UNIVERSITY NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH

Supervisor:
Dr. James SMITH

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

September 13, 2017

Contents

List of Figures

List of Tables

List of Abbreviations

SCA	**S**ide **C**hannel **A**ttack

List of Symbols

$GF(2^b)$	Galois Field of order 2^b

Part I

Context and State of the Art

Chapter 1

Introduction

1.1 Introduction to Cryptography

The terms *Cryptography* (from the Greek *kryptòs* (secret) and *graphein* (writing)) and *Cryptanalysis*, denotes two branches of a science named *Cryptology*, or *science of the secret*. Cryptography initially refers to the art of *encrypting* messages, which means writing meaningful messages in such a way to appear nonsense to anyone unaware of the encryption process. In general, cryptography aims to construct protocols to secure communication, while cryptanalysis studies the resistance of cryptographic techniques, developing *attacks* to break the cryptosystems' security claims. These two complementary domains evolve in parallels, since the evolution of attack techniques allows conceiving more resistant cryptographic algorithms, and inversely the resistance of such algorithms requires the conception of more sophisticated attacks.

The art of cryptography is very ancient, probably as ancient as the language, but only the development of information technology made cryptology take the shape of a proper science, sometimes referred to as *Modern cryptology*. The last be seen as a branch of different disciplines, such as applied mathematics, computer science, electrical engineering, and communication science. Modern cryptosystems exploit algorithms based on mathematical tools and are implemented as computer programs, or electronic circuits. Their goal is to provide security functionality for communications that use *insecure channels*, for example the internet. In particular, modern cryptosystems are designed in order to ensure at least one of the four following information security properties:

a. *confidentiality*: the transmitted message must be readable only by a chosen pool of authorized entities;

b. *authenticity*: the receiver can verify the identity of the sender of a message;

c. *non-repudiation*: the sender of a message cannot deny having sent the message afterwards;

d. *data integrity*: the receiver can be convinced that the message has not been corrupted during the transmission.

Two branches of cryptography may be distinguished: the *symmetric cryptography* and the *asymmetric cryptography*. The first one historically appeared before and is based on the hypothesis that the two communicating entities share a common secret, or private key; for this reason this is also called *secret key cryptography*. The second one, introduced around 1970, allows any entity to encrypt a message in such a way that only a unique chosen other entity could decrypt it; this is also called *public key cryptography*.

A general principle in cryptography, nowadays widely accepted by cryptography researchers, is the one given by Kerckhoff in in 19th century: it states that cryptosystems should be secure even if everything about the system, except the key, is public knowledge. Following this principle, today many industrials and governmental agencies exploit for their security services cryptosystems based over standardized algorithms. Such algorithms are of public domain, thus have been tested and tried to be broken by a large amount of people, before, during and after the standardization process. Resistance to many attempts of attacks is actually the strengths of standard algorithms.

In the following part of this section a description of the two standard cryptographic primitives, *i.e.* building block algorithms used to build cryptographic protocols, that will be used in this thesis will; a symmetric one, the AES, and an asymmetric one, the RSA.

1.1.1 Description of AES

The *Advanced Encryption Standard* (AES) has been standardized in 2001 by the United States governmental agency *National Institute of Standards and Technology* (NIST) through the *Federal Information Processing Standards Publication 197* (FIPS PUB 197) [NIS]. It is a symmetric *block cipher*, *i.e.* an algorithm operating on fixed-length groups of bits.[1] The AES operates on blocks of 128 bits of plaintext, and can use keys of size 128, 192 or 256 bits. The encryption is done by rounds. The number of executed rounds depends on the key size (10 rounds for 128 bits, 12 for 192 and 14 pour 256). The basic unit for processing in the AES algorithm is a byte. For AES internal operations, bytes are arranged on a two-dimensional array of bytes called the *state*, denoted s. Such a state has 4 rows and 4 columns, thus contains 16 bytes. The byte lying at the i-th row, j-th column of s will be denoted by $s_{i,j}$ for $i, j \in \{0, 1, 2, 3\}$. The 16 input bytes and the 16 output bytes are indexed column-wise as shown in Fig. 1.1. Each element $s_{i,j}$ of a state is mathematically seen as an element of the *Rjindael finite field*, defined as $GF(2^8) = \mathbb{Z}/2\mathbb{Z}[X]/P(X)$ where $P(X) = X^8 + X^4 + X^3 + X + 1$. Five functions are performed during the AES, named KeySchedule, AddRoundKey, SubBytes, ShiftRow and MixColumn. At high level the AES algorithm is described hereafter:

Key Expansion: derivation of round keys from secret key through the KeySchedule function

Round 0:	AddRoundKey
Rounds 1 to penultimate:	SubBytes
	ShiftRow
	MixColumn
	AddRoundKey
Last Round:	SubBytes
	ShiftRow
	AddRoundKey

A description of the five functions is provided hereafter.

[1] in contrast with *stream ciphers*, which operate over a single plaintext bit at time

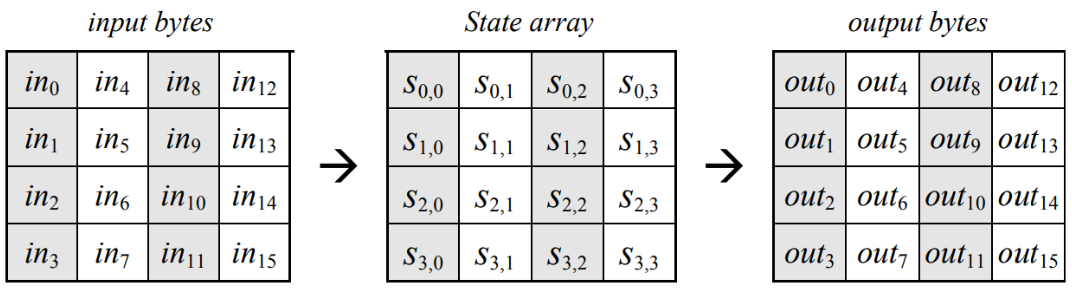

input bytes
State array
output bytes
in_0 in_4 in_8 in_{12}
in_1 in_5 in_9 in_{13}
in_2 in_6 in_{10} in_{14}
in_3 in_7 in_{11} in_{15}
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
out_0 out_4 out_8 out_{12}
out_1 out_5 out_9 out_{13}
out_2 out_6 out_{10} out_{14}
out_3 out_7 out_{11} out_{15}

FIGURE 1.1: State array input and output. Source: [NIS].

KeySchedule

The key round of the initial round of AES coincides with the secret encryption key $\boldsymbol{K} = (k_{0,0}, k_{0,1}, \dots, k_{0,3}, k_{1,0}, \dots, k_{1,3}, \dots, k_{3,3})$. The i-th round key is given by

$$\boldsymbol{K_i} = (k_{4i,0}, k_{4i,1}, \dots, k_{4i,3}, k_{4i+1,0}, \dots, k_{4i+1,3}, \dots, k_{4i+3,3}),$$

where, for $i > 3$

$$\begin{cases} k_{a,b} = k_{a-4,b} \oplus k_{a-1,b} & \text{if } a \not\equiv 0 \bmod 4 \\ k_{a,b} = k_{a-4,b} \oplus \mathrm{Sbox}(k_{a-1,(b+1)\bmod 4}) \oplus \mathrm{Rcon}(a) & \text{if } a \equiv 0 \bmod 4, \end{cases}$$

where $Rcon(a) = 2^{a-1}$ in the Rjindael finite field.[2]

AddRoundKey

SubByte

ShiftRow

MixColumn

1.1.2 Description of RSA

1.2 Secured Components

1.3 Smart Cards and Related Devices

As we have seen in the previous section, modern cryptography proposes solutions to secure communications that ask for electronic computations and repose their security over some secret keys. Keys are represented as long bit strings, impossible to be memorized by users. Thus, keys need to be stored in a secure medium, and never delivered in clear over insecure channels. Smart cards were historically conceived as a practical solution to such a key storage issue: they consist in small devices a user can easily carry around with, which not only store secret keys, but also are able to internally perform cryptographic operations, in such a way that keys are never asked to be delivered. The registrations of a first patent by Roland Moreno in 1974 and of a second one by Michel Ugon in 1977 are often referred to date the smart card invention, finally produced for the first time in 1979. Smart cards are pocket-sized

[2]where $2 = (00000010)_2$ is represented by the polynomial x

plastic-made cards equipped with a secured component, which is typically an integrated circuit containing a some computational units and some memories.

Today, about 40 years after its invention, they still have a huge diffusion, both in terms of applicative domains and in terms of quantity of exemplars. Indeed, they serve as credit or ATM cards, healthy cards, ID cards, public transport payment cards, fuel cards, identification and access badges, authorization cards for pay television, etc. Slightly changing the card support, we find other applications of the same kind of integrated circuits, for example the mobile phone SIMs (*Subscriber Identity Module* and the electronic passports. In terms of quantity, it seems that in 2014 8.8 billion smart cards have been sold, *i.e.* the same order of magnitude of the global population.

In addition to smart cards, the recent growing and variation of security needs lead to the development and specification of other kinds of secured components, for example the *Trusted Execution Environment* (TEE), which as a secured part of the main processor of a smartphone or tablet, and the *Trusted Platform Module* (TPM), which is a secure element providing cryptographic functionalities to a motherboard.

1.3.1 Embedded Cryptography Vulnerabilities

Side-Channel Attacks

Until the middle of nineties, the security of embedded cryptosystems was considered as equivalent to the mathematical security of the cryptographic algorithm. In classical cryptanalysis an attacker has usually the knowledge of the algorithm (in accordance to the Kerckhoff's principle) and of some inputs and/or outputs. Starting from this data, his goal is to retrieve the secret key. This attack model considers the algorithm computation as a black box, in the sense that no internal variable can be observed during execution, only inputs and/or outputs. With his seminal paper about Side-Channel Attacks (SCAs) in 1996, Paul Kocher showed that such a black-box model fails once the algorithm is implemented over a material component [Koc96]: an attacker can indeed inspect its component during the execution of the cryptographic algorithm, monitor some physical quantities, for example the execution time [Koc96] or the instantaneous power consumption [KJJ99] and deduct information about internal variables of the algorithm. Depending on the attacked algorithm, making inference over some well chosen internal variable (the so-called *sensitive variables* of the algorithm) is sufficient to retrieve the secret key. After these first works it was shown that other observable physical quantities contained *leakages* of sensitive information, for example the electromagnetic radiation emanating from the device [GMO01; QS01] and the acoustic emanations [GST14]. Moreover, if until few years ago it was thought that only small devices, equipped with slow microprocessors and in a small-size architecture, such as smart cards were vulnerable to this kind of Side-Channel Attacks, this recent work about acoustic emanations, together with other works exploiting electromagnetic fluctuations pointed out that much faster and bigger devices, *i.e.* laptops and desktop computers, are vulnerable as well [Gen+15; GPT15; Gen+16].

A Classification of the Attacks to Secured Components

The Side-Channel Attacks outlined in previous paragraph belong to a much bigger family of attacks that can be performed to break cryptographic devices security

TABLE 1.1: Classification of Attacks

	Passive	Active	
Hardware Attacks			Invasive
			Semi-Invisive
	SCAs	FAs	Non-Invasive
Software Attacks			

claims. First of all software attacks and hardware attacks must be distinguished. Software attacks only exploit vulnerability coming from the way the component's code is written. Hardware attacks are still commonly classified on the base of two criteria: on one hand we can distinguish passive and active attacks, on the other hand we can distinguish invasive, semi-invasive, non-invasive attacks.

Passive attacks: in passive attack, the device is let run respecting its specifications. The attacker observes its behaviour without provoking any alteration;

Active attacks: in active attacks a special manipulation is performed in order to make the normal behaviour of the device change.

Invasive attacks: in invasive attack, the device is depackaged and inspected at the level of the components technology. The circuit can be modified, broken, signals can be accessed via a probing station, etc. There is no limits to the manipulations attacker can do to the components;

Semi-invasive attacks: as in invasive attacks the device is depackaged, but in contrast to them, no direct electrical contact to the chip is done;

Non-invasive attacks: in non-invasive attacks the device is not modified and only accessible interfaces are exploited.

In literature, the term Side-Channel Attacks commonly denotes the passive non-invasive attacks. In the same way, active non-invasive attacks are often referred to as *Fault Injection Attacks*.

1.3.2 Certification of a Secure Hardware

parla anche degli attachi per perturbazione (tesi alexandre)

Chapter 2

Introduction to Side-Channel Attacks

2.1 Introduction to Side-Channel Attacks

2.1.1 Historical Overview

2.1.2 Terminology and Generalities

Target and Leakage Model

Points of Interest

Simple vs Advanced SCAs

Vertical vs Horizontal SCAs

Profiled vs Non-Profiled SCAs

Side-Channel Algebraic Attacks

Distinguishers

SCA Metrics

2.2 Main Side-Channel Countermeasures

2.2.1 Random Delays and Jitter

2.2.2 Shuffling

2.2.3 Masking

2.3 Higher-Order Attacks

2.3.1 Higher-Order Moments Analysis and Combining Functions

2.3.2 Profiling Higher-Order Attacks

Profiling with Masks Knowledge

Profiling without Masks Knowledge

Chapter 3

Introduction to Machine Learning

3.1 Basic Concepts of Machine Learning

3.1.1 The Task, the Experience and the Performance

3.1.2 Supervised, Semi-Supervised, Unsupervised Learning

3.1.3 Training, Validation and Test Sets

3.1.4 Underfitting, Overfitting and Regularization

3.1.5 Data Augmentation

3.1.6 No Free Lunch Theorem

3.2 Machine Learning Applications in Side-Channel Context

3.2.1 Profiled Attack as a Classification Problem

Support Vector Machine

Random Forest

Neural Networks

Part II

Contributions

Chapter 4

Introduction

4.1 Foreword of this Thesis: Research of Points of Interest

4.2 Dimensionality Reduction Approach

4.2.1 Linear Methods for First-Order Attacks

4.2.2 Kernel Methods for Higher-Order Attacks

4.3 Neural Network Approach

4.3.1 Toward Getting Rid of Information-Loosing Preprocessing

Chapter 5

Points of Interest

5.1 Motivations

5.1.1 The Curse of Dimensionality

5.2 Selection on Points of Interest: Classical Statistics

5.3 Related Issues: Leakage Detection and Leakage Assessment

test statistici agli ordini superiori, soprattutto per masking scheme hardware, CHES 2017 e precedenti

5.4 Generalized SNR for Multi-Variate Attacks

5.5 Observations Leading to Take a Dimensionality Reduction Approach

Chapter 6

Linear Dimensionality Reduction

6.1 Introduction

6.1.1 Principal Component Analysis

6.1.2 Linear Discriminant Analysis

6.1.3 Projection Pursuits

6.2 Principal Component Analysis

6.2.1 Statistical Point of View

6.2.2 Geometrical Point of View

6.3 Application of PCA in SCAs

6.3.1 Original vs Class-Oriented PCA

Remark. Stacked Auto-Encoders...

6.3.2 The Choice of the Principal Components

The introduction of the PCA method in SCA context (either in its classical or class-oriented version) has raised some important questions: *how many* principal components and *which ones* are sufficient/necessary to reduce the trace size (and thus the attack processing complexity) without losing important discriminative information?

Until now, an answer to the questions above has been given in [CK14], linked to the concept of *explained variance* (or *explained global variance*, EGV for short) of a PC $\boldsymbol{\alpha}_i$:

$$\mathrm{EGV}(\boldsymbol{\alpha}_i) = \frac{\lambda_i}{\sum_{k=1}^{r} \lambda_k} , \tag{6.1}$$

where r is the rank of the covariance matrix $\mathbf{S}$, and λ_j is the eigenvalue associated to the j-th PC $\boldsymbol{\alpha}_j$. $\mathrm{EGV}(\boldsymbol{\alpha}_i)$ is the variance of the data projected over the i-th PC (which equals λ_i) divided by the total variance of the original data (given by the trace of the covariance matrix $\mathbf{S}$, *i.e.* by the sum of all its non-zero eigenvalues). By definition of EGV, the sum of all the EGV values is equal to 1; that is why this quantity is often multiplied by 100 and expressed as percentage. Exploiting the EGV to choose among the PCs consists in fixing a wished *cumulative explained variance* β and in keeping C different PCs, where C is the minimum integer such that

$$\mathrm{EGV}(\boldsymbol{\alpha}_1) + \mathrm{EGV}(\boldsymbol{\alpha}_2) + \cdots + \mathrm{EGV}(\boldsymbol{\alpha}_C) \geq \beta . \tag{6.2}$$

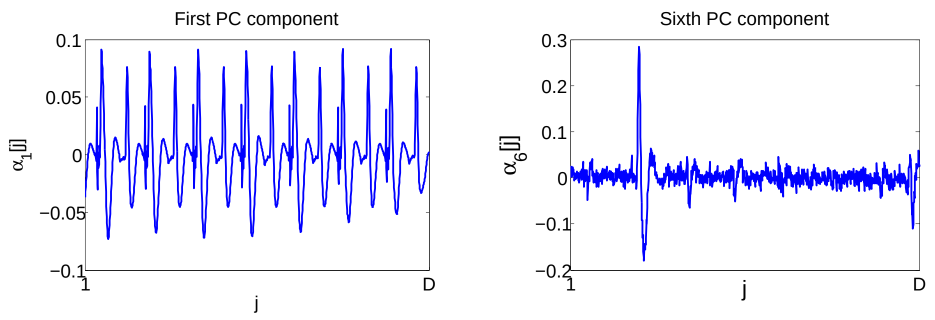


FIGURE 6.1: First and sixth PCs in DPA contest v4 trace set (between time samples 198001 and 199000)

However, if the adversary has a constraint for the reduced dimension C, the EGV notion simply suggests to keep the first C components, taking for granted that the optimal way to chose PCs is in their natural order. This assumption is not always confirmed in SCA context: in some works, researchers have already remarked that the first components sometimes contain more noise than information [BHW12; Spe+15] and it is worth discarding them. For the sake of providing a first example of this behaviour on publicly accessible traces, we applied a class-oriented PCA on 3000 traces from the DPA contest v4 [Par]; we focused over a small 1000-dimensional window in which, in complete knowledge about masks and other countermeasures, information about the first Sbox processing leaks (during the first round). In Fig. 6.1 the first and the sixth PCs are plotted. It may be noticed that the first component indicates that one can attend a high variance by exploiting the regularity of the traces, given by the clock signal, while the sixth one has high coefficients localised in a small time interval, very likely to signalize the instants in which the target sensitive variable leaks.

To the best of our knowledge, a single method adapted to SCA context has been proposed until now to automatically choose PCs [Mav+12] while dealing with the issue raised in Fig. 6.1. It is based on the following assumption:

Assumption 1. The leaking side-channel information is localised in few points of the acquired trace.

In the rest of the paper, we conduct our own analyses under Assumption 1 that we think to be reasonable in SCA contexts where the goal of the security developers is to minimize the number of leaking points. Under this assumption, the authors of [Mav+12] use for side-channel attack purposes the *Inverse Participation Ratio* (IPR), a measure widely exploited in Quantum Mechanics domain (see for example [**guhr1998random**]). They propose to use such a score to evaluate the eigenvectors *localization*. It is defined as follows:

$$\mathrm{IPR}(\boldsymbol{\alpha}_i) = \sum_{j=1}^{D} \boldsymbol{\alpha}_i[j]^4 \,. \tag{6.3}$$

The authors of [Mav+12] suggest to collect the PCs in decreasing order with respect to the IPR score.

The selection methods provided by the evaluation of the EGV and of the IPR are somehow complementary: the former is based only on the eigenvalues associated to the PCs and does not consider the form of the PCs themselves; the latter completely

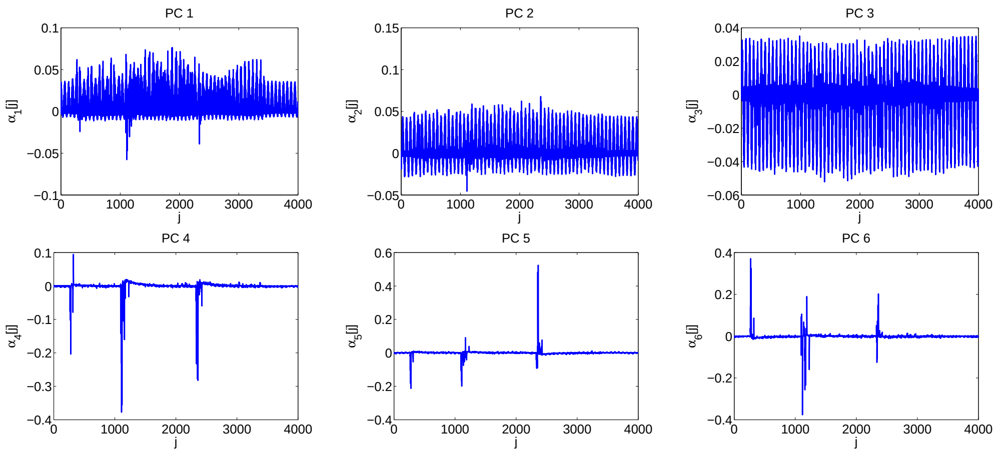


FIGURE 6.2: The first six PCs. Acquisition campaign on an 8-bit AVR Atmega328P (see Sec. 6.6).

discards the information given by the eigenvalues of the PCs, considering only the distribution of their coefficients. One of the contributions of the present paper is to propose a new selection method, that builds a bridge between the EGV and the IPR approaches. As we will argue, our method, based on the so-called *explained local variance*, does not only lead to the construction of a new selection criterion, but also permits to modify the PCs, choosing individually the coefficients to keep and those to discard.

6.3.3 The Explained Local Variance Selection Method

Riprendere le notazioni e mettere apposto i newcommand

Definition 1. The *Explained Local Variance* of a PC $\boldsymbol{\alpha}_i$ in a sample j, is defined by

$$\mathrm{ELV}(\boldsymbol{\alpha}_i, j) = \frac{\lambda_i \boldsymbol{\alpha}_i[j]^2}{\sum_{k=1}^{r} \lambda_k} = \mathrm{EGV}(\boldsymbol{\alpha}_i)\boldsymbol{\alpha}_i[j]^2 \,. \tag{6.4}$$

Let $\mathcal{J} = \{j_1^i, j_2^i, \ldots, j_D^i\} \subset \{1, 2, \ldots, D\}$ be a set of indexes sorted such that $\mathrm{ELV}(\boldsymbol{\alpha}_i, j_1^i) \geq \mathrm{ELV}(\boldsymbol{\alpha}_i, j_2^i) \geq \cdots \geq \mathrm{ELV}(\boldsymbol{\alpha}_i, j_D^i)$. It may be observed that the sum over all the $\mathrm{ELV}(\boldsymbol{\alpha}_i, j)$, for $j \in [1, \ldots, D]$, equals $\mathrm{EGV}(\boldsymbol{\alpha}_i)$. If we operate such a sum in a cumulative way following the order provided by the sorted set $\mathcal{J}$, we obtain a complete description of the trend followed by the component $\boldsymbol{\alpha}_i$ to achieve its EGV. As we can see in Fig. **??**, where such cumulative ELVs are represented, the first 3 components are much slower in achieving their final EGV, while the 4th, the 5th and the 6th achieve a large part of their final EGVs very quickly (*i.e.* by adding the ELV contributions of much less time samples). For instance, for $i = 4$, the sum of the $\mathrm{ELV}(\boldsymbol{\alpha}_4, j_k^4)$, with $k \in [1, \ldots, 30]$, almost equals $\mathrm{EGV}(\boldsymbol{\alpha}_4)$, whereas the same sum for $i = 1$ only achieves about the 15% of $\mathrm{EGV}(\boldsymbol{\alpha}_1)$. Actually, the EGV of the 4th, the 5th and the 6th component only essentially depends on a very few time samples. This observation, combined with Assumption 1, suggests that they are more suitable for SCA than the three first ones. To validate this statement, it suffices to look at the form of such components (Fig. 6.2): the leading ones are very influenced by the clock, while the latest ones are well localised over the leaking points.

Operating a selection of components *via* ELV, in analogy with the EGV, requires to fix the reduced space dimension C, or a threshold β for the cumulative ELV. In the first case, the maximal ELVs of each PC are compared, and the C components achieving the highest values of such ELVs are chosen. In the second case, all pairs (PC, time sample) are sorted in decreasing order with respect to their ELV, and summed until the threshold β is achieved. Then only PCs contributing in this sum are selected.

We remark that the ELV is a score associated not only to the whole components, but to each of their coefficients. This interesting property can be exploited to further remove, within a selected PC, the non-significant points, *i.e.* those with a low ELV. In practice this is done by setting these points to zero. That is a natural way to exploit the ELV score in order to operate a kind of *denoising* for the reduced data, making them only depend on the significant time samples. In Sec. 6.6 (scenario 4) we test the performances of an attack varying the number of time samples involved in the computation of the reduced data, and showing that such a denoising processing might impact significantly.

6.4 Linear Discriminant Analysis

6.4.1 Statistical Point of View

6.4.2 Geometrical Point of View

6.5 Application of LDA in SCAs

6.5.1 The Small Sample Size problem

6.6 Experimental Results

In this section we compare the different extractors provided by the PCA and the LDA in association with the different techniques of components selection. Defining an universal criterion to compare the different extractors would not make sense since the latter one should encompass a lot of parameters, sometimes opposite, that vary according to the context (amount of noise, specificity of the information leakage, nature of the side channel, etc.). For this reason we choose to split our comparisons into four scenarios. Each scenario has a single varying parameter that, depending on the attacker context, may wish to be minimized. Hereafter the definition of the four scenario. In the following only results of the two first is reported, the interested reader might refer to AppendixA for results of in the two other scenarios.

Scenario 1 varying parameter: number of attack traces N_a,

Scenario 2 varying parameter: number of profiling traces N_p,

Scenario 3 varying parameter: number of projecting components selected C,

Scenario 4 varying parameter: number of original time samples implied into the trace preprocessing computation $\sharp$PoI .

For scenarios in which
$nbProfilingTraces$ is fixed, the value of N_p is chosen high enough to avoid the SSS problem, and the extensions of LDA presented in Sec. **??** are not evaluated. This choice of N_p will imply that the LDA is always performed in a favourable situation, which makes expect the LDA to be particularly efficient for these experiments. Consequentely, for the scenarios in which N_p is high, our goal is to study whether the

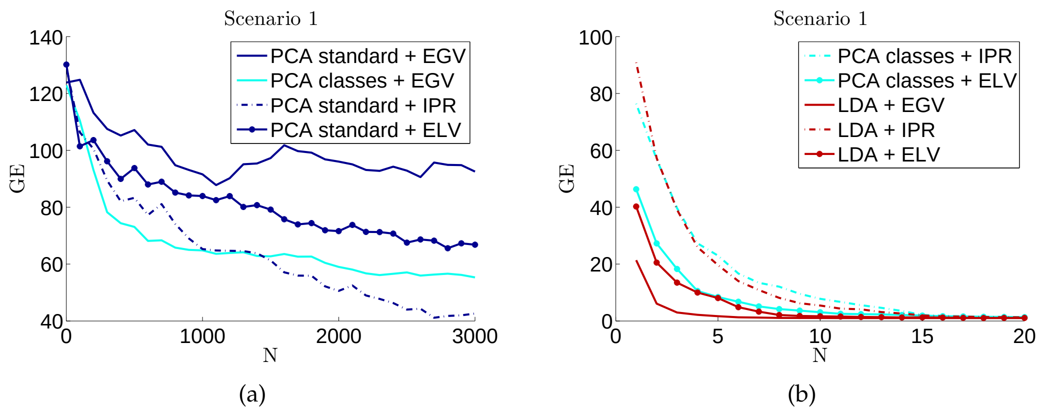


FIGURE 6.3: Guessing Entropy as function of the number of attack traces for different extraction methods. All Guessing Entropies are estimated as the average rank of the right key over 100 independent experiments.

PCA can be made almost as efficient as the LDA thanks to the component selection methods discussed in Sec. 6.3.2.

This part will maybe be useless: somewhere I will have described all trace sets

The testing adversary.

Scenario 1.

cos'e N_z To analyse the dependence between the extraction methods presented in Sections **??** and **??** and the number of attack traces N_a needed to achieve a given GE, we fixed the other parameters as follows: $N_z = 50$ $(N_p = 50 \times 256)$, $C = 3$ and $\sharp\text{PoI} = 3996$ (all points are allowed to participate in the building of the PCs and of the LDCs). The experimental results, depicted in Fig. 6.3(a)-(b), show that the PCA standard method has very bad performances in SCA, while the LDA outperforms the others. Concerning the class-oriented PCA, we observe that its performance is close to that of LDA when combined with the selection methods ELV (which performs best) or IPR.

Scenario 2.

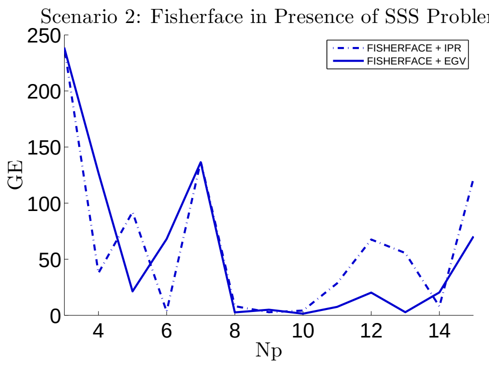
Now we test the behaviour of the extraction methods as function of the number N_z of available profiling traces per class. The number of components C is still fixed to 3, $\sharp\text{PoI} = 3996$ again and the number of attack traces is $N_a = 100$. This scenario has to be divided into two parts: if $N_z \leq 15$, then $N_p < D$ and the SSS problem occurs. Thus, in this case we test the four extensions of LDA presented in Sec. **??**, associated to either the standard selection, to which we abusively refer as EGV,[1] or to the IPR selection. We compare them to the class-oriented PCA associated to EGV, IPR or ELV. The ELV selection is not performed for the techniques extending LDA, since for some of them the projecting LDCs are not associated to some eigenvalues in a meaningful way. On the contrary, if $N_z \geq 16$ there is no need to approximate the LDA technique, so the classical one is performed. Results for this scenario are shown in Fig. 6.4. It may be noticed that the combinations class-oriented PCA + ELV/IPR select exactly the same components, for our data, see Fig. 6.4(e) and do

[1]It consists in keeping the C first LDCs (the C last for the Direct LDA)

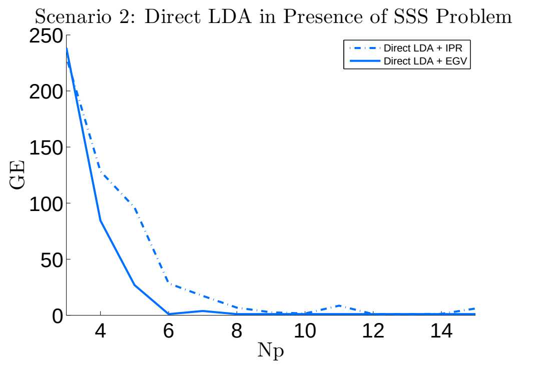
not suffer from the lack of profiling traces. They are slightly outperformed by the $\mathbf{S_W}$ Null Space method associated with the EGV, see Fig.6.4(d). The Direct LDA (Fig. 6.4(b)) method also provides a good alternative, while the other tested methods do not show a stable behaviour. The results in absence of the SSS problem (Fig.6.4(f)) confirm that the standard PCA is not adapted to SCA, even when provided with more profiling traces. It also shows that among class-oriented PCA and LDA, the class-oriented PCA converges faster.

6.7 Misaligning Effects

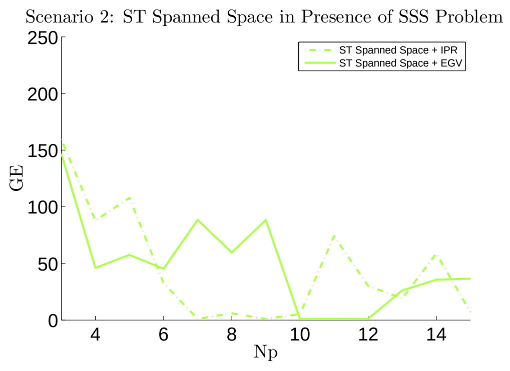
give parameters: 6 4 citare Choudary, Template Attacks over different devices In this section we experimentally show how the approach based on linear dimensionality reduction described in this chapter is affected by traces misalignment. To this aim, we simply take the same data and parameters exploited for Scenario 1 in Sec. 6.6, and artificially misalign them through the technique proposed in Appendix A.2 with parameters **parameters here**. Then we tried to pre-process attack them through the 9 methodology tested in Scenario 1. It may be noticed in Fig. 6.5 that none of the 9 techniques is still efficient, included the optimal LDA+EGV that lead to null guessing entropy the synchronized traces using less 7 attack traces. In this case it cannot lead to successful attack in less than 3000 traces.



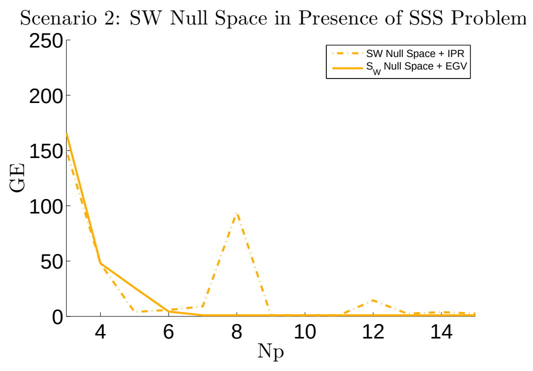
(a)



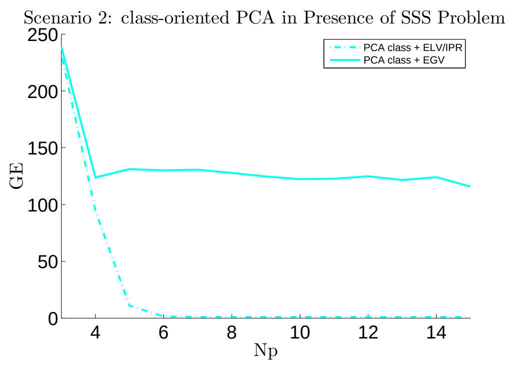
(b)



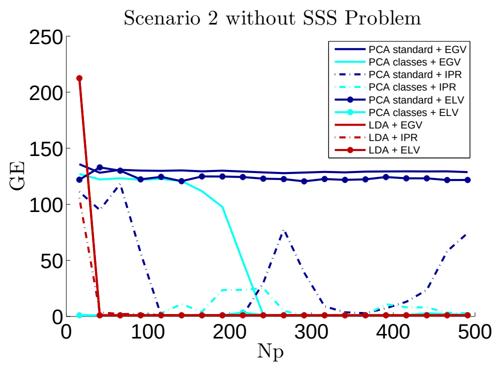
(c)



(d)



(e)



(f)

FIGURE 6.4: Guessing Entropy as function of the number of profiling traces. Figures (a)-(d): methods extending the LDA in presence of SSS problem; Figure (e): class-oriented PCA in presence of the SSS problem; Figure (f): number of profiling traces high enough to avoid the SSS problem.

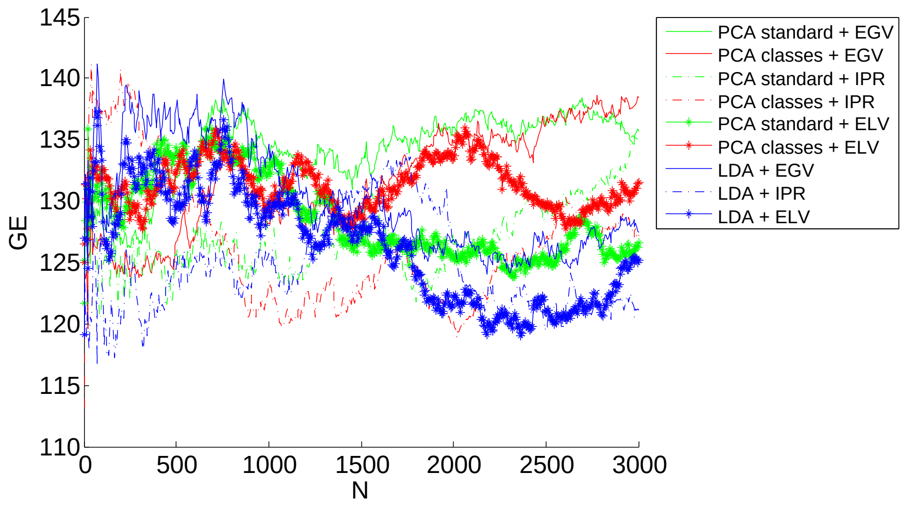


FIGURE 6.5: Degradation of linear-reduction-based template attacks in presence of misalignment.

Chapter 7

Kernel Dimensionality Reduction

7.1 Motivation

7.1.1 Higher-Order Attacks

Higher-Order Version of Projection Pursuits

7.2 Kernel Function and Kernel Trick

7.2.1 Local Kernel Functions as Similarity Metrics

7.3 Kernel Discriminant Analysis

7.4 Experiments over Atmega328P

7.4.1 The Regularization Problem

7.4.2 The Multi-Class Trade-Off

7.4.3 Multi-Class vs 2-class Approach

7.4.4 Asymmetric Preprocessing/Attack Approach

Comparison with Projection Pursuits

7.5 Drawbacks of Kernel Methods

Misalignment Effects

Memory Complexity and Actual Number of Parameters

Two-Phases Approach: Preprocessing-Templates

Chapter 8

Convolutional Neural Networks against Jitter-Based Countermeasures

8.1 Moving from Kernel Machines to Neural Networks

8.2 Misalignment of Side-Channel Traces

8.2.1 The Necessity and the Risks of Applying Realignment Techniques

8.2.2 Analogy with Image Recognition Issues

8.3 Convolutional Layers to Impose Shift-Invariance

8.4 Data Augmentation for Misaligned Side-Channel Traces

8.5 Experiments against Software Countermeasures

8.6 Experiments against Artificial Hardware Countermeasures

8.7 Experiments against Real-Case Hardware Countermeasures

Chapter 9

KDA vs Neural Networks Approach for HO-Attacks

9.1 Simulated Experiment for Profiled HO-Attacks

9.1.1 The Simulations

9.1.2 Comparison between KDA and MLP

9.2 Real-Case Experiments over ARM Cortex-M4

Chapter 10

Siamese Neural Networks for Collision Attacks

10.1 Introduction

10.2 Siamese Neural Networks

10.2.1 Distances and Loss Functions

10.2.2 Relation with Kernel Machines

10.3 Collision Attacks with Siamese NNs

10.3.1 Experimental Results

Chapter 11

Conclusions and Perspectives

11.1 Summary

11.2 Strengthen Embedded Security: the Main Challenge for Machine Learning Applications

Appendix A

Scenario 3 and 4 of CARDIS '15 paper

A.1 Scenario 3.

Let C be now variable and let the other parameters be fixed as follows: $N_a = 100, N_z = 200, \sharp\text{PoI} = 3996$. Looking at Fig. **??**, we might observe that the standard PCA might actually well perform in SCA context if provided with a larger number of kept components; on the contrary, a little number of components suffices to the LDA. Finally, keeping more of the necessary components does not worsen the efficiency of the attack, which allows the attacker to choose C as the maximum value supported by his computational means.

Remark. In our experiments the ELV selection method only slightly outperforms the IPR. Nevertheless, since it relies on more sound and more general observations, *i.e.* the maximization of explained variance concentrated into few points, it is likely to be more robust and less case-specific. For example, in Fig. 6.4(f) it can be remarked that while the class-oriented PCA + ELV line keeps constant on the value 0 of guessing entropy, the class-oriented PCA + IPR is sometimes higher than 0.

Is the table with results overview interesting?

A.2 Scenario 4.

This is the single scenario in which we allow the ELV selection method to not only select the components to keep but also to modify them, keeping only some coefficients within each component, setting the other ones to zero. We select pairs *(component, time sample)* in decreasing order of the ELV values, allowing the presence of only $C = 3$ components and $\sharp$PoI different times samples: *i.e.*, we impose that the matrix A defining the extractor (see (**??**)) has $C = 3$ rows (storing the 3 chosen components, transposed) and exactly $\sharp$PoI non-zero columns. Looking at Fig. **??** we might observe that the LDA allows to achieve the maximal guessing entropy with only 1 PoI in each of the 3 selected components. Actually, adding PoIs worsen its performances, which is coherent with the assumption that the vulnerable information leaks in only a few points. Such points are excellently detected by the LDA. Adding contribution from other points raises the noise, which is then compensated by the contributions of further noisy points, in a very delicate balance. Such a behaviour is clearly visible in standard PCA case: the first 10 points considered raise the level of noise, that is then balanced by the last 1000 points.

Artificially Simulate Jitter

Bibliography

[BHW12] Lejla Batina, Jip Hogenboom, and Jasper G.J. van Woudenberg. "Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis". English. In: *Topics in Cryptology CT-RSA 2012*. Ed. by Orr Dunkelman. Vol. 7178. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 383–397. ISBN: 978-3-642-27953-9. DOI: 10.1007/978-3-642-27954-6_24. URL: http://dx.doi.org/10.1007/978-3-642-27954-6_24.

[CK14] Omar Choudary and Markus G Kuhn. "Efficient template attacks". In: *Smart Card Research and Advanced Applications*. Springer, 2014, pp. 253–270.

[Gen+15] Daniel Genkin et al. "Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2015, pp. 207–228.

[Gen+16] Daniel Genkin et al. "ECDH key-extraction via low-bandwidth electromagnetic attacks on PCs". In: *Cryptographers' Track at the RSA Conference*. Springer. 2016, pp. 219–235.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. "Electromagnetic analysis: Concrete results". In: *Cryptographic Hardware and Embedded Systems - CHES 2001*. Springer. 2001, pp. 251–261.

[GPT15] Daniel Genkin, Itamar Pipman, and Eran Tromer. "Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs". In: *Journal of Cryptographic Engineering* 5.2 (2015), pp. 95–112.

[GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. "RSA key extraction via low-bandwidth acoustic cryptanalysis". In: *International Cryptology Conference*. Springer. 2014, pp. 444–461.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential power analysis". In: *Annual International Cryptology Conference*. Springer. 1999, pp. 388–397.

[Koc96] Paul C Kocher. "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems". In: *Annual International Cryptology Conference*. Springer. 1996, pp. 104–113.

[Mav+12] Dimitrios Mavroeidis et al. "PCA, Eigenvector Localization and Clustering for Side-Channel Attacks on Cryptographic Hardware Devices". English. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by PeterA. Flach, Tijl De Bie, and Nello Cristianini. Vol. 7523. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 253–268. ISBN: 978-3-642-33459-7. DOI: 10.1007/978-3-642-33460-3_22. URL: http://dx.doi.org/10.1007/978-3-642-33460-3_22.

[NIS] FIPS PUB NIST. *197," Advanced Encryption Standard (AES)," November 2001*.

[Par] TELECOM ParisTech. "DPA Contest 4". In: (). http://www.DPAcontest.org/v4/. URL: http://www.DPAcontest.org/v4/.

[QS01] Jean-Jacques Quisquater and David Samyde. "Electromagnetic analysis (ema): Measures and counter-measures for smart cards". In: *Smart Card Programming and Security* (2001), pp. 200–210.

[Spe+15] Robert Specht et al. "Improving Non-Profiled Attacks on Exponentiations Based on Clustering and Extracting Leakage from Multi-Channel High-Resolution EM Measurements". In: *Sixth International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2015)*. 2015.