{P}}_{XY}$. An error occurs if $\mathbf{y}(1)$ and $\mathbf{x}$ are not jointly typical, or if there is another $i\neq 1$ such that $\mathbf{y}(i)$ and $\mathbf{x}$ are jointly typical. Let us note B_i the event: $\mathbf{y}(i)$ and $\mathbf{x}$ are jointly typical. We notice that if index 1 is sent, then, for any $i\neq 1$, $\mathbf{Y}(i)$ and $\mathbf{X}$ will be independent as the code is chosen randomly. The average probability of error over all the codebooks is therefore:

$$
\begin{aligned}
\mathbb{P}(\mathcal{E}|M=i) &= \mathbb{P}\left[B_1^c\cup B_2\cup\ldots\cup B_{2^{qR}}\right] \\
&\leq \mathbb{P}(B_1^c)+\sum_{i=2}^{2^{qR}}\mathbb{P}(B_i)
\end{aligned}
$$

Let $\varepsilon > 0$. According to Lemma 1.2, we know that there exists q large enough such that $\mathbb{P}(B_1^c) \leq \varepsilon$. Moreover, for B_i with $i \neq 1$, the probabilities are independent. Therefore, we have, according to Lemma 1.2 we have: $\mathbb{P}(B_i) \leq 2^{-n(I(X;Y)-\mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X})-3\varepsilon)}$ Therefore, we have, for $\varepsilon > 0$ and q sufficiently large:

$$\begin{aligned}
\mathbb{P}(\mathcal{E}|M=i) &\leq \varepsilon + \sum_{i=2}^{2^{qR}} 2^{-q(I(X;Y)-\mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X})-3\varepsilon)} \\
&\leq \varepsilon + (2^{qR}-1)2^{-q(I(X;Y)-\mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X})-3\varepsilon)} \\
&\leq \varepsilon + 2^{q(R-I(X;Y)+\mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X})+3\varepsilon)} \\
&\leq 2\varepsilon \quad \text{if} \quad R - I(X;Y) + \mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X}) + 3\varepsilon < 0
\end{aligned}$$

□

Remark 1.1. We notice that $I(X;Y) - \mathrm{D}(\mathbb{P}_{Y|X}||\widehat{\mathbb{P}}_{Y|X})$ is equal to the *Perceived information* metric proposed by François-Xavier Standeart in [30]. This means that, in a side-channel context, if the perceived information if positive, it is possible to recover the secret key of the device.

However, we are not able to tell how fast the key recovery will be with this approach since this calculation only shows the achievability o fa coding rate.

A.6.2 Discussion About Merhav's Paper

In 1994, Merhav et al. published an article dealing with mismatch decoding in information theory. When the decoding is performed with a maximum likelihood distinguisher based on an estimation of the distribution $\widehat{\mathbb{P}}$ instead of $\mathbb{P}$, the article show that the the maximum achievable rate of transmission exists and can be calculated. Moreover, if the coding rate is higher than this limit, the probability of error tends towards 1. In side-channel analysis, to result does not tell how fast the probability of success of the attack P_s will converge to 1.

Appendix B

Appendix about the Monobit Leakages

B.1 Proof of Lemma 5.3

The MIA distinguisher is expressed as

$$\mathcal{D}(k) = I(Y(k^*)+N; Y(k)) = h(Y(k^*)+N) - h(Y(k^*)+N \mid Y(k)). \tag{B.1}$$

From Section 5.3.1, $Y(k^*)$ knowing $Y(k)$ is a binary random variable with probability $\kappa(k)$. As N is Gaussian independent from $Y(k)$, the pdf of $Y(k^*)+N$ knowing $Y(k)$ is a Gaussian mixture that can take two forms:

$$p_{\kappa(k)}(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma}[\kappa(k)e^{-\frac{(x-1)^2}{2\sigma^2}} + (1-\kappa(k))e^{-\frac{(x+1)^2}{2\sigma^2}}] \\ \frac{1}{\sqrt{2\pi}\sigma}[\kappa(k)e^{-\frac{(x+1)^2}{2\sigma^2}} + (1-\kappa(k))e^{-\frac{(x-1)^2}{2\sigma^2}}] \end{cases}, \tag{B.2}$$

By symmetry, their entropy $h(Y(k^*)+N \mid Y(k))$ will be the same and we can take any of these pdfs. Letting φ be the standard normal density, we can write

$$\begin{aligned} p_{\kappa(k)}(x) &= p_{1/2}(x) - 2(1/2-\kappa(k))\varphi(x)e^{-\frac{1}{\sigma^2}}\sinh(\frac{x}{\sigma^2}) && \text{(B.3)} \\ &= p_{1/2}(x)(1-2(1/2-\kappa(k))\tanh(\frac{x}{2\sigma^2}). && \text{(B.4)} \end{aligned}$$

where

$$p_{1/2}(x) = \frac{1}{2\sqrt{2\pi}\sigma}[e^{-\frac{(x-1)^2}{2\sigma^2}} + e^{-\frac{(x+1)^2}{2\sigma^2}}] = \frac{1}{\sigma}e^{-\frac{1}{2\sigma^2}}\varphi(\frac{x}{\sigma})\cosh(\frac{x}{\sigma^2}). \tag{B.5}$$

For notational convenience define $\varepsilon = 2(1/2-\kappa(k))$, $p = p_{1/2}(x)$, and $t = \tanh(x)$. Then

$$\begin{aligned} I(X;Y(k)) &= h(Y(k^*)+N) - h(Y(k^*)+N \mid Y(k)) && \text{(B.6)} \\ &= -\int p\log_2 p + \int (p(1-\varepsilon t))\log_2(p(1-\varepsilon t)) && \text{(B.7)} \\ &= -\int \varepsilon pt\log_2 p + \int p\log_2(1-\varepsilon t) - \int p\varepsilon t\log_2(1-\varepsilon t). && \text{(B.8)} \end{aligned}$$

The first term vanishes since p is even and t odd. We apply a Taylor expansion:

$$I(X;Y(k)) = \int p[-\varepsilon t - \frac{\varepsilon^2 t^2}{2} - \frac{\varepsilon^3 t^3}{3} + O(\varepsilon^4)] - \int \varepsilon pt[-\varepsilon t - \frac{\varepsilon^2 t^2}{2} - \frac{\varepsilon^3 t^3}{3} + O(\varepsilon^4)]. \tag{B.9}$$

The odd terms of the expansion are null as t is odd and p even. We therefore obtain:

$$I(X;Y(k)) = \int p[-\frac{\varepsilon^2 t^2}{2} + O(\varepsilon^4)] - \int [-\varepsilon^2 pt^2 + O(\varepsilon^4)] = \int \frac{\varepsilon^2 pt^2}{2} + O(\varepsilon^4). \tag{B.10}$$

Thus, finally,

$$\mathcal{D}(k) = 2\log_2(e)(1/2-\kappa(k))^2 g(\sigma), \tag{B.11}$$

where

$$g(\sigma) = \frac{1}{\sigma}e^{-\frac{1}{2\sigma^2}}\int_{\mathbb{R}} \varphi(\frac{x}{\sigma})\cosh(\frac{x}{\sigma^2})\tanh^2(\frac{x}{\sigma^2})\mathrm{d}x. \tag{B.12}$$

There are several ways to express $g(\sigma)$. For example, we have:

$$g(\sigma) = e^{-\frac{1}{2\sigma^2}} \int_{\mathbb{R}} \varphi(x) \cosh(\frac{x}{\sigma}) \tanh^2(\frac{x}{\sigma}) \mathrm{d}x. \tag{B.13}$$

This expression can be reduced to:

$$g(\sigma) = \frac{1}{2}\mathbb{E}_X \left[\tanh^2(\frac{X}{\sigma} + \frac{1}{\sigma^2}) + \tanh^2(\frac{X}{\sigma} - \frac{1}{\sigma^2})\right], \tag{B.14}$$

where $X \sim \mathcal{N}(0,1)$. By the dominated convergence theorem ($\tanh^2(\frac{X}{\sigma} + \frac{1}{\sigma^2})$ is always smaller than 1) when $\sigma \to 0$, we obtain $g(0) = 1$ and when $\sigma \to \infty$ we obtain the equivalent $\frac{1}{\sigma^2}$.

B.2 Proof of Lemma 5.4

The success exponent is defined by

$$\mathsf{SE} = \frac{\mathbb{E}[\widehat{\mathcal{D}}(k^*) - \widehat{\mathcal{D}}(k)]^2}{2\mathrm{Var}(\widehat{\mathcal{D}}(k^*) - \widehat{\mathcal{D}}(k))}. \tag{B.15}$$

where in our case

$$\widehat{\mathcal{D}}(k) = \frac{1}{q\sqrt{1+\sigma^2}}\Big|\sum_{i=1}^{q} X_i Y_i(k)\Big|. \tag{B.16}$$

First for large q we can consider that $\mathbb{E}[|\sum_i X_i Y_i(k)|] = |\mathbb{E}[\sum_i X_i Y_i(k)]|$.

$$\mathbb{E}[\widehat{\mathcal{D}}(k)] = |\mathbb{E}[XY(k)]| = \frac{2 \times |1/2 - \kappa(k)|}{\sqrt{1+\sigma^2}} \tag{B.17}$$

hence

$$\mathbb{E}[\widehat{\mathcal{D}}(k^*) - \widehat{\mathcal{D}}(k)] = \frac{1 - 2 \times |1/2 - \kappa(k)|}{\sqrt{1+\sigma^2}}. \tag{B.18}$$

Secondly we have

$$\mathrm{Var}(\widehat{\mathcal{D}}(k^*) - \widehat{\mathcal{D}}(k)) = \frac{1}{q^2(1+\sigma^2)}\mathrm{Var}\Big(\Big|\sum_{i=1}^{q} X_i Y_i(k^*)\Big| - \Big|\sum_{i=1}^{q} X_i Y_i(k)\Big|\Big). \tag{B.19}$$

To remove the absolute values, we distinguish two cases whether the sum is positive or negative. We consider that q is large enough to have strictly positive or negative values.

$$\begin{aligned}
\mathrm{Var}(\widehat{\mathcal{D}}(k^*) - \widehat{\mathcal{D}}(k)) &= \frac{1}{q^2(1+\sigma^2)}\mathrm{Var}\Big(\sum_{i=1}^{q} X_i Y_i(k^*) \mp \sum_{i=1}^{q} X_i Y_i(k)\Big) && \text{(B.20)}\\
&= \frac{1}{q^2(1+\sigma^2)}\mathrm{Var}\Big(\sum_{i=1}^{q} X_i \big(Y_i(k^*) \mp Y_i(k)\big)\Big) && \text{(B.21)}\\
&= \frac{1}{q(1+\sigma^2)}\mathrm{Var}\big(X\big(Y(k^*) \mp Y(k)\big)\big) && \text{(B.22)}\\
&= \frac{1}{q(1+\sigma^2)}\mathrm{Var}\big((Y(k^*) + N)\big(Y(k^*) \mp Y(k)\big)\big) && \text{(B.23)}\\
&= \frac{1}{q(1+\sigma^2)}\mathrm{Var}\big(\mp Y(k^*)Y(k) + N(Y(k^*) \mp Y(k))\big). && \text{(B.24)}
\end{aligned}$$

The variance term is the difference of the two following quantities

$$\mathbb{E}\Big[(\mp Y(k^*)Y(k) + N(Y(k^*) \mp Y(k)))^2\Big] = 1 + 2\sigma^2(1 - 2|{}^1\!/\!_2 - \kappa(k)|) \tag{B.25}$$

$$\mathbb{E}\Big[\mp Y(k^*)Y(k) + N(Y(k^*) \mp Y(k))\Big]^2 = \Big(2({}^1\!/\!_2 - \kappa(k))\Big)^2. \tag{B.26}$$

Combining all the above expressions we obtain (5.33).

B.3 Proof of Lemma 5.5

To prove the success rate of KSA, we first need an estimator for the cumulative density function. We take as kernel a function Φ as simple as possible i.e. the Heaviside function $\Phi(x) = 0$ if $x < 0$ and $\Phi(x) = 1$ if $x \geq 0$.

With this function and for $x \in \mathbb{R}$, we can estimate $F(x|Y(k) = 1) - F(x)$ by the following estimator:

$$\widetilde{F}(x|Y(k) = 1) - \widetilde{F}(x) = \frac{\sum_{i|Y_i(k)=1} \Phi(x - X_i)}{\sum_{i|Y_i(k)=1} 1} - \frac{\sum_i \Phi(x - X_i)}{q}. \tag{B.27}$$

We suppose that q is large enough to consider that $\sum_{i|Y_i(k)=1} 1 = \frac{q}{2}$ (by the law of large numbers). Therefore we have:

$$\widetilde{F}(x|Y(k) = 1) - \widetilde{F}(x) = \frac{\sum_{i|Y_i(k)=1} \Phi(x - X_i)}{q} - 2\frac{\sum_i \Phi(x - X_i)}{q}. \tag{B.28}$$

We notice that $\sum_{i|Y_i(k)=1} \Phi(x - X_i) = \frac{1}{2}\sum_i (Y_i(k) + 1)\Phi(x - X_i)$. Therefore

$$\widetilde{F}(x|Y(k) = 1) - \widetilde{F}(x) = \frac{1}{q}\sum_{i=1}^{q} Y_i(k)\Phi(x - X_i). \tag{B.29}$$

This estimator is a sum of i.i.d. random variables. We can therefore apply the central limit theorem.

$$\begin{aligned}
\mathbb{E}[\widetilde{F}(x|Y(k) = 1) - \widetilde{F}(x)] &= \mathbb{E}[Y(k)\Phi(x - X_i)] && \text{(B.30)}\\
&= \mathbb{E}[Y(k)\Phi(x - Y(k^*) - N)] && \text{(B.31)}\\
&= \frac{1}{2}(\kappa(k) - 0.5)\Big(\operatorname{erf}\Big(\frac{1-x}{\sigma\sqrt{2}}\Big) + \operatorname{erf}\Big(\frac{1+x}{\sigma\sqrt{2}}\Big)\Big). && \text{(B.32)}
\end{aligned}$$

The maximum of the absolute value is for $x = 0$ and we obtain:

$$\|\mathbb{E}[\widetilde{F}(x|Y(k) = 1) - \widetilde{F}(x)]\|_\infty = |0.5 - \kappa(k)| \operatorname{erf}\Big(\frac{1}{\sigma\sqrt{2}}\Big). \tag{B.33}$$

We notice that $\|\mathbb{E}[\widetilde{F}(x|Y(k)=1)-\widetilde{F}(x)]\|_\infty = \|\mathbb{E}[\widetilde{F}(x|Y(k)=-1)-\widetilde{F}(x)]\|_\infty$. To calculate the variance, we consider that $x=0$ as it is the value that maximizes the expectation of the distinguisher.

$$\mathrm{Var}(\widehat{\mathfrak{D}}(k^*)-\widehat{\mathfrak{D}}(k)) = \mathrm{Var}\Big(\frac{1}{q}\Big(\sum_{i=1}^{q}\Phi(x-X_i)(Y_i(k^*)-Y_i(k))\Big)\Big) \tag{B.34}$$

The computation of this variance gives:

$$\mathrm{Var}(\widehat{\mathfrak{D}}(k^*)-\widehat{\mathfrak{D}}(k)) = 2(0.5-|0.5-\kappa(k)|) - \mathrm{erf}\Big(\frac{1}{\sigma\sqrt{2}}\Big)^2(0.5-|0.5-\kappa(k)|)^2. \tag{B.35}$$

Overall, the success exponent is:

$$\mathsf{SE} = \frac{1}{2}\min_{k\neq k^*}\frac{\mathrm{erf}\big(\frac{1}{\sqrt{2}\sigma}\big)^2(1/2-|1/2-\kappa(k)|)}{2-\mathrm{erf}\big(\frac{1}{\sqrt{2}\sigma}\big)^2(1/2-|1/2-\kappa(k)|)}. \tag{B.36}$$

B.4 Proof of Lemma 5.6

For MIA, we refer to [36, Section 5.3] for the theoretical justifications. In order to obtain a simple closed-form expression of the success exponent, we suppose that $\sigma \gg 1$ and that the probability density functions are all Gaussian. This means that $X|Y(k)$ is a Gaussian random variable of standard deviation $\sqrt{4\kappa(k)(1-\kappa(k))+\sigma^2}$. Moreover, we will keep only the first order approximation in $\mathsf{SNR}=\sigma^{-2}$ of the SE.

$$\begin{aligned}
h(X|Y(k))-h(X|Y(k^*)) &= \frac{1}{2}\log_2(2\pi e\cdot(4\kappa(k)(1-\kappa(k))+\sigma^2)-\frac{1}{2}\log_2(2\pi e\cdot\sigma^2) && \text{(B.37)}\\
&= \frac{1}{2}\log_2\frac{4\kappa(k)(1-\kappa(k))+\sigma^2}{\sigma^2} && \text{(B.38)}\\
&\approx \frac{\log_2(e)4\kappa(k)(1-\kappa(k))}{2\sigma^2} && \text{(B.39)}
\end{aligned}$$

The Fisher information of a Gaussian random variable of standard deviation ζ is equal to $\frac{1}{\zeta^2}$. Therefore the Fisher information of X knowing $Y=y(k)$ is:

$$F(X|Y(k)=y(k)) = \frac{1}{4\kappa(k)(1-\kappa(k))+\sigma^2}. \tag{B.40}$$

As this value does not depend on the value of $Y(k)$, we have:

$$\begin{aligned}
F(X|Y(k)) &= \frac{1}{4\kappa(k)(1-\kappa(k))+\sigma^2} && \text{(B.41)}\\
J(X|Y(k))-J(X|Y(k^*)) &= \frac{1}{4\kappa(k)(1-\kappa(k))+\sigma^2}-\frac{1}{\sigma^2} && \text{(B.42)}\\
&\approx -\frac{\kappa(k)(1-\kappa(k))}{\sigma^4}. && \text{(B.43)}
\end{aligned}$$

Last, we have to calculate $\mathrm{Var}(-\log_2 p(X|Y(k) = y(k)))$. Let $\zeta^2 = \sigma^2 + 4\kappa(k)(1-\kappa(k))$ and C the normalization constant. We have:

$$\begin{aligned}
\mathrm{Var}(-\log_2 p(X|Y(k) = y(k))) &= \mathrm{Var}\Big(-\log_2\Big(C\exp\Big(-{}^1\!/\!_2\frac{(X-\mu)^2}{\zeta^2}\Big)\Big)\Big) && \text{(B.44)}\\
&= \mathrm{Var}\Big(-\log_2(C) + {}^1\!/\!_2\frac{(X-\mu)^2}{\zeta^2}\Big) && \text{(B.45)}\\
&= \frac{1}{4}\mathrm{Var}\Big(\frac{(X-\mu)^2}{\zeta^2}\Big) = \frac{1}{4\zeta^4}\mathrm{Var}(X^2) && \text{(B.46)}\\
&= \frac{1}{4(\sigma^2 + 4\kappa(k)(1-\kappa(k)))^2}2(1+\sigma^2)^2 \approx \frac{1}{2}. && \text{(B.47)}
\end{aligned}$$

Overall, the success exponent defined in [36, Proposition 6] can be simplified in the case of monobit leakage as:

$$\mathsf{SE} \approx \min_{k \neq k^*} 4\frac{\log_2(e)^2\kappa(k)^2(1-\kappa(k))^2}{\sigma^4}. \tag{B.48}$$

Appendix C

The OpenSSL source code

C.1 The OpenSSL AES Encryption Code

We have copied here the OpenSSL C code for the encryption function. We notice that this is a straightline code, and that there is a use of Look Up Tables (the T boxes) that may cause the non constant time.

```
void AES_encrypt(const unsigned char *in, unsigned char *out,
   const AES_KEY *key) {

 const u32 *rk;
 u32 s0, s1, s2, s3, t0, t1, t2, t3;

 int r;



# 796 "aes_core.c" 3
((void)0)
# 796 "aes_core.c"
                        ;
 rk = key->rd_key;





 s0 = (((u32)(in)[0] << 24) ^ ((u32)(in)[1] << 16) ^ ((u32)(in)
    [2] << 8) ^ ((u32)(in)[3])) ^ rk[0];
 s1 = (((u32)(in + 4)[0] << 24) ^ ((u32)(in + 4)[1] << 16) ^ ((
    u32)(in + 4)[2] << 8) ^ ((u32)(in + 4)[3])) ^ rk[1];
 s2 = (((u32)(in + 8)[0] << 24) ^ ((u32)(in + 8)[1] << 16) ^ ((
    u32)(in + 8)[2] << 8) ^ ((u32)(in + 8)[3])) ^ rk[2];
 s3 = (((u32)(in + 12)[0] << 24) ^ ((u32)(in + 12)[1] << 16) ^
    ((u32)(in + 12)[2] << 8) ^ ((u32)(in + 12)[3])) ^ rk[3];


    t0 = Te0[(s0 >> 24)] ^ Te1[(s1 >> 16) & 0xff] ^ Te2[(s2 >>
       8) & 0xff] ^ Te3[s3 & 0xff] ^ rk[ 4];
    t1 = Te0[s1 >> 24] ^ Te1[(s2 >> 16) & 0xff] ^ Te2[(s3 >> 8)
        & 0xff] ^ Te3[s0 & 0xff] ^ rk[ 5];
```

```
t2 = Te0[s2 >> 24] ^ Te1[(s3 >> 16) & 0xff] ^ Te2[(s0 >> 8)
    & 0xff] ^ Te3[s1 & 0xff] ^ rk[ 6];
t3 = Te0[s3 >> 24] ^ Te1[(s0 >> 16) & 0xff] ^ Te2[(s1 >> 8)
    & 0xff] ^ Te3[s2 & 0xff] ^ rk[ 7];

s0 = Te0[t0 >> 24] ^ Te1[(t1 >> 16) & 0xff] ^ Te2[(t2 >> 8)
    & 0xff] ^ Te3[t3 & 0xff] ^ rk[ 8];
s1 = Te0[t1 >> 24] ^ Te1[(t2 >> 16) & 0xff] ^ Te2[(t3 >> 8)
    & 0xff] ^ Te3[t0 & 0xff] ^ rk[ 9];
s2 = Te0[t2 >> 24] ^ Te1[(t3 >> 16) & 0xff] ^ Te2[(t0 >> 8)
    & 0xff] ^ Te3[t1 & 0xff] ^ rk[10];
s3 = Te0[t3 >> 24] ^ Te1[(t0 >> 16) & 0xff] ^ Te2[(t1 >> 8)
    & 0xff] ^ Te3[t2 & 0xff] ^ rk[11];

t0 = Te0[s0 >> 24] ^ Te1[(s1 >> 16) & 0xff] ^ Te2[(s2 >> 8)
    & 0xff] ^ Te3[s3 & 0xff] ^ rk[12];
t1 = Te0[s1 >> 24] ^ Te1[(s2 >> 16) & 0xff] ^ Te2[(s3 >> 8)
    & 0xff] ^ Te3[s0 & 0xff] ^ rk[13];
t2 = Te0[s2 >> 24] ^ Te1[(s3 >> 16) & 0xff] ^ Te2[(s0 >> 8)
    & 0xff] ^ Te3[s1 & 0xff] ^ rk[14];
t3 = Te0[s3 >> 24] ^ Te1[(s0 >> 16) & 0xff] ^ Te2[(s1 >> 8)
    & 0xff] ^ Te3[s2 & 0xff] ^ rk[15];

s0 = Te0[t0 >> 24] ^ Te1[(t1 >> 16) & 0xff] ^ Te2[(t2 >> 8)
    & 0xff] ^ Te3[t3 & 0xff] ^ rk[16];
s1 = Te0[t1 >> 24] ^ Te1[(t2 >> 16) & 0xff] ^ Te2[(t3 >> 8)
    & 0xff] ^ Te3[t0 & 0xff] ^ rk[17];
s2 = Te0[t2 >> 24] ^ Te1[(t3 >> 16) & 0xff] ^ Te2[(t0 >> 8)
    & 0xff] ^ Te3[t1 & 0xff] ^ rk[18];
s3 = Te0[t3 >> 24] ^ Te1[(t0 >> 16) & 0xff] ^ Te2[(t1 >> 8)
    & 0xff] ^ Te3[t2 & 0xff] ^ rk[19];

t0 = Te0[s0 >> 24] ^ Te1[(s1 >> 16) & 0xff] ^ Te2[(s2 >> 8)
    & 0xff] ^ Te3[s3 & 0xff] ^ rk[20];
t1 = Te0[s1 >> 24] ^ Te1[(s2 >> 16) & 0xff] ^ Te2[(s3 >> 8)
    & 0xff] ^ Te3[s0 & 0xff] ^ rk[21];
t2 = Te0[s2 >> 24] ^ Te1[(s3 >> 16) & 0xff] ^ Te2[(s0 >> 8)
    & 0xff] ^ Te3[s1 & 0xff] ^ rk[22];
t3 = Te0[s3 >> 24] ^ Te1[(s0 >> 16) & 0xff] ^ Te2[(s1 >> 8)
    & 0xff] ^ Te3[s2 & 0xff] ^ rk[23];
    s0 = Te0[t0 >> 24] ^ Te1[(t1 >> 16) & 0xff] ^ Te2[(t2 >> 8)
        & 0xff] ^ Te3[t3 & 0xff] ^ rk[24];
    s1 = Te0[t1 >> 24] ^ Te1[(t2 >> 16) & 0xff] ^ Te2[(t3 >> 8)
        & 0xff] ^ Te3[t0 & 0xff] ^ rk[25];
    s2 = Te0[t2 >> 24] ^ Te1[(t3 >> 16) & 0xff] ^ Te2[(t0 >> 8)
        & 0xff] ^ Te3[t1 & 0xff] ^ rk[26];
    s3 = Te0[t3 >> 24] ^ Te1[(t0 >> 16) & 0xff] ^ Te2[(t1 >> 8)
        & 0xff] ^ Te3[t2 & 0xff] ^ rk[27];

    t0 = Te0[s0 >> 24] ^ Te1[(s1 >> 16) & 0xff] ^ Te2[(s2 >> 8)
        & 0xff] ^ Te3[s3 & 0xff] ^ rk[28];
    t1 = Te0[s1 >> 24] ^ Te1[(s2 >> 16) & 0xff] ^ Te2[(s3 >> 8)
        & 0xff] ^ Te3[s0 & 0xff] ^ rk[29];
    t2 = Te0[s2 >> 24] ^ Te1[(s3 >> 16) & 0xff] ^ Te2[(s0 >> 8)
        & 0xff] ^ Te3[s1 & 0xff] ^ rk[30];
    t3 = Te0[s3 >> 24] ^ Te1[(s0 >> 16) & 0xff] ^ Te2[(s1 >> 8)
        & 0xff] ^ Te3[s2 & 0xff] ^ rk[31];

    s0 = Te0[t0 >> 24] ^ Te1[(t1 >> 16) & 0xff] ^ Te2[(t2 >> 8)
        & 0xff] ^ Te3[t3 & 0xff] ^ rk[32];
    s1 = Te0[t1 >> 24] ^ Te1[(t2 >> 16) & 0xff] ^ Te2[(t3 >> 8)
        & 0xff] ^ Te3[t0 & 0xff] ^ rk[33];
    s2 = Te0[t2 >> 24] ^ Te1[(t3 >> 16) & 0xff] ^ Te2[(t0 >> 8)
        & 0xff] ^ Te3[t1 & 0xff] ^ rk[34];
    s3 = Te0[t3 >> 24] ^ Te1[(t0 >> 16) & 0xff] ^ Te2[(t1 >> 8)
        & 0xff] ^ Te3[t2 & 0xff] ^ rk[35];

    t0 = Te0[s0 >> 24] ^ Te1[(s1 >> 16) & 0xff] ^ Te2[(s2 >> 8)
        & 0xff] ^ Te3[s3 & 0xff] ^ rk[36];
    t1 = Te0[s1 >> 24] ^ Te1[(s2 >> 16) & 0xff] ^ Te2[(s3 >> 8)
        & 0xff] ^ Te3[s0 & 0xff] ^ rk[37];
    t2 = Te0[s2 >> 24] ^ Te1[(s3 >> 16) & 0xff] ^ Te2[(s0 >> 8)
        & 0xff] ^ Te3[s1 & 0xff] ^ rk[38];
    t3 = Te0[s3 >> 24] ^ Te1[(s0 >> 16) & 0xff] ^ Te2[(s1 >> 8)
        & 0xff] ^ Te3[s2 & 0xff] ^ rk[39];
    rk += key->rounds << 2;
# 944 "aes_core.c"
 s0 =
  (Te2[(t0 >> 24) ] & 0xff000000) ^
  (Te3[(t1 >> 16) & 0xff] & 0x00ff0000) ^
  (Te0[(t2 >> 8) & 0xff] & 0x0000ff00) ^
  (Te1[(t3 ) & 0xff] & 0x000000ff) ^
  rk[0];
 { (out)[0] = (u8)((s0) >> 24); (out)[1] = (u8)((s0) >> 16); (
    out)[2] = (u8)((s0) >> 8); (out)[3] = (u8)(s0); };
 s1 =
  (Te2[(t1 >> 24) ] & 0xff000000) ^
  (Te3[(t2 >> 16) & 0xff] & 0x00ff0000) ^
  (Te0[(t3 >> 8) & 0xff] & 0x0000ff00) ^
  (Te1[(t0 ) & 0xff] & 0x000000ff) ^
  rk[1];
 { (out + 4)[0] = (u8)((s1) >> 24); (out + 4)[1] = (u8)((s1) >>
     16); (out + 4)[2] = (u8)((s1) >> 8); (out + 4)[3] = (u8)(
    s1); };
 s2 =
  (Te2[(t2 >> 24) ] & 0xff000000) ^
  (Te3[(t3 >> 16) & 0xff] & 0x00ff0000) ^
  (Te0[(t0 >> 8) & 0xff] & 0x0000ff00) ^
  (Te1[(t1 ) & 0xff] & 0x000000ff) ^
  rk[2];
 { (out + 8)[0] = (u8)((s2) >> 24); (out + 8)[1] = (u8)((s2) >>
     16); (out + 8)[2] = (u8)((s2) >> 8); (out + 8)[3] = (u8)(
    s2); };
 s3 =
  (Te2[(t3 >> 24) ] & 0xff000000) ^
  (Te3[(t0 >> 16) & 0xff] & 0x00ff0000) ^
  (Te0[(t1 >> 8) & 0xff] & 0x0000ff00) ^
  (Te1[(t2 ) & 0xff] & 0x000000ff) ^
  rk[3];
 { (out + 12)[0] = (u8)((s3) >> 24); (out + 12)[1] = (u8)((s3)
    >> 16); (out + 12)[2] = (u8)((s3) >> 8); (out + 12)[3] = (
    u8)(s3); };
}
```

Bibliography

[1] Hassan Aly and Mohammed ElGayyar. Attacking AES Using Bernstein's Attack on Modern Processors. In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors, *Progress in Cryptology - AFRICACRYPT 2013, 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings*, volume 7918 of *Lecture Notes in Computer Science*, pages 127–139. Springer, 2013. 163

[2] Suguru Arimoto. On the converse to the coding theorem for discrete memoryless channels (corresp.). *IEEE Transactions on Information Theory*, 19(3):357–359, May 1973. 29, 30

[3] Lejla Batina and Matthew Robshaw, editors. *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*. Springer, 2014. 199, 200

[4] Normand J. Beaudry and Renato Renner. An intuitive proof of the data processing inequality. *Quantum Info. Comput.*, 12(5-6):432–441, May 2012. 38

[5] Daniel J. Bernstein. Cache-timing attacks on AES, April 14 2005. `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`. 122, 160

[6] Daniel J. Bernstein. Cache-timing attacks on AES, April 2005. `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`. 163

[7] Claude Berrou and Alain Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Communications*, 44(10):1261–1271, 1996. 14

[8] Sarani Bhattacharya, Chester Rebeiro, and Debdeep Mukhopadhyay. Hardware prefetchers leak: A revisit of SVF for cache-timing attacks. In *45th Annual IEEE/ACM International*

Symposium on Microarchitecture, MICRO 2012, Workshops Proceedings, Vancouver, BC, Canada, December 1-5, 2012, pages 17–23. IEEE Computer Society, 2012. 122

[9] Richard E. Blahut. *Principles and Practice of Information Theory*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987. 82

[10] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011. 8

[11] Éric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004. 28, 70, 73, 76, 81

[12] Éric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA. 97

[13] Billy Bob Brumley and Nicola Tuveri. Remote Timing Attacks Are Still Practical. In Vijay Atluri and Claudia Díaz, editors, *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 355–371. Springer, 2011. 121

[14] David Brumley and Dan Boneh. Remote Timing Attacks Are Practical. In *Proceedings of the 12th USENIX Security Symposium, Washington, D.C., USA, August 4-8, 2003*. USENIX Association, 2003. 121

[15] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is More - Dimensionality Reduction from a Theoretical Perspective. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 22–41. Springer, 2015. 59

[16] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks Will Fall Off – Higher-Order Optimal Distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 344–365. Springer, 2014. 177

[17] Mathieu Carbone, Sébastien Tiran, Sébastien Ordas, Michel Agoyan, Yannick Teglia, Gilles R. Ducharme, and Philippe Maurine. On adaptive bandwidth selection for efficient MIA. In Prouff [65], pages 82–97. 97

[18] Claude Carlet, Annelie Heuser, and Stjepan Picek. Trade-Offs for S-Boxes: Cryptographic Properties and Side-Channel Resilience. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, volume 10355 of *Lecture Notes in Computer Science*, pages 393–414. Springer, 2017. 72

[19] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, 2002. Second edition. ISBN-10: 0534243126 – ISBN-13: 978-0534243128. 99

[20] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA. 101, 121, 122

[21] Christophe Clavier. DPA Contest 2008–2009, Less than 50 traces allow to recover the key, September 6-9 2009. CHES Special Session 1: DPA Contest. Lausanne, Switzerland, (slides). 13

[22] Common Criteria Consortium. Common Criteria (*aka* CC) for Information Technology Security Evaluation (ISO/IEC 15408), 2013.
Website: `http://www.commoncriteriaportal.org/`. 96

[23] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, July 18 2006. ISBN-10: ISBN-10: 0471241954, ISBN-13: 978-0471241959, 2nd edition. 37, 38, 46, 50, 79, 179, 180, 181

[24] Joan Daemen and Vincent Rijmen. AES Proposal: Rijndael, September 3 1999. 52

[25] Joan Daemen and Vincent Rijmen. Rijndael for AES. In *AES Candidate Conference*, pages 343–348, 2000. 71

[26] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES – The Advanced Encryption Standard*. Springer, 2002. 8, 121

[27] Jean-Luc Danger, Nicolas Debande, Sylvain Guilley, and Youssef Souissi. High-order Timing Attacks. In *Proceedings of the First Workshop on Cryptography and Security in Computing Systems*, CS2 '14, pages 7–12, New York, NY, USA, 2014. ACM. 121

[28] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *J. Cryptographic Engineering*, 1(2):123–144, 2011. 97

[29] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015. 20, 29, 30, 57, 177

[30] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to Certify the Leakage of a Chip? In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014. 20, 22, 27, 98, 178, 182

[31] Yunsi Fei, A. Adam Ding, Jian Lao, and Liwei Zhang. A statistics-based success rate model for DPA and CPA. *J. Cryptographic Engineering*, 5(4):227–243, 2015. 73

[32] Yunsi Fei, Qiasi Luo, and A. Adam Ding. A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Prouff and Schaumont [68], pages 233–250. 23, 45, 72, 74, 75, 76

[33] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. Introduction to the Dirichlet Distribution and Related Processes. Technical Report 206, 2010. 130

[34] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *CHES, 10th International Workshop*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA. 70, 73, 77, 81, 97, 99, 102, 103, 109, 133, 136

[35] Vincent Grosso and François-Xavier Standaert. Masking Proofs Are Tight and How to Exploit it in Security Evaluations. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 385–412. Springer, 2018. 51

[36] Sylvain Guilley, Annelie Heuser, and Olivier Rioul. A Key to Success - Success Exponents for Side-Channel Distinguishers. In Alex Biryukov and Vipul Goyal, editors, *Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, volume 9462 of *Lecture Notes in Computer Science*, pages 270–290. Springer, 2015. 28, 30, 45, 72, 75, 77, 83, 84, 159, 187, 188

[37] Annelie Heuser, Michael Kasper, Werner Schindler, and Marc Stöttinger. A New Difference Method for Side-Channel Analysis with High-Dimensional Leakage Models. In Orr Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 365–382. Springer, 2012. 108

[38] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. A Theoretical Study of Kolmogorov-Smirnov Distinguishers — Side-Channel Analysis vs. Differential Cryptanalysis. In Prouff [65], pages 9–28. 23, 72, 75, 81

[39] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory. In Batina and Robshaw [3], pages 55–74. 14, 21, 22, 24, 28, 30, 34, 35, 55, 70, 71, 73, 76, 81, 85, 97, 98, 101, 105, 128

[40] Annelie Heuser and Michael Zohner. Intelligent Machine Homicide - Breaking Cryptographic Devices Using Support Vector Machines. In Werner Schindler and Sorin A. Huss, editors, *COSADE*, volume 7275 of *LNCS*, pages 249–264. Springer, 2012. 164

[41] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, 1996. 5

[42] Auguste Kerckhoffs. La cryptographie militaire (1). *Journal des sciences militaires*, 9:5–38, January 1883. http://en.wikipedia.org/wiki/Kerckhoffs_law. 7

[43] Su Min Kim Kim, Tan Tai Do, Tobias J. Oechtering, and Gunnar Peters. On the Entropy Computation of Large Complex Gaussian Mixture Distributions. *IEEE Transactions on Signal Processing*, 63(17):4710–4723, Sept 2015. 51

[44] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996. 27, 122

[45] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999. 27, 70, 73, 76, 81

[46] Yuichi Komano, Hideo Shimizu, and Shinichi Kawamura. Built-in determined sub-key correlation power analysis. Cryptology ePrint Archive, Report 2009/161, 2009. http://eprint.iacr.org/2009/161. 62

[47] W. Kozaczuk. *Enigma: how the German machine cipher was broken, and how it was read by the Allies in World War Two*. Foreign intelligence book series. University Publications of America, 1984. 6

[48] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to Estimate the Success Rate of Higher-Order Side-Channel Attacks. In Batina and Robshaw [3], pages 35–54. 28, 30, 72, 73

[49] Victor Lomné, Emmanuel Prouff, and Thomas Roche. Behind the Scene of Side Channel Attacks. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 506–525. Springer, 2013. 97, 103

[50] Stefan Mangard. Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004. San Francisco, CA, USA. 28

[51] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, `http://www.dpabook.org/`. 10, 11, 23

[52] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. 81

[53] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for All - All for One: Unifying Standard DPA Attacks. *Information Security, IET*, 5(2):100–111, 2011. ISSN: 1751-8709 ; Digital Object Identifier: 10.1049/iet-ifs.2010.0096. 20

[54] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011. 73

[55] James L. Massey. Guessing and entropy. In *Proceedings of 1994 IEEE International Symposium on Information Theory*, pages 204–, Jun 1994. 45, 46

[56] Seiichi Matsuda and Shiho Moriai. Lightweight Cryptography for the Cloud: Exploit the Power of Bitslice Implementation. In Prouff and Schaumont [68], pages 408–425. 106

[57] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai Shitz. On information rates for mismatched decoders. *IEEE Transactions on Information Theory*, 40(6):1953–1967, Nov 1994. 23

[58] Neri Merhav, Gideon Kaplan, Amos Lapidoth, and Shlomo Shamai. On information rates for mismatched decoders. *IEEE Trans. Information Theory*, 40(6):1953–1967, 1994. 178

[59] ST Microelectronics. STM32F4DISCOVERY Discovery kit with STM32F407VG MCU. `http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419?sc=internet/evalboard/product/252419.jsp` [Accessed March 19, 2016]. 23, 142

[60] Amir Moradi, Nima Mousavi, Christof Paar, and Mahmoud Salmasizadeh. A Comparative Study of Mutual Information Analysis under a Gaussian Assumption. In *WISA (Information Security Applications, 10th International Workshop)*, volume 5932 of *Lecture Notes in Computer Science*, pages 193–205. Springer, August 25-27 2009. Busan, Korea. 97, 107

[61] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. *IACR Cryptology ePrint Archive*, 2014:409, June 2 2014. 133

[62] Elke De Mulder, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Practical DPA attacks on MDPL. In *First IEEE International Workshop on Information Forensics and Security, WIFS 2009, London, UK, December 6-9, 2009*, pages 191–195. IEEE, 2009. 105

[63] NIST. AES Proposal: Rijndael (now FIPS PUB 197), April 2003. `http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf`. 143

[64] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962. 122

[65] Emmanuel Prouff, editor. *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, volume 8622 of *Lecture Notes in Computer Science*. Springer, 2014. 197, 199, 203

[66] Emmanuel Prouff and Matthieu Rivain. Theoretical and practical aspects of mutual information-based side channel analysis. *International Journal of Applied Cryptography (IJACT)*, 2(2):121–138, 2010. 103

[67] Emmanuel Prouff and Matthieu Rivain. Masking against Side-Channel Attacks: A Formal Security Proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013. 20, 28, 30

[68] Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012. 198, 201

[69] Chester Rebeiro and Debdeep Mukhopadhyay. Boosting Profiled Cache Timing Attacks With A Priori Analysis. *Information Forensics and Security, IEEE Transactions on*, 7(6):1900–1905, 2012. 122

[70] Chester Rebeiro, A. David Selvakumar, and A. S. L. Devi. Bitslice implementation of AES. In David Pointcheval, Yi Mu, and Kefei Chen, editors, *Cryptology and Network*

Security, 5th International Conference, CANS 2006, Suzhou, China, December 8-10, 2006, Proceedings, volume 4301 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2006. 106

[71] Mathieu Renauld, Dina Kamel, François-Xavier Standaert, and Denis Flandre. Information Theoretic and Security Analysis of a 65-Nanometer DDSLL AES S-Box. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *LNCS*, pages 223–239. Springer, 2011. 163

[72] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Generic DPA Attacks: Curse or Blessing? In Prouff [65], pages 98–111. 99, 103

[73] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. A note on the use of margins to compare distinguishers. In Prouff [65], pages 1–8. 72

[74] Matthieu Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In *Selected Areas in Cryptography*, volume 5381 of *LNCS*, pages 165–183. Springer, August 14-15 2008. Sackville, New Brunswick, Canada. 28, 30, 73

[75] Werner Schindler. A Timing Attack against RSA with the Chinese Remainder Theorem. In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2000. 121

[76] Werner Schindler. Optimized timing attacks against public key cryptosystems. *Statistics & Risk Modeling*, 20(1-4):191–210, 2002. DOI: 10.1524/strm.2002.20.14.191. 121

[77] Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, editor, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK. 74

[78] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948. 13

[79] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. 21, 178

[80] Peter W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994. 8

[81] François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA. 59

[82] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany. 9, 10, 21, 101, 128

[83] François-Xavier Standaert, Eric Peeters, Cédric Archambeau, and Jean-Jacques Quisquater. Towards security limits in side-channel attacks. In *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 30–45. Springer, October 10-13 2006. Yokohama, Japan. 27

[84] TELECOM ParisTech SEN research group. DPA Contest, 2008–2009. `http://www.DPAcontest.org/`. xiii, 11, 59, 60

[85] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual Information Analysis: How, When and Why? In *CHES*, volume 5747 of *LNCS*, pages 429–443. Springer, September 6-9 2009. Lausanne, Switzerland. 133

[86] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual Information Analysis: How, When and Why? In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 429–443. Springer, 2009. 97, 107

[87] Michael Weiß, Benedikt Heinz, and Frederic Stumpf. A cache timing attack on AES in virtualization environments. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers*, volume 7397 of *Lecture Notes in Computer Science*, pages 314–328. Springer, 2012. 122

[88] 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. 97, 103, 118

[89] Carolyn Whitnall and Elisabeth Oswald. A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. *J. Cryptographic Engineering*, 1(2):145–160, 2011. 72, 73, 88, 99

[90] Carolyn Whitnall, Elisabeth Oswald, and Luke Mather. An Exploration of the Kolmogorov-Smirnov Test as a Competitor to Mutual Information Analysis. In Emmanuel Prouff, editor, *CARDIS*, volume 7079 of *Lecture Notes in Computer Science*, pages 234–251. Springer, 2011. 70, 73, 76, 81, 97

[91] Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The Myth of Generic DPA . . . and the Magic of Learning. In Josh Benaloh, editor, *CT-RSA*, volume 8366 of *Lecture Notes in Computer Science*, pages 183–205. Springer, 2014. 99, 112

DÉPARTEMENT
Sciences et Technologies
de l'Information
et de la Communication

Titre: Vers une meilleure formalisation des attaques par canaux cachés.

Mots clés: Cryptographie, canaux cachés, Théorie de l'information, optimisation

Résumé: Dans le cadre de la sécurité des systèmes embarqués, il est nécessaire de connaître les attaques logicielles et physiques pouvant briser la sécurité de composants cryptographiques garantissant l'intégrité, la fiabilité et la confidentialité des données. Étant donné que les algorithmes utilisés aujourd'hui comme Advanced Encryption Standard (AES) sont considérés comme résistants contre la cryptanalyse linéaire et différentielle, d'autres méthodes plus insidieuses sont utilisés pour récupérer les secrets de ces composants. En effet, la clé secrète utilisée pour le chiffrement de données peut fuiter pendant l'algorithme. Il est ainsi possible de mesurer cette fuite et de l'exploiter. Cette technique est appelée attaque par canal auxiliaire.

Le principal objectif de ce manuscrit de thèse est de consolider les connaissances théoriques sur ce type de menace. Pour cela, nous appliquons des résultats de théorie de l'information à l'étude par canal auxiliaire. Nous montrons ainsi comment il est possible de comparer un modèle de fuite à un modèle de transmission de l'information.

Dans un premier temps, nous montrons que la sécurité d'un composant est fortement dépendante du rapport signal à bruit de la fuite. Ce résultat a un impact fort car il ne dépend pas de l'attaque choisie. Lorsqu'un designer équipe son produit, il ne connaît pas encore la manière dont son système embarqué pourra être attaqué plusieurs années plus tard. Les outils mathématiques proposés dans ce manuscrit pourront aider les concepteurs à estimer le niveau de fiabilité de leurs puces électroniques.

Title: Towards a Better Formalisation of the Side-Channel Threat.

Keywords: Cryptography, Side-Channel, Information Theory, Optimization

Abstract: In the field of the security of the embedded systems, it is necessary to know and understand the possible physical attacks that could break the security of cryptographic components. Since the current algorithms such as Advanced Encryption Standard (AES) are very resilient against differential and linear cryptanalysis, other methods are used to recover the secrets of these components. Indeed, the secret key used to encrypt data leaks during the computation of the algorithm, and it is possible to measure this leakage and exploit it. This technique to recover the secret key is called side-channel analysis.

The main target of this Ph. D. manuscript is to increase and consolidate the knowledge on the side-channel threat. To do so, we apply some information theoretic results to side-channel analysis. The main objective is show how a side-channel leaking model can be seen as a communication channel.

We first show that the security of a chip is dependant to the signal-to-noise ratio (SNR) of the leakage. This result is very useful since it is a generic result independent from the attack. When a designer builds a chip, he might not be able to know in advance how his embedded system will be attacked, maybe several years later. The tools that we provide in this manuscript will help designers to estimated the level of liability of their chips.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France