

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

August 30, 2017

Contents

I	Context and State of the Art	1
1	Introduction	3
1.1	Introduction to Cryptography	3
1.1.1	Secret-Key Cryptography	3
1.1.2	Public-Key Cryptography	3
1.2	Secure Hardware and Embedded Cryptography	3
1.2.1	The Example of the Smart Card	3
1.2.2	Certification of a Secure Hardware	3
1.2.3	Modern More Complex Devices to Certify	3
1.2.4	Embedded Cryptography Vulnerabilities	3
2	Introduction to Side-Channel Attacks	5
2.1	Introduction to Side-Channel Attacks	6
2.1.1	Historical Overview	6
2.1.2	Terminology and Generalities	6
	Target and Leakage Model	6
	Points of Interest	6
	Simple vs Advanced SCAs	6
	Vertical vs Horizontal SCAs	6
	Profiled vs Non-Profiled SCAs	6
	Side-Channel Algebraic Attacks	6
	Distinguishers	6
	SCA Metrics	6
2.2	Main Side-Channel Countermeasures	6
2.2.1	Random Delays and Jitter	6
2.2.2	Shuffling	6
2.2.3	Masking	6
2.3	Higher-Order Attacks	6
2.3.1	Higher-Order Moments Analysis and Combining Functions	6
2.3.2	Profiling Higher-Order Attacks	6
	Profiling with Masks Knowledge	6
	Profiling without Masks Knowledge	6
2.4	Thesis Contribution and Organization	6
2.4.1	Foreword of this Thesis: Research of Points of Interest	6
2.4.2	Dimensionality Reduction Approach	6
	Linear Methods for First-Order Attacks	6
	Kernel Methods for Higher-Order Attacks	6
2.4.3	Neural Network Approach	6
	Toward Getting Rid of Information-Losing Preprocessing	6

3	Introduction to Machine Learning	7
3.1	Basic Concepts of Machine Learning	7
3.1.1	The Task, the Experience and the Performance	7
3.1.2	Supervised, Semi-Supervised, Unsupervised Learning	7
3.1.3	Training, Validation and Test Sets	7
3.1.4	Underfitting, Overfitting and Regularization	7
3.1.5	Data Augmentation	7
3.1.6	No Free Lunch Theorem	7
3.2	Machine Learning Applications in Side-Channel Context	7
3.2.1	Profiled Attack as a Classification Problem	7
	Support Vector Machine	7
	Random Forest	7
	Neural Networks	7
II	Contributions	9
4	Points of Interest	11
4.1	Motivations	11
4.1.1	The Curse of Dimensionality	11
4.2	Selection on Points of Interest: Classical Statistics	11
4.3	Related Issues: Leakage Detection and Leakage Assessment	11
4.4	Generalized SNR for Multi-Variate Attacks	11
4.5	Observations Leading to Take a Dimensionality Reduction Approach	11
5	Linear Dimensionality Reduction	13
5.1	Introduction	13
5.1.1	Principal Component Analysis	13
5.1.2	Linear Discriminant Analysis	13
5.1.3	Projection Pursuits	13
5.2	Principal Component Analysis	13
5.2.1	Statistical Point of View	13
5.2.2	Geometrical Point of View	13
5.3	Application of PCA in SCAs	13
5.3.1	Original vs Class-Oriented PCA	13
5.3.2	The Choice of the Principal Components	13
5.3.3	The Explained Local Variance Selection Method	15
5.4	Linear Discriminant Analysis	16
5.4.1	Statistical Point of View	16
5.4.2	Geometrical Point of View	16
5.5	Application of LDA in SCAs	16
5.5.1	The Small Sample Size problem	16
5.6	Experimental Results	16
	The testing adversary.	17
	Scenario 1.	17
	Scenario 2.	17
5.7	Misaligning Effects	18

6	Kernel Dimensionality Reduction	21
6.1	Motivation	21
6.1.1	Higher-Order Attacks	21
	Higher-Order Version of Projection Pursuits	21
6.2	Kernel Function and Kernel Trick	21
6.2.1	Local Kernel Functions as Similarity Metrics	21
6.3	Kernel Discriminant Analysis	21
6.4	Experiments over Atmega328P	21
6.4.1	The Regularization Problem	21
6.4.2	The Multi-Class Trade-Off	21
6.4.3	Multi-Class vs 2-class Approach	21
6.4.4	Asymmetric Preprocessing/Attack Approach	21
	Comparison with Projection Pursuits	21
6.5	Drawbacks of Kernel Methods	21
	Misalignment Effects	21
	Memory Complexity and Actual Number of Parameters	21
	Two-Phases Approach: Preprocessing-Templates	21
7	Convolutional Neural Networks against Jitter-Based Countermeasures	23
7.1	Moving from Kernel Machines to Neural Networks	23
7.2	Misalignment of Side-Channel Traces	23
7.2.1	The Necessity and the Risks of Applying Realignment Techniques	23
7.2.2	Analogy with Image Recognition Issues	23
7.3	Convolutional Layers to Impose Shift-Invariance	23
7.4	Data Augmentation for Misaligned Side-Channel Traces	23
7.5	Experiments against Software Countermeasures	23
7.6	Experiments against Artificial Hardware Countermeasures	23
7.7	Experiments against Real-Case Hardware Countermeasures	23
8	KDA vs Neural Networks Approach for HO-Attacks	25
8.1	Simulated Experiment for Profiled HO-Attacks	25
8.1.1	The Simulations	25
8.1.2	Comparison between KDA and MLP	25
8.2	Real-Case Experiments over ARM Cortex-M4	25
9	Siamese Neural Networks for Collision Attacks	27
9.1	Introduction	27
9.2	Siamese Neural Networks	27
9.2.1	Distances and Loss Functions	27
9.2.2	Relation with Kernel Machines	27
9.3	Collision Attacks with Siamese NNs	27
9.3.1	Experimental Results	27
10	Conclusions and Perspectives	29
10.1	Summary	29
10.2	Strengthen Embedded Security: the Main Challenge for Machine Learning Applications	29
	Scenario 3.	31
	Scenario 4.	31

List of Figures

5.1	First and sixth PCs in DPA contest v4 trace set (between time samples 198001 and 199000)	14
5.2	The first six PCs. Acquisition campaign on an 8-bit AVR Atmega328P (see Sec. 5.6).	15
5.3	Guessing Entropy as function of the number of attack traces	17
5.4	Guessing Entropy as function of the number of profiling traces.	19
5.5	Degradation of linear-reduction-based template attacks in presence of misalignment.	20

List of Tables

List of Abbreviations

SCA Side Channel Attack

List of Symbols

Part I

Context and State of the Art

Chapter 1

Introduction

1.1 Introduction to Cryptography

1.1.1 Secret-Key Cryptography

1.1.2 Public-Key Cryptography

1.2 Secure Hardware and Embedded Cryptography

1.2.1 The Example of the Smart Card

1.2.2 Certification of a Secure Hardware

1.2.3 Modern More Complex Devices to Certify

1.2.4 Embedded Cryptography Vulnerabilities

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

2.4 Thesis Contribution and Organization

2.4.1 Foreword of this Thesis: Research of Points of Interest

2.4.2 Dimensionality Reduction Approach

Linear Methods for First-Order Attacks

Kernel Methods for Higher-Order Attacks

2.4.3 Neural Network Approach

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

Points of Interest

4.1 Motivations

4.1.1 The Curse of Dimensionality

4.2 Selection on Points of Interest: Classical Statistics

4.3 Related Issues: Leakage Detection and Leakage Assessment

4.4 Generalized SNR for Multi-Variate Attacks

4.5 Observations Leading to Take a Dimensionality Reduction Approach

Chapter 5

Linear Dimensionality Reduction

5.1 Introduction

5.1.1 Principal Component Analysis

5.1.2 Linear Discriminant Analysis

5.1.3 Projection Pursuits

5.2 Principal Component Analysis

5.2.1 Statistical Point of View

5.2.2 Geometrical Point of View

5.3 Application of PCA in SCAs

5.3.1 Original vs Class-Oriented PCA

Remark. Stacked Auto-Encoders...

5.3.2 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 α_i :

$$\text{EGV}(\alpha_i) = \frac{\lambda_i}{\sum_{k=1}^r \lambda_k}, \quad (5.1)$$

where r is the rank of the covariance matrix \mathbf{S} , and λ_j is the eigenvalue associated to the j -th PC α_j . $\text{EGV}(\alpha_i)$ 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

$$\text{EGV}(\alpha_1) + \text{EGV}(\alpha_2) + \cdots + \text{EGV}(\alpha_C) \geq \beta. \quad (5.2)$$

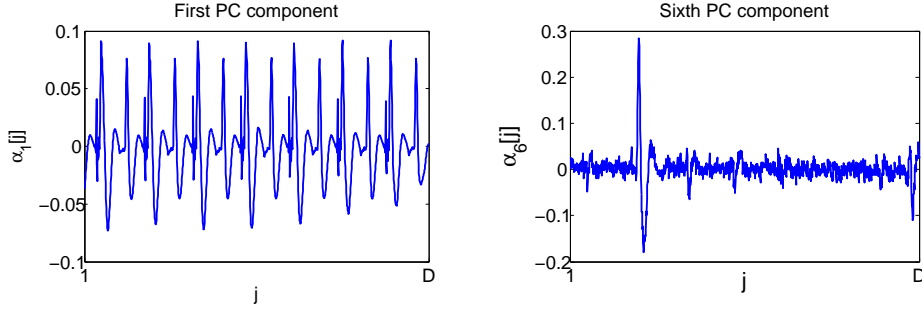


FIGURE 5.1: First and sixth PCs in DPA contest v4 trace set (between time samples 198001 and 199000)

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 choose 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. 5.1 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. 5.1. 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:

$$\text{IPR}(\alpha_i) = \sum_{j=1}^D \alpha_i[j]^4. \quad (5.3)$$

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

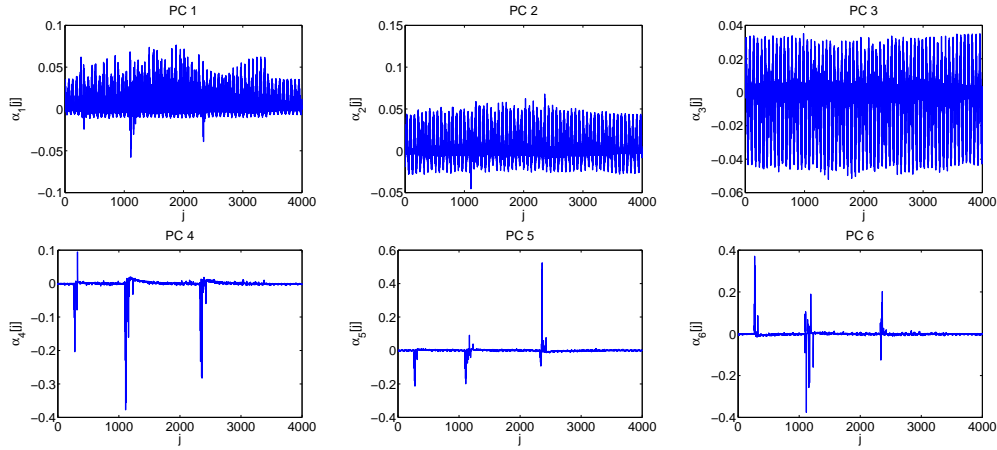


FIGURE 5.2: The first six PCs. Acquisition campaign on an 8-bit AVR Atmega328P (see Sec. 5.6).

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.

5.3.3 The Explained Local Variance Selection Method

Riprendere le notazioni e mettere apposto i newcommand

Definition 1. The *Explained Local Variance* of a PC α_i in a sample j , is defined by

$$\text{ELV}(\alpha_i, j) = \frac{\lambda_i \alpha_i[j]^2}{\sum_{k=1}^r \lambda_k} = \text{EGV}(\alpha_i) \alpha_i[j]^2. \quad (5.4)$$

Let $\mathcal{J} = \{j_1^i, j_2^i, \dots, j_D^i\} \subset \{1, 2, \dots, D\}$ be a set of indexes sorted such that $\text{ELV}(\alpha_i, j_1^i) \geq \text{ELV}(\alpha_i, j_2^i) \geq \dots \geq \text{ELV}(\alpha_i, j_D^i)$. It may be observed that the sum over all the $\text{ELV}(\alpha_i, j)$, for $j \in [1, \dots, D]$, equals $\text{EGV}(\alpha_i)$. 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 α_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 $\text{ELV}(\alpha_4, j_k^4)$, with $k \in [1, \dots, 30]$, almost equals $\text{EGV}(\alpha_4)$, whereas the same sum for $i = 1$ only achieves about the 15% of $\text{EGV}(\alpha_1)$. 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. 5.2): 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. 5.6 (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.

5.4 Linear Discriminant Analysis

5.4.1 Statistical Point of View

5.4.2 Geometrical Point of View

5.5 Application of LDA in SCAs

5.5.1 The Small Sample Size problem

5.6 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 Appendix 10.2 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 $\#PoI$.

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. Consequently, for the scenarios in which N_p is high, our goal is to study whether the

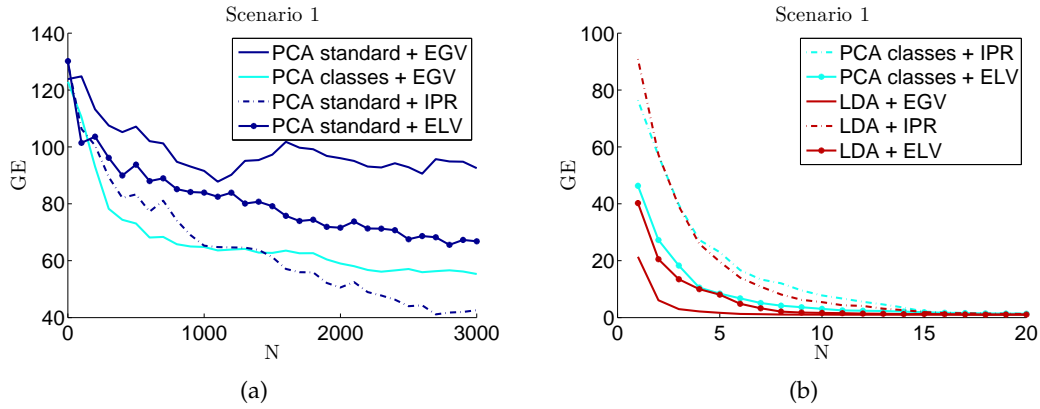


FIGURE 5.3: Guessing Entropy as function of the number of attack traces for different extraction methods. All Guessing Entropies are estimated as the average rank of the right key over 100 independent experiments.

PCA can be made almost as efficient as the LDA thanks to the component selection methods discussed in Sec. 5.3.2.

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. 5.3(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. 5.4. It may be noticed that the combinations class-oriented PCA + ELV/IPR select exactly the same components, for our data, see Fig. 5.4(e) and do

¹It 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.5.4(d). The Direct LDA (Fig. 5.4(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.5.4(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.

5.7 Misaligning Effects

give parameters: 6 4 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. 5.6, and artificially misalign them through the technique proposed in Appendix 10.2 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. 5.5 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.



FIGURE 5.4: 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.

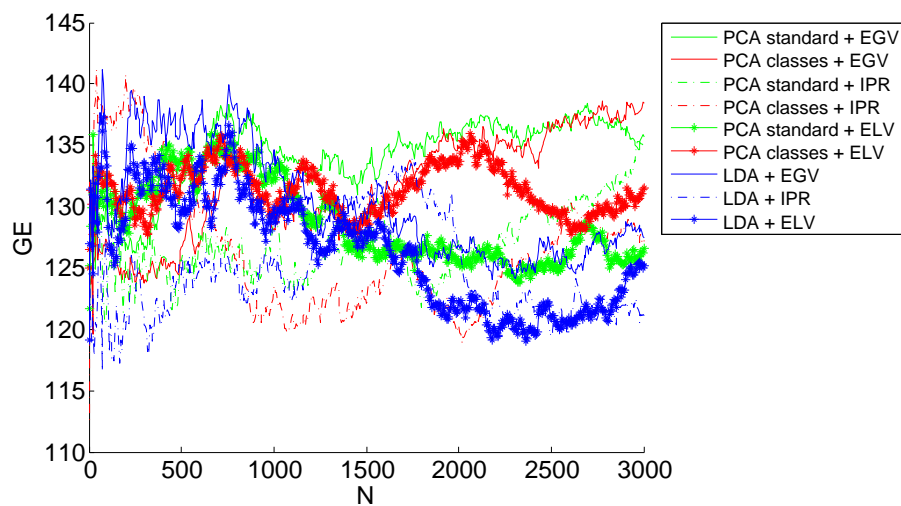


FIGURE 5.5: Degradation of linear-reduction-based template attacks in presence of misalignment.

Chapter 6

Kernel Dimensionality Reduction

6.1 Motivation

6.1.1 Higher-Order Attacks

Higher-Order Version of Projection Pursuits

6.2 Kernel Function and Kernel Trick

6.2.1 Local Kernel Functions as Similarity Metrics

6.3 Kernel Discriminant Analysis

6.4 Experiments over Atmega328P

6.4.1 The Regularization Problem

6.4.2 The Multi-Class Trade-Off

6.4.3 Multi-Class vs 2-class Approach

6.4.4 Asymmetric Preprocessing/Attack Approach

Comparison with Projection Pursuits

6.5 Drawbacks of Kernel Methods

Misalignment Effects

Memory Complexity and Actual Number of Parameters

Two-Phases Approach: Preprocessing-Templates

Chapter 7

Convolutional Neural Networks against Jitter-Based Countermeasures

7.1 Moving from Kernel Machines to Neural Networks

7.2 Misalignment of Side-Channel Traces

7.2.1 The Necessity and the Risks of Applying Realignment Techniques

7.2.2 Analogy with Image Recognition Issues

7.3 Convolutional Layers to Impose Shift-Invariance

7.4 Data Augmentation for Misaligned Side-Channel Traces

7.5 Experiments against Software Countermeasures

7.6 Experiments against Artificial Hardware Countermeasures

7.7 Experiments against Real-Case Hardware Countermeasures

Chapter 8

KDA vs Neural Networks Approach for HO-Attacks

8.1 Simulated Experiment for Profiled HO-Attacks

8.1.1 The Simulations

8.1.2 Comparison between KDA and MLP

8.2 Real-Case Experiments over ARM Cortex-M4

Chapter 9

Siamese Neural Networks for Collision Attacks

9.1 Introduction

9.2 Siamese Neural Networks

9.2.1 Distances and Loss Functions

9.2.2 Relation with Kernel Machines

9.3 Collision Attacks with Siamese NNs

9.3.1 Experimental Results

Chapter 10

Conclusions and Perspectives

10.1 Summary

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

Scenario 3 and 4 of CARDIS '15 paper

Scenario 3.

Let C be now variable and let the other parameters be fixed as follows: $N_a = 100$, $N_z = 200$, $\#PoI = 3996$. 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. 5.4(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?

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 $\#PoI$ 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 $\#PoI$ 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](https://doi.org/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.
- [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 Peter A. 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](https://doi.org/10.1007/978-3-642-33460-3_22). URL: http://dx.doi.org/10.1007/978-3-642-33460-3_22.
- [Par] TELECOM ParisTech. “DPA Contest 4”. In: (). <http://www.DPAcontest.org/v4/>. URL: <http://www.DPAcontest.org/v4/>.
- [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.