LDA-Based Clustering as a Side-Channel Distinguisher

Rauf Mahmudlu[1,2], Valentina Banciu[1], Lejla Batina[2], and Ileana Buhan[1]

[1] Riscure BV, Delftechpark 49, 2628 XJ Delft, The Netherlands
lastname@riscure.com
[2] Digital Security Group, Radboud University, The Netherlands
r.mahmudlu@student.ru.nl,lejla@cs.ru.nl

Abstract Side-channel attacks put the security of the implementations of cryptographic algorithms under threat. Secret information can be recovered by analyzing the physical measurements acquired during the computations and using key recovery distinguishing functions to guess the best candidate. Several generic and model based distinguishers have been proposed in the literature. In this work[1] we describe two contributions that lead to better performance of side-channel attacks in challenging scenarios. First, we describe how to transform the physical leakage traces into a new space where the noise reduction is near-optimal. Second, we propose a new generic distinguisher that is based upon minimal assumptions. It approaches a key distinguishing task as a problem of classification and ranks the key candidates according to the separation among the leakage traces. We also provide experiments and compare their results to those of the Correlation Power Analysis (CPA). Our results show that the proposed method can indeed reach better success rates even in the presence of significant amount of noise.

1 Introduction

Side-Channel Analysis (SCA) attacks have become a powerful tool for extracting secret information from cryptographic devices since the introduction of Differential Power Analysis (DPA) by Kocher et al. [18]. These attacks exploit the relationship between the side-channel measurements and the data-dependent leakage models to reveal some part of the key. The Correlation Power Analysis (CPA) method [6] is among the most efficient distinguishers when the relationship of the leakage and data can be approximated with a linear model. However, due to process variation in nano-scale devices and consequently the increase in the contribution of the leakage component of the power consumption, different leakage models become necessary. Since the performance of the CPA method strongly depends on the assumed (linear) leakage model, imprecise predictions can lead to complete failure of the method. Another major cause of the sub-optimal performance of key recovery attacks is the presence of noise in leakage

[1] This paper has been accepted for publication at RFIDSec 2016 and will be available at link.springer.com.

traces. While the performance of all SCA distinguishers are similar for a large Signal-to-Noise Ratio (SNR) [20], in real world scenarios it is common that the physical leakage measurements contain a significant amount of noise originating from multiple sources such as the power supply, the specifics of the measurement set-up, the clock generator, parallel computations etc. As discussed by Mangard et al. [19], the success of SCA attacks is heavily dependent on the SNR, and thus multiple noise reduction methods such as filtering, Principal Component Analysis (PCA) [17], Linear Discriminant Analysis (LDA) [15], singular spectrum analysis [16] etc. have been studied in the domain of SCA attacks.

Summarizing, we note that there are two main directions for improving key recovery methods: finding optimal distinguishers, and reducing the noise level in measurements. In this work we shall address and combine both aspects.

1.1 Related Work

With respect to data (pre-)processing and transformation methods, various ideas ranging from machine learning, pattern recognition and other localization techniques have been suggested. As an example, some of the techniques have been utilized for conducting template attacks as first introduced by Chari et al. [8]. Template attacks are the strongest form of side-channel attacks from the information theoretic point of view, and can successfully extract secret information from a limited number of traces. These attacks are typically carried out in two main steps: a profiling step during which templates corresponding to each sub-key candidate are derived, and a template matching step during which a new trace is matched to the templates.

LDA and PCA are among the data transformation methods that have been used [1,9,22] for feature extraction and dimensionality reduction in template attacks. While the performance of PCA-based attacks is close to that of LDA-based attacks when the measurements feature a high SNR, it deteriorates substantially when the SNR gets lower. LDA-based template attacks have been shown to lead to better templates especially in the presence of higher noise levels, because of the better separation of the classes in the transformed subspace and the near-optimal noise reduction [7]. PCA has also been studied for both data preprocessing and as a method for key recovery. Batina et al. [4] propose to utilize it as a preprocessing technique before conducting the DPA attack. The observed benefits of PCA in such scenarios are the noise reduction in the traces and the better performance of the DPA after the transformation of the traces into a lower dimension subspace spanned by eigenvectors. In contrast to this, Souissi et al. [21] have investigated the applicability of the PCA as another distinguisher by merely using the first principal component.

The Differential Cluster Analysis (DCA) technique introduced by Batina et al. [2] is also framing key recovery as a classification problem. The authors use metrics such as *sum-of-squared-error* and *sum-of-squares* to derive statistics about clusters. This method does not require an accurate leakage model, however including it would enhance the performance. The ANOVA (ANalysis Of VAriance) F-test is using a distance measure between the classes, which is similar

to what we propose in this work [5]. The metric called Normalized Inter-Class Variance (NICV) is used for leakage detection in SCA. While efficient in determining the time where the sensitive information is computed, comparing different leakage models or speeding up attacks on asymmetric cryptography, this method cannot be used as a distinguisher for recovering the secret information.

1.2 Contribution

Our main contribution is a new distinguisher which exploits the near-optimal noise reduction offered by the LDA transformation. The new distinguisher is versatile and can be adapted to any leakage model. We test the performance of our distinguisher using two different low SNR trace sets and show that it has superior performance compared to CPA.

This paper is organized as follows. In Section 2 we discuss background information relevant to this work. In Section 3 we introduce our attack method. In Section 4 we address the caveats. In Section 5 we discuss the results of our experiments and compare the Global Success Rates (GSR) of our attack to that of standard CPA. We conclude in Section 6.

2 Background

Let X denote a random variable over a space $\mathcal{X}$ with realization x. $\mathbf{X}$ is a d-dimensional $(X_1, X_2, \ldots, X_d) \in \mathcal{X}^d$ row vector with realization $\mathbf{x}$.

The term classification has two distinct meanings: first, it can mean the process of assigning an instance to a category, and second, it can mean finding categories or clusters within some data set; note that in the first case, a separation of the data space into classes is readily available. Throughout this paper, by classification we refer to the second meaning.

2.1 Side-Channel Analysis

We adopt the terminology and notations of [3], and consider the schematic representation of a classic SCA represented in Figure 1. In this scenario, a targeted cryptographic implementation is performing an encryption $E_k(p)$ of the plaintext p using a constant key k. During computation, the sensitive intermediate value $V_{s,p}$ that depends on a part s of the key k, and the plaintext p are handled. The physical leakage generated during the computation of $V_{s,p}$ is denoted as $Y_{k,p}$ since the leakage may potentially depend on the whole key k. The adversary acquires leakage traces by sampling or measuring the side-channel observables (power, electromagnetic emanation) at successive time instances. The value $Y_{k,p}$ can be captured in one sample or spread over multiple samples depending on the implementation details and the parameters of the acquisition. To recover the key, the adversary predicts the intermediate values handled during the computation of $E_k(p)$ and calculates the values $V_{j,p}$ for every possible subkey candidate

$j \in \mathcal{S}$. The adversary maps the intermediate values $V_{j,p}$ to the hypothetical leakage value $X_{j,p}$ by applying an estimated leakage model. To recover k the same steps are repeated for all the subkeys s.

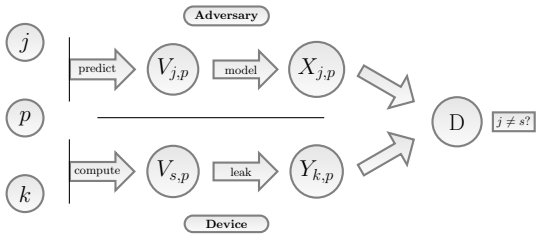


Figure 1: Schematic illustration of a side-channel key recovery

2.2 Linear Discriminant Analysis (LDA)

LDA is a dimensionality reduction technique used for classification purposes in machine learning, pattern recognition, etc. For a given data set, LDA seeks the linear combination of features which preserves the class-discriminant information. Then, the between-class (S_B) and within-class(S_W) scatter matrices are calculated according to Equation (1) and Equation (2) respectively, where μ is the mean of all the observations.

$$S_B = \sum_{j=1}^{|\mathcal{C}|} N_j (\mu_j - \mu)(\mu_j - \mu)^T \tag{1}$$

$$S_W = \sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{N_j} (\mathbf{x}_{i,j} - \mu_j)(\mathbf{x}_{i,j} - \mu_j)^T \tag{2}$$

The two matrix values are used to find the projection directions W which maximize the separation between classes. The separation - J between the classes is calculated according to Equation (3). After determining the projection directions, the observations are transformed to the new space as $\widehat{\mathbf{x}} = \mathbf{x}W$.

$$J(W) = \frac{W^T S_B W}{W^T S_W W} \tag{3}$$

2.3 Information theoretic definitions

The *entropy* of a random variable X [10] represents the uncertainty or the amount of information content and is defined as:

$$H[X] = \sum_{x \in \mathcal{X}} Pr[X = x] \cdot \log \left(\frac{1}{Pr[X = x]} \right). \tag{4}$$

The *conditional entropy*, $H[X|Y]$ of a random variable X given variable Y is the measure of the uncertainty left about X when Y is known. Finally, *mutual information* $I(X;Y)$ is a measure of the dependence between the random variables X and Y and the amount of information they have in common.

2.4 Experimental Setup

For this research, we consider software implementations of AES128 [14] and DES [13] running on an ARM Cortex-M4F core based board operating at a 168 MHz clock frequency. The board has been physically modified and programmed in order to be a target for SCA and it accurately models current 32-bit embedded devices. As discussed in Section 1, the SNR of side-channel traces is an indication of their quality. Since we are interested in noisy side-channel traces, we acquire electro-magnetic (EM) measurements which have lower SNR than the power measurements (i.e., a more challenging scenario). To do so, we build a standard setup (as described e.g. in [19]). We utilize a PicoScope 3207B [23] digital oscilloscope with a 500 MHz sampling rate. We carry out two measurement campaigns (one for each cryptographic algorithm implementation), as follows:

$TraceSet_1$**:** 50 000 traces were obtained for the implementation of the AES128 algorithm. The key was fixed and the traces were obtained for random plaintext inputs. The SNR value is 1.01 dB.

$TraceSet_2$**:** 50 000 traces were obtained for the implementation of the DES algorithm. The key was fixed and the traces were obtained for random plaintext inputs. The SNR value is 2.78 dB.

The noticeable difference in the SNR values of the measurements originates from the architectural designs of the implementations. The parallel S-box lookups during the AES rounds generates more algorithmic noise which leads to a lower SNR value.

3 Attack description

The key recovery attack proposed in this paper relies on the central assumption that all leakages corresponding to the processing of some fixed key dependent intermediate value are similar. In other words, when a set of physical leakages $Y_{k,p}$ is classified according to the values of $X_{s,p}$ as defined in Section 2.1, the between-class to within-class scatter matrices ratio is large. Note that the above requirement is indeed met in the context of side-channel attacks, as the instantaneous power consumption of a cryptographic implementation is generally expected to be data dependent. However, in practice side-channel measurements often include noise, which leads to a weaker separation amongst classes and in consequence decreases the success rate of key recovery attacks.

The approach proposed in this work targets such challenging scenarios where the SNR is low, and achieves key extraction with fewer traces. It consists of two steps: (i) the leakage transformation step; and (ii) the distinguishing step.

In the following we describe in more detail the working principles of our attack. In Section 3.1 we describe how parts of the plaintext can be used for classification purposes and how measured leakages can be projected into a subspace where they are maximally separated and the SNR level is higher. Then in Section 3.2 we propose a function that enumerates subkeys based on the separation of the model based classes.

3.1 The Leakage Transformation Step

The objective of the leakage transformation step is to identify and select time samples where the difference between mean traces corresponding to distinct classes of intermediates is maximized. In order to apply a LDA transformation in this step, information that allows for the separation of traces into classes must be available, e.g. one must know the plaintexts or ciphertexts. The sensitive key dependent intermediate variables are predicted as $V_{j,p}$, as represented in Figure 2. Although the correct intermediate values $V_{s,p}$ depend on the unknown subkey $s \in \mathcal{S}$, they may still be classified based only on the value of the plaintext due to the fact that for any $j \in \mathcal{S}$ and $(p_1, p_2) \in \mathcal{P}$, if $p_1 = p_2$ then $V_{j,p_1} = V_{j,p_2}$. After separating the physical leakages into groups based on the plaintext or ciphertext values, the projection directions are calculated and the leakages are projected onto the new subspace. The transformed leakages are subsequently used for key recovery, as represented in Figure 2.

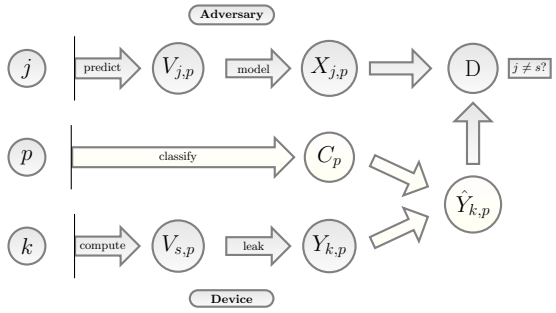


Figure 2: Schematic illustration of the proposed attack

3.2 The Distinguishing Step

The objective of this step is to distinguish between the key candidates. Note that because the traces have previously been linearly transformed to maximize the separation between classes, the correlation between the traces and the hypothetical power consumption may be lost. By definition, the transformation is the sum of the inner product between the leakage with the projection directions

where each direction is a column of the transformation matrix $\widetilde{W}$. It follows that the magnitudes of the coefficients of each direction are proportional to the contribution of the corresponding samples to the transformation. Figure 3 shows the Pearson correlation coefficients and the first projection direction for $TraceSet_1$. While there are clear peaks in the 159 µs to 164 µs time interval, the dominating samples in the first projection direction are situated in different regions. Therefore, the need for a new distinguisher that better matches the properties of the transformed traces arises. To this end, we propose to use the ratio of the between- and the within-class scatter. The features extracted through the LDA transformation correspond to the linear combination of the leakage samples that maximally separate classes. At the same time, for a given leakage model, traces corresponding to the same values of $X_{s,p}$ are expected to have similar features. Since for each projection direction the contribution of each sample of the side-channel leakages towards this direction is the same, when the projected leakages are labelled according to the model obtained from the correct key, the separation of the clusters should be maximum. Whereas, if the model obtained from the wrong key is used for labelling, the lack of similar features within classes should lead to a weaker separation as shown in Figure 4. Since the objective of the distinguisher is to retrieve ordinal information about the variance of the ratio matrix, its largest eigenvalue can be a numerical measure for separation [24].

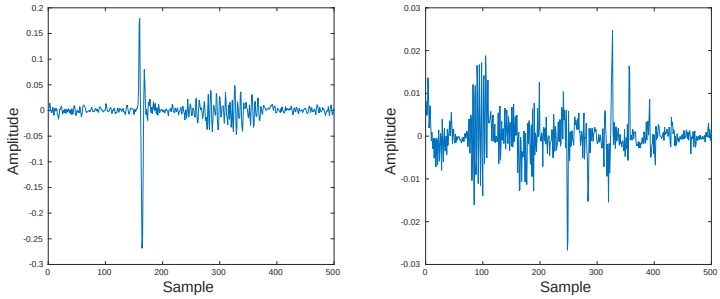
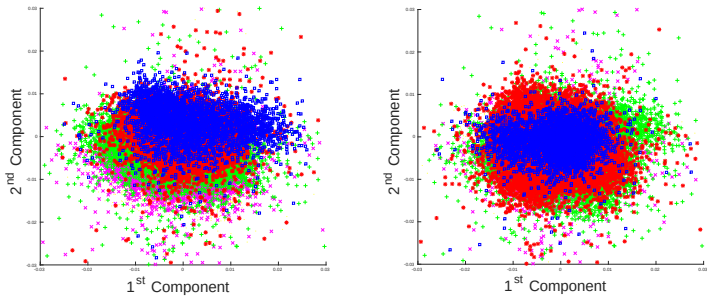


Figure 3: Known key correlation (left) vs. the first projection eigenvector (right)

Summarizing, in the second stage $|\mathcal{S}|$ models (each corresponding to a different $j \in \mathcal{S}$) are computed and the transformed physical leakages are classified accordingly. After calculating between-class ($\hat{S}_B$) and within-class ($\hat{S}_W$) scatter of $\hat{Y}_{k,p}$, the diagonal matrix of eigenvalues $\hat{\Delta}$ is calculated by eigendecomposing $\hat{S}_W^{-\frac{1}{2}} \hat{S}_B \hat{S}_W^{-\frac{1}{2}} = \hat{U} \hat{\Delta} \hat{U}^T$. The eigenvalue is assigned as the candidate score. Finally, the candidate leading to the largest score is selected as the correct key.



(a) Correct key candidate (b) Incorrect key candidate

Figure 4: Visualization of the class separation under different key candidates

4 Caveats

In this section we explore the two caveats of our method, which are due to intrinsic characteristics of the LDA transformation.

First, the number of side-channel traces must be larger than the number of analysed samples. To overcome the need for a very large trace set, it is possible to analyse only a selected block of samples at a time. In this case for each key candidate the number of discriminant scores will be the same as the number of blocks. If a selected block does not include samples related to the calculation of the predicted intermediate values, classification of the leakages according to possible values of the subkey candidate will not be significantly different from each other. Whereas, in the block where leakage occurs, the correct key candidate should lead to significantly better separation among the classes. In order to find the block where the leakage occurs, the scores for each block have to be normalised and the one with the highest ratio of the scores for the first and second candidates is chosen as the leaking block. The first candidate of the leaking block is subsequently chosen as the correct key.

Second, the size of the plaintext space $\mathcal{P}$ must be reasonably small. To estimate the between-class scatter, more than one trace should belong to each class. Since in the classification and transformation stage the number of classes is equal to $|\mathcal{P}|$, the number of leakage traces needed for finding the projection directions would be significantly high. This restriction can be avoided by obtaining the measurements for chosen plaintexts such that text space size is small.

5 Experimental Validation

We now validate our attack methodology using the trace sets described in Section 2.4 under different leakage assumptions. Section 5.1 describes the calculation of the projection directions and transformation of the traces. In Section 5.2 we describe the attacks where the hypothetical power consumption is linked to the

Hamming weight (HW) of intermediate values, and in Section 5.3 we describe how the (partial) identity leakage model can be exploited. We report the performance of the attacks by looking at the GSR, i.e. the ratio of the correctly guessed subkeys to the total number of subkeys.

5.1 Leakage Transformation

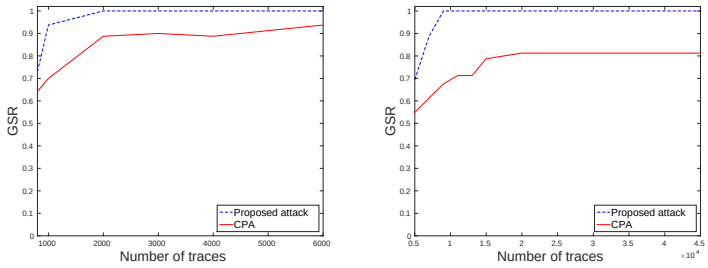
As described in Section 2.2, the projection directions that will map the traces into a new subspace where the ratio of the between-class (S_B) and within-class(S_W) scatter matrices are maximised have to be calculated. During the calculation of these matrices the traces are classified as described in Section 3.1. The matrix of projection directions is built as $W = S_W^{-\frac{1}{2}}U$ [9], where U is the matrix of eigenvectors obtained by eigendecomposing $S_W^{-\frac{1}{2}}S_BS_W^{-\frac{1}{2}} = U\Delta U^T$. Δ denotes diagonal matrix of eigenvalues. The projection matrix can be truncated according to the Eckart-Young theorem [12] as $\widetilde{W} = S_W^{-\frac{1}{2}}\widetilde{U}$, where $\widetilde{U}$ is the matrix of eigenvectors corresponding to the m largest eigenvalues.

5.2 HW Leakage Model

As shown in Figure 1, the intermediate values for both of the implementations are predicted as $V_{j,p} = \text{Sbox}(j \oplus p)$ and the leakages are modelled as the HW of the intermediate values. The subkeys of the first round key were targeted at every implementation with the goal of recovering the full round key. As studied by Doget et al. [11], when the chosen leakage model exactly corresponds to the actual leakage function of the implementation, CPA has one of the best performances for key extraction. Therefore, we have used this method as a reference for comparing the performance of the proposed attack. It should be noted that while the CPA attack is based upon an assumption of linear dependence between the HW of the intermediate values and the actual power consumption, our attack does not require such a strict relation. We only assume that the power consumption corresponding to the processing of intermediate values that have the same HW is consistent and it differs from that corresponding to other HW values.

For CPA attacks, the hypothetical power consumption models for each possible value of the subkey were built and the Pearson correlation coefficients were calculated for each sample of the trace sets. The key candidate which maximizes the absolute value of the correlation coefficient was chosen as the correct key. Both the proposed attack and CPA were run on randomly selected subsets of the trace sets multiple times and the average results were compared. Figure 5 reports the GSR for both implementations. This figure clearly shows that the proposed attack is outperforming CPA for both implementations.

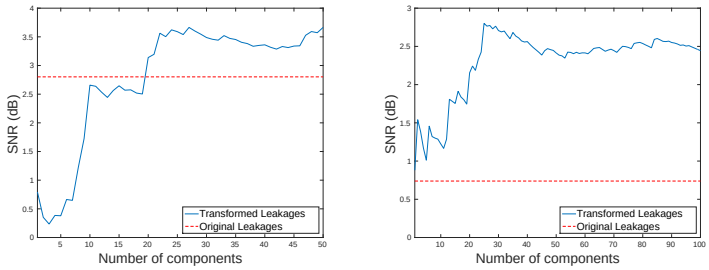
The analysis of the leakage traces after the LDA transformation shows that depending on the number of retained components, the SNR level can be significantly higher compared to the original traces. The graph in Figure 6 shows the SNR levels as the function of the projection directions retained after the transformation. Since the increase in SNR together with the supervised classification



(a) DES implementation (b) AES implementation

Figure 5: Global Success Rate (GSR)

are the reasons for the better performance of the proposed attack method, it is important to select a significantly large number of components. We have adapted the heuristics of keeping the directions corresponding to the 95th percentile of the eigenvalues after the eigendecomposition of $S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}}$ [25].



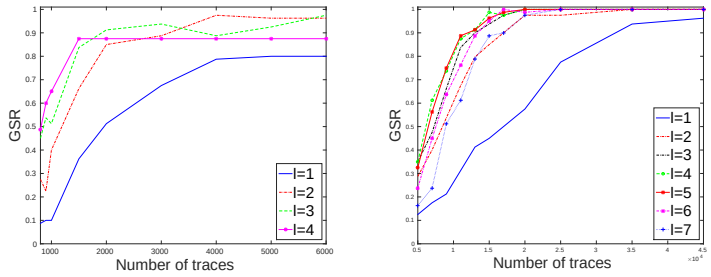
(a) DES implementation (b) AES implementation

Figure 6: The SNR before and after the LDA transformation

5.3 Identity Leakage Model

To further extend our experiments, we have also investigated key extraction when no assumptions about the leakage model are made. To this end, instead of classifying leakage traces according to the HW of intermediate values, we separate them according to some selected bits of the intermediate values. Due to intrinsic properties of the AES ans DES encryption algorithms (in particular: the bijectivity of the S-box), we will analyse them separately.

AES Encryption. The intermediate values in this case were also predicted as $V_{j,p} = \text{Sbox}(j \oplus p)$. The classification of the leakage traces does not depend on



(a) GSR for DES implementation (b) GSR for AES implementation

Figure 7: GSR for the target implementations

the value of key candidate j due to the bijectivity of the S-box function. The absence of mutual information leads to the conclusion that the classification based on the hypothetical intermediate values will be the same for each key candidate. Therefore, instead of assigning identical intermediate values, we assign similar intermediate values to the same class. In this context, we define similar intermediate values as those whose preselected $l \in \{1 \dots 7\}$ bits are equal.

DES Encryption. The intermediate values were again chosen as $V_{j,p} = \text{Sbox}(j \oplus p)$. The mutual information between the classification based on the intermediate values and the key candidate is larger than 0 due to the non-bijectivity of the S-boxes. Therefore, it is possible to select l in the interval of $\{1 \dots 4\}$.

As can be seen from the results plotted in Figure 7, the GSR for the Identity Model is lower than that of the HW model when the implementations are attacked with the proposed method. When compared to the results of CPA, it can be observed that depending on the number of selected bits and traces the new attack can be more successful in extracting the subkeys. The empirical study of the S-box functions of the encryption algorithms reveals that the mutual information between the key candidate and the classification increases with decreasing l (see Table 1), while the GSR does not follow the same pattern. When l gets smaller, the number of distinct intermediate values that are assigned to the same class increases, which leads to weaker separation among classes. Therefore, a compromise between getting maximum possible mutual information and keeping the classes well separable has to be made. Given that for fairly large amount of traces the performance of the attack is better than CPA even without making any assumptions about the leakage model, we can argue that the proposed attack is preferable.

Table 1: The analysis of the mutual information between the key and the classification for AES and DES S-box outputs

	Mutual Information								
	AES	DES							
1	S	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
8	0	–	–	–	–	–	–	–	–
7	0.060	–	–	–	–	–	–	–	–
6	0.116	–	–	–	–	–	–	–	–
5	0.204	–	–	–	–	–	–	–	–
4	0.340	0.306	0.294	0.302	0.263	0.285	0.297	0.306	0.296
3	0.545	0.538	0.541	0.541	0.543	0.537	0.544	0.544	0.537
2	0.812	0.859	0.832	0.858	0.827	0.831	0.861	0.867	0.856
1	1.001	1.037	1.023	1.039	1.025	1.011	1.028	1.053	1.019

5.4 Computational Complexity

While the success rates of different key extraction attacks may be high, their adaptation in real world scenarios is also bounded by the computational complexity. Since the side-channel security evaluations of cryptographic devices can involve millions of traces, it is desirable to be able to perform the analysis within the bounds of target time interval. We note that it is not feasible to run the analysis using the proposed method on a large number of traces.

The analysis of the attack algorithm described in Algorithm 1 shows that the costly part is the transformation of the original leakage traces to the new subspace spanned by the eigenvectors of the ratio of scatter matrices. In particular, the calculation of the between-class and within-class scatter matrices have the complexity of $\mathcal{O}(md^2)$ where m is the number of leakage traces and d is the number of samples. Similarly, the complexities of the operations in lines 2-4 are equal to $\mathcal{O}(d^3)$. Since the number of traces is larger than the number of samples as described in Section 4, the complexity of the attack is $\mathcal{O}(md^2)$. The linear relation between the computational complexity and the number of traces implies that the attack can indeed be carried out using large number of leakage traces if the number of samples per trace is kept small.

6 Conclusion

In this paper we have introduced a new method for conducting a key recovery side-channel attack. We have described how the matrix that transforms the side-channel leakage traces into a new subspace where the SNR is increased can be constructed. Later, a distinguisher which compares the classifications of the traces based on different values of the key candidates has been introduced. The method has been tested against noisy trace sets with and without making assumptions about the leakage model of the implementations. We have also discussed the theoretical restrictions arising from the application of the LDA transformation and proposed a method for achieving a higher GSR with lower number of traces. The experiments conducted on the software implementations

Algorithm 1: Pseudo-code of the proposed attack

Input: Matrix of leakage traces: Y $(m \times d)$
Input: Vector of plaintexts: P $(m \times 1)$
Output: Vector of key candidate scores: k $(|\mathcal{S}| \times 1)$

$[S_W, S_B] = \text{scatter}(Y, P)$;
$T = S_W^{-\frac{1}{2}}$;
$M = TS_BT$;
$[U, \Delta] = \text{eig}(M)$;
$I = \text{sort}(\Delta)$;
$\widetilde{U} = U(I)$;
$\widetilde{W} = T\widetilde{U}$;
$\widehat{Y} = Y\widetilde{W}$;
for $j \in \mathcal{S}$ **do**
    $X_P = \text{model}(P, j)$;
    $[\widehat{S}_W, \widehat{S}_B] = \text{scatter}(\widehat{Y}, X_P)$;
    $\widehat{T} = \widehat{S}_W^{-\frac{1}{2}}$;
    $\widehat{M} = \widehat{T}\widehat{S}_W\widehat{T}$;
    $[\widehat{U}, \widehat{\Delta}] = \text{eig}(\widehat{M})$;
    $k(j) = \max(\widehat{\Delta})$;
end

of the AES and DES encryption have confirmed the efficiency of the proposed method. We have compared the new method to the CPA and have observed that significantly less number of traces were needed to achieve the same GSR.

Acknowledgments. This work has been funded partially by Riscure BV through the Internship@Riscure program, by the Dutch government and the Netherlands Technology Foundation STW through project 13499 - TYPHOON & ASPASIA, project 12624 - SIDES, and by the Netherlands Organization for Scientific Research NWO through project 628.001.007 - ProFIL.

References

1. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *Cryptographic Hardware and Embedded Systems–CHES 2006*, pages 1–14. Springer, 2006.
2. Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Differential Cluster Analysis. In *Cryptographic Hardware and Embedded Systems–CHES 2009*, pages 112–127. Springer, 2009.
3. Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: A Comprehensive Study. *Journal of Cryptology*, 24(2):269–291, 2011.

4. Lejla Batina, Jip Hogenboom, and Jasper GJ van Woudenberg. Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis. In *Topics in Cryptology–CT-RSA 2012*, pages 383–397. Springer, 2012.
5. Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage. In *International Symposium on Electromagnetic Compatibility, Tokyo–EMC 2014*, pages 310–313. IEEE, 2014.
6. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *Cryptographic Hardware and Embedded Systems–CHES 2004*, pages 16–29. Springer, 2004.
7. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is More - Dimensionality Reduction from a Theoretical Perspective. In *Cryptographic Hardware and Embedded Systems–CHES 2015*, pages 22–41. Springer, 2015.
8. Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template Attacks. In *Cryptographic Hardware and Embedded Systems–CHES 2002*, pages 13–28. Springer, 2002.
9. Omar Choudary and Markus G Kuhn. Efficient Template Attacks. In *Smart Card Research and Advanced Applications*, pages 253–270. Springer, 2013.
10. Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.
11. Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate Side Channel Attacks and Leakage Modeling. *Journal of Cryptographic Engineering*, 1(2):123–144, 2011.
12. Carl Eckart and Gale Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218, 1936.
13. PUB FIPS. 46-3: Data Encryption Standard (DES). *National Institute of Standards and Technology*, 25, 1999.
14. PUB FIPS. 197: Advanced Encryption standard (AES). *National Institute of Standards and Technology*, 26, 2001.
15. Ronald Aylmer Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of eugenics*, 7(2):179–188, 1936.
16. Nina Golyandina and Anatoly Zhigljavsky. *Singular Spectrum Analysis for Time Series*. Springer Science & Business Media, 2013.
17. Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.
18. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology–CRYPTO 1999*, pages 388–397. Springer, 1999.
19. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, volume 31. Springer Science & Business Media, 2008.
20. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for All – All for One: Unifying Standard Differential Power Analysis Attacks. *Information Security, IET*, 5(2):100–110, 2011.
21. Youssef Souissi, Maxime Nassar, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In *International Conference on Information Security and Cryptology*, pages 407–419. Springer, 2010.
22. François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *Cryptographic Hardware and Embedded Systems–CHES 2008*, pages 411–425. Springer, 2008.

23. Pico Technology. PicoScope 3000 Series. `https://www.picotech.com/download/datasheets/PicoScope3200ABSeriesDataSheet.pdf`, 2013.
24. Matthijs Joost Warrens. *Similarity Coefficients for Binary Data: Properties of Coefficients, Coefficient Matrices, Multi-Way Metrics and Multivariate Coefficients*. Psychometrics and Research Methodology Group, Leiden University Institute for Psychological Research, Faculty of Social Sciences, Leiden University, 2008.
25. Li-Jen Weng and Chung-Ping Cheng. Parallel Analysis with Unidimensional Binary Data. *Educational and Psychological Measurement*, 65(5):697–716, 2005.