MINISTRY OF EDUCATION & TRAINING
LE QUY DON TECHNICAL UNIVERSITY

DO NGOC TUAN

SIDE CHANNEL ATTACK BASED ON DISTRIBUTION OF SAMPLING CORRELATION AND MULTI-OUTPUT NEURAL NETWORK FOR HARDWARE SECURITY

Major: Electronic Engineering
Code: 9 52 02 03

SUMMARY OF TECHNICAL DOCTORAL DISSERTATION

HaNoi - 2023

THIS WORK IS COMPLETED AT
LE QUY DON TECHNICAL UNIVERSITY

Supervisor: 1. Assoc. Prof. HOANG VAN PHUC
2. Prof. PHAM CONG KHA

Opponent 1: Assoc. Prof. Tran Xuan Tu

Opponent 2: Assoc. Prof. Nguyen Hieu Minh

Opponent 3: Assoc. Prof. Nguyen Xuan Quyen

This thesis will be defended before The National-Level Doctoral Examination Board according to the Decision No 2789/QĐ-HV, 07/06/2023 of the President of Le Quy Don Technical University, meeting at the Le Quy Don Technical University on ../../20..

This thesis could be found at:
- National Library of Vietnam
- Library of Le Quy Don Technical University

PUBLICATIONS

J1. **N. -T. Do**, V. -P. Hoang and C. -K. Pham, “Low Complexity Correlation Power Analysis by Combining Power Trace Biasing and Correlation Distribution Techniques," *IEEE Access, (SCIE)*, vol. 10, pp. 17578-17589, 2022, doi: 10.1109/ACCESS.2022.3150833.

J2. V. -P. Hoang, **N. -T. Do** and V. S. Doan, "Efficient Non-profiled Side Channel Attack Using Multi-output Classification Neural Network," in *IEEE Embedded Syst. Lett, (SCIE)*, 2022, doi: 10.1109/LES.2022.3213443.

J3. **Ngoc-Tuan Do**, Van-Phuc Hoang, Van Sang Doan, Cong-Kha Pham, “On the Performance of Non-Profiled Side Channel Attacks Based on Deep Learning Techniques," *IET Inf. Sec., (SCIE)*, 2022, doi: 10.1049/ise2.12102.

J4. **Ngoc-Tuan Do**, Van-Phuc Hoang, Van Sang Doan, “A novel non-profiled side channel attack based on multi-output regression neural network," *J Cryptogr Eng, (SCIE)*, 2023, https://doi.org/10.1007/s13389-023-00314-4

J5. V.-P. Hoang, **N.-T. Do**, V. S. Doan, “Performance Analysis of Deep Learning Based Non-profiled Side Channel Attacks Using Significant Hamming Weight Labeling," *Mobile Networks and Applications,(SCIE)*, 2023, Accepted.

C1. **N. -T. Do**, V. P. Hoang, “An Efficient Side Channel Attack Technique with Improved Correlation Power Analysis," in *INISCOM 2020*, vol 334. Springer, Cham. https://doi.org/10.1007/978-3-030-63083-6_22. (Scopus)

C2. **N. -T. Do**, V. -P. Hoang and V. -S. Doan, “Performance Analysis of Non-Profiled Side Channel Attacks Based on Convolutional Neural Networks," *APCCAS 2020*, 2020, pp. 66-69, doi: 10.1109/APCCAS50809.2020.9301673.

C3. **N. -T. Do**, V. -P. Hoang and V. -S. Doan, “Performance Analysis of Non-profiled Side Channel Attack Based on Multi-Layer Perceptron Using Significant Hamming Weight Labeling," in *INISCOM 2022*, vol 444. (Scopus)

C4. **Ngoc-Tuan Do**, P. C. Le, V. P. Hoang, V. S. Doan and C. K. Pham, “MO-DLSCA: Deep Learning Based Non-profiled Side Channel Analysis Using Multi-output Neural Networks," ATC 2022, pp. 245-250, (Scopus)

CONCLUSIONS AND SUGGESTIONS FOR FUTURE STUDIES

Research on SCA attacks is crucial for detecting and preventing potential hardware security problems. This thesis introduces different effective SCA methods to reduce the attack time and enhance the success rate of SCA security testing. The major contributions of the thesis can be summarized as follows:

- Two low-complexity correlation power analysis (CPA) techniques called P-CPA and BP-CPA, are proposed based on the correlation distribution and power trace biasing technique [C1], [J1].
- A dimensional reduction based on P-CPA and the SHW labeling techniques are proposed for solving imbalanced dataset problems, reducing the measurements needed as well as improving the performance of non-profiled DLSCA [C2, C3], [J3, J5], [P1].
- Several DLSCA techniques based on multi-output neural networks are proposed to enhance significantly the performance of non-profiled DLSCA regarding both the execution time and the success rate of SCA evaluation processes [C4], [J2, J4].

Besides numerous effectiveness in attacking SCA data, the proposals still contain several limitations, such as the complexity of second-order preprocessing and the number of measurements needed for MO-DLSCA. For future works, we focus on the following directions:
- Firstly, future works will be directed toward investigating the suggestions of SCA countermeasures against proposed attacks in the real scenario.
- Secondly, investigate BP-CPA techniques on different SCA platforms, especially in hardware implementation.
- Finally, other advanced DL architectures, such as LSTM, and RNN, will be investigated in the SCA domain. More importantly, an online DLSCA method will be employed to reduce the attack time and determine the minimum measurements needed for DLSCA.

INTRODUCTION

Side-channel attacks (SCAs) that exploit the information leakage from the operations of a cryptographic device have become a realistic threat to the implementations of cryptographic algorithms. Various countermeasures have been suggested in order to improve the hardware security against SCA. However, many real attacks must be performed repeatedly to clarify the effectiveness of the underlying countermeasure in practice. Therefore, an efficient SCA evaluation that certifies the SCA resistance plays a vital role in reducing the time to market. More importantly, it helps designers to detect potential hardware threats.

In this thesis, various effective SCA techniques based on statistics and deep learning (DL) are proposed to reduce the computation time and enhance the success rate of attack-based security testing. Concretely, the proposed techniques focus on improving the efficiency of the most commonly used non-profiled SCA techniques like CPA and DDLA in different evaluation conditions, such as high dimensional data, imbalanced datasets, and the presence of the SCA countermeasures. The main contributions of the dissertation can be summarized as follows:

- To propose low-complexity Correlation Power Analysis (CPA) techniques based on the distribution of sampling correlation and the power trace biasing technique.
- To introduce dimensional reduction and labeling techniques for DL-based non-profiled SCA (non-profiled DLSCA).
- To propose new SCA techniques based on multi-output neural networks (MO-DLSCA) that reduce the attack time from several hours to less than half hours.

The thesis is organized into four chapters. Chapter 1 gives the backgrounds related to SCA attacks. Chapter 2 presents the improved CPA techniques based on the distribution of sampling correlation. The two remains chapters introduce the proposals for SCA attacks using different deep learning techniques in a non-profiled context.

Chapter 1

OVERVIEW OF SIDE-CHANNEL ATTACKS

1.1 Introduction

Side-channel attack (SCA) is the most popular attack on cryptographic devices that has received significant attention from the hardware research community in recent years. The basic idea of SCA is to reveal the secret key of a cryptographic device by analyzing its side-channel data, such as timing, power consumption, and electromagnetic (EM) radiation, as depicted in Fig. 1.1. These side-channel data are relatively easy to be acquired from various electronic devices.

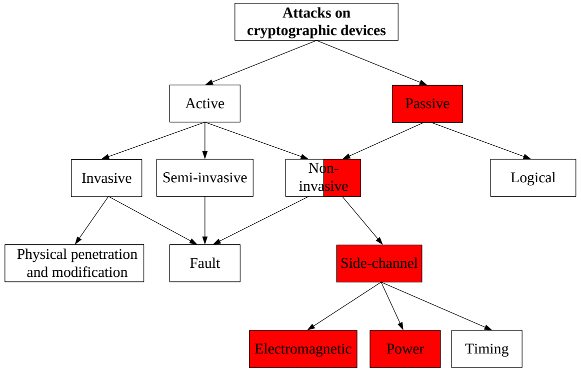


Figure 1.1: Classification of attacks on cryptographic devices. The red parts indicate the attacks that are covered in this thesis.

This thesis focuses on the SCA techniques using power consumption recorded directly from cryptographic devices by a power measurement circuit or indirectly by an EM probe.

1.2 Side-channel data and measurement setup

Digital circuits are built based on the CMOS logic cells, which are constructed from the complementary pull-up and pull-down networks. The logic cells in the circuit process the input signal and draw the current from the power supply. Therefore, the total power consumption of a CMOS cir-

the attack time is reduced significantly, especially in the case of the proposed technique based on the MOC model, as shown in Table. 4.2. More interestingly, DL-based attacks could reveal the secret key without any preprocessing techniques. However, DL techniques require a huge number of measurements compared to statistic-based techniques.

4.6 Summary

This chapter proposes different DL models based on the multi-output neural network. The proposals can reveal the correct key after only one training process. The experimental results have indicated that the proposed models remarkably outperform the DDLA attack in terms of execution time and success rate. However, the attack results are not optimized by using a fixed number of epochs. In addition, noise generation countermeasure is also investigated as a potential candidate for preventing MO-DLSCA.

Table 4.2: The comparison of attack results on masking countermeasure using different models.

Model	Data	No. of traces	No. of epochs	Attack time (second)	Results (*)	Power model
MLP_{DDLA}[1]	ASCAD	20,000	30	772.2	S	LSB
$\mathbf{MLP}_{\mathbf{MOC}}$[1]		20,000	30	**109.1**	S	LSB-vector
MLP_{DDLA}[2]		20,000	50	1950.9	S	LSB
MLP_{PL}[2]		20,000	50	693.8	S	LSB
$\mathbf{MLP}_{\mathbf{SL}}$[2]		20,000	50	**14.5**	S	LSB
1-order CPA[1]		1,200	-	-	F	HW
2-order CPA[1]		1,200	-	1188.4	S	HW
BP-CPA[1]		**1,200**	-	446.9	S	HW
MLP_{DDLA}[1]	CHES2018-CTF	40,000	16	N/A	F	LSB
MLP_{MOR}[1]		40,000	16	N/A	S	ID
CNN_{DDLA}[2]	CW-shifted	10,000	50	7069.1	S	LSB
CNN_{PL}[2]		10,000	50	3983.3	S	LSB
$\mathbf{CNN}_{\mathbf{SL}}$[2]		10,000	50	**31.9**	S	LSB
CNN_{DDLA}[1]		10,000	100	20,792.4	S	LSB
CNN_{MOC}[1]		10,000	100	703.7	S	LSB-vector
$\mathbf{CNN}_{\mathbf{MOR}}$[1]		10,000	50	**270**	S	ID
1-order CPA[1]		10,000	-	-	F	HW

[1] This work, Keras, Intel Corei5-9500, 24GB RAM; (*) S: Success; F: Failed

[2] Keras, NVIDIA GeForce GTX 1080Ti GPU, Intel Corei7-8700K, 48GB RAM.

the loss metric of a training process is exploited to reveal the correct key. As depicted in Fig. 4.5, these results demonstrate that the CNN model can break the de-synchronization countermeasure based on the translation-invariance property. However, the attack time of CNN_{MOC} is shorter by approximately 30 times compared to CNN_{DDLA}. In addition, CNN_{MOC} provides a clear distinction at a very early epoch compared to that of CNN_{DDLA}.

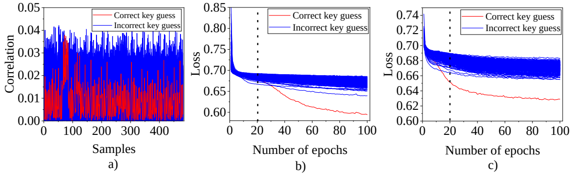


Figure 4.5: Attack results on de-synchronized power traces using CPA, CNN_{DDLA}, and CNN_{MOC}. a) CPA; b) CNN_{DDLA}; c) CNN_{MOC}.

As depicted in Fig 4.6, all CNN_{MOR} models provide good discrimination of the correct key and the incorrect ones. Interestingly, the results are better when the number of filters increases. Compared to CNN_{DDLA}, CNN_{MOR} does not achieve a clear distinction, especially in the case of CNN_{MOR4} as depicted in Fig 4.6.a.

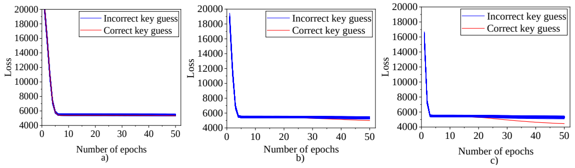


Figure 4.6: Attack results on de-synchronized power traces using the MOR-CNN models. a) 4 filters; b) 8 filters; c) 16 filters

In summary, the attack results on the masking dataset show that statistic-based techniques could reveal the secret key with only 1200 power traces. However, the drawbacks of these attacks are high memory usage required and time-consuming. Moreover, a pre-processing technique must be applied to attack high-order leakage data. Compared to DL-based techniques,

cuit is the sum of the power consumption of logic cells making up the circuit. Digital systems usually draw both *dynamic* and *static* power. Dynamic power is the power used for charging the capacitance as signals change between 0 and 1. Static power is the power used even when signals do not change and the system is idle. In the SCA domain, dynamic power consumption is considered to exploit in most cases.

For power analysis attacks, it is necessary to measure the power consumption of a cryptographic device while it executes cryptographic algorithms. Fig 1.2.b depicts a typical measurement setup that requires some components, such as a monitoring personal computer (PC), the device under test (DUT), and a digital oscilloscope.

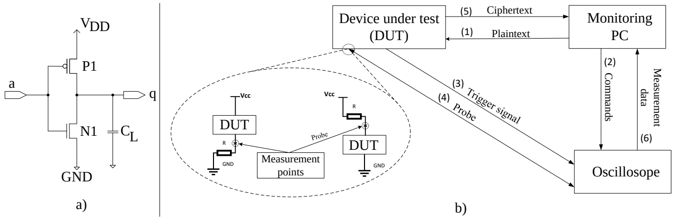


Figure 1.2: Power consumption measurement setups.

1.2.1 *The data used in thesis*

The datasets used in this thesis are divided into two groups: unprotected and protected datasets. The AES-128 algorithm is selected to perform the power consumption data acquisition since it has a wide range of applications. Regarding unprotected AES-128, the side-channel data were recorded from ChipWhisperer[1] (CW) and Sakura-G board[2], which are popular platforms in the SCA research community. To investigate the SCA attacks on protected AES algorithm, the thesis uses a public ASCAD[3] dataset and CHES2018-CTF[4]. Simultaneously, two other simulated datasets are created to investigate the hiding countermeasures.

[1] https://chipwhisperer.readthedocs.io/en/latest/getting-started.html
[2] https://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html
[3] https://github.com/ANSSI-FR/ASCAD
[4] https://chesctf.riscure.com/2018/content?show=training

1.3 Non-profiled SCA methods

In the non-profiled context, CPA and Differential Deep Learning Analysis (DDLA) are the most attractive SCA techniques in the research community.

Correlation Power Analysis (CPA)

CPA[5] was proposed by Brier *et al.* in 2004, and can be considered a special form of the PPA attack. CPA exploits the correlation between the real power consumption and the power consumption model Hamming weight (HW) or Hamming distance (HD) of the manipulated data.

In the CPA attack, the Pearson correlation coefficient is the common measure to determine the linear relationship between two variables. The attacker estimates the correlation ρ between two variables based on D power traces as follows:

$$r_{j,s} = \frac{\sum_{i=1}^{D} (h_{i,j} - \bar{h}_j)(t_{i,s} - \bar{t}_s)}{\sqrt{\sum_{j=1}^{D} (h_{i,j} - \bar{h}_j)^2 \sum_{j=1}^{D} (t_{i,s} - \bar{t}_s)^2}} \tag{1.1}$$

where $\bar{h}_j$ and $\bar{t}_s$ are the average values of the power consumption model and real power consumption at the instant t_s $(1 \leq s \leq S)$, respectively. S denotes the number of samples on a power trace.

The Pearson correlation between the power consumption model and the real power consumption is calculated for every value of k and t_s. It results in the matrix $\boldsymbol{R} = r_{1\ldots K,1\ldots S}$ of correlation coefficients.

Deep learning based non-profiled SCA

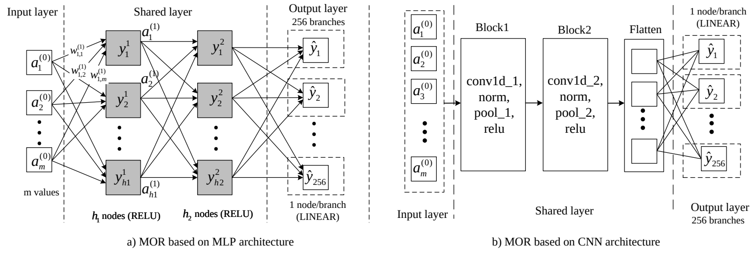
Algorithm 1[6] summarizes the DDLA procedure to perform a non-profiled attack using Deep Learning.

1.4 SCA for hardware security

Research on SCA techniques is important in hardware security. On the one hand, it can be used to develop appropriate SCA countermeasures. On the other hand, SCA techniques are applied to evaluate the efficiency of the equipped countermeasures as depicted in Fig. 1.3. Regarding SCA

[5] Brier, E., Clavier, C., Olivier, F. (2004). Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, JJ. (eds) CHES 2004

[6] Timon, B. (2019). Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis. TCHES 2019, 107–131.



a) MOR based on MLP architecture

b) MOR based on CNN architecture

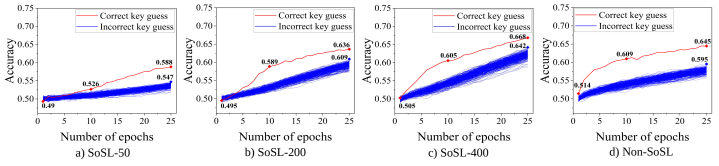
Figure 4.3: Structure of proposed multi-output regression neural network.

The most significant difference between the proposed MOC models and the proposed MOR models is that the output layer of each branch contains only one node instead of two nodes, as depicted in Fig. 4.3. Therefore, the separate losses of proposed MOR models are calculated by Eqs. (4.3) instead of Eqs. (4.1)

$$\mathcal{L}^{[k]}(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} (y_i - \hat{y}_i)^2 \tag{4.3}$$

4.5 Validation experiments

All experiments were performed by Keras framework on a personal computer with Intel Core i5-9500 CPU, DDR4 24GB memory. It means that the complexity of our proposal is acceptable and can be implemented on a personal computer easily. As depicted in Fig. 4.4, there is an increasing trend of accuracy along with the increment of SoSL. Accordingly, the model of SoSL-50 achieves poor discrimination in the first ten epochs compared to other models. In contrast, the model of SoSL-400 can discriminate the correct key from incorrect ones very early in Fig. 4.4c.



a) SoSL-50

b) SoSL-200

c) SoSL-400

d) Non-SoSL

Figure 4.4: The experimental results of MLP_{MOC} models on masking countermeasure using different SoSL on each branch.

to the original MLP$_{\text{DDLA}}$ model (hidden layer: 20x10-Relu, output layer: 2-Softmax).

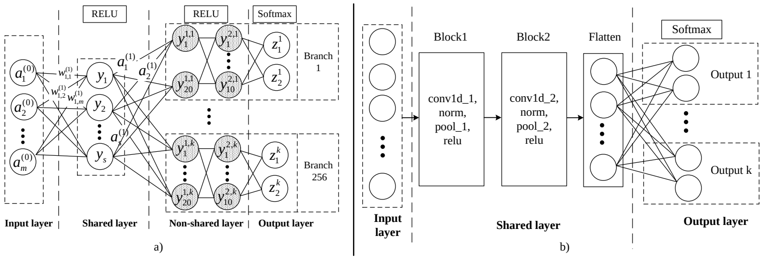


Figure 4.2: Structure of proposed multi-output classification neural network.

In terms of CNN architecture, the shared layer consists of two blocks. Each block includes a 1D convolutional (*conv1d*) layer, an average pooling (*pool*) layer, a batch normalization (*norm*) layer, and a rectified linear unit (*relu*) layer. These layers are placed in order as follows *conv1d-norm-pool-relu* as depicted in Fig. 4.2.b. In this case, the proposed models use the separate losses ($\mathcal{L}^{[k]}$) to discriminate the correct key and the total loss ($\mathcal{L}_{total}$) for update the weights as follows.

$$\mathcal{L}^{[k]}(\theta) = -\frac{1}{N_s}\sum_{j=1}^{2} y_{true}\ln(z) \tag{4.1}$$

where θ represents the set of all parameters of the model, y_{true} and z are the ground-truth and the predicted values, respectively. N_s denotes the number of training samples.

$$\mathcal{L}_{total} = \sum_{k=1}^{256} \gamma_k * \mathcal{L}^{[k]}(\theta) \tag{4.2}$$

γ_k is used as the weighted factor of branch number k^{th} and set as 1 for all branches (weights of each branch is equivalent).

4.4 Proposed multi-output regression neural networks

This section introduces MLP and CNN models based on MOR architectures. Similar to proposed MOC models, proposed MOR models consist of an input layer and shared layers, followed by the output layer divided into K branches.

Algorithm 1 Differential Deep Learning Analysis (DDLA)

```
Input: $D$ traces $(\boldsymbol{t}_i)_{1\le i\le D}$, corresponding plaintexts $(d_i)_{1\le i\le D}$, and $K$ key hypotheses. A network $Net$ and number of epochs $n_e$
Output: $k_{cr} \in K$
Set training data as $X = (\boldsymbol{t}_i)_{1\le i\le D}$.
for $k_j \in K$ do
    Re-initialize trainable parameters of $Net$
    Compute the series of hypothetical values $(h_{i,j})_{1\le i\le D}$
    Set training labels as $Y_i = (h_{i,j})_{1\le i\le D}$
    Perform Deep Learning training: $DL\,(Net, X, Y_i, n_e)$
end for
return key $k_{cr}$ which leads to the best $DL$ training metrics
```

evaluation, there exist today two popular security certification programs “*conformance-based*” and “*attack-based*” testing. The thesis focuses on attack-based testing (the red part as illustrated in Fig. 1.3)

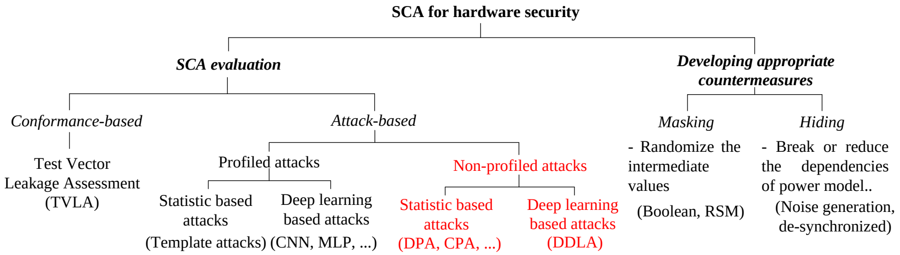

SCA for hardware security
SCA evaluation
Developing appropriate countermeasures
Conformance-based
Attack-based
Masking
Hiding
Test Vector Leakage Assessment (TVLA)
Profiled attacks
Non-profiled attacks
- Randomize the intermediate values
(Boolean, RSM)
- Break or reduce the dependencies of power model..
(Noise generation, de-synchronized)
Statistic based attacks
(Template attacks)
Deep learning based attacks
(CNN, MLP, ...)
Statistic based attacks
(DPA, CPA, ...)
Deep learning based attacks
(DDLA)

Figure 1.3: Classification of side-channel security evaluation

1.5 The related works and research directions

In terms of the statistic-based non-profiled SCA techniques, CPA is the most popular attack used in SCA evaluation. This technique has been used to break different block ciphers such as DES, AES, PRESENT, PICCOLO, and GIFT. In addition, it is also the most commonly used technique to clarify the efficiency of SCA countermeasures. Therefore, improving the efficiency of CPA has received significant interest in recent years. To enhance the performance of CPA attack, the signal-to-noise ratio (SNR) is usually exploited. Accordingly, a small set of power traces are extracted with a high signal-to-noise ratio (SNR) distributed in both tails of the distribution range. This method aims to enlarge the variance of the exploitable consumption component in the power trace. However, this method discards too many traces in the extractor and limits these techniques to scenarios where the attacker has a large number of traces to estimate the mean values. In the case of countermeasures added, a dedicated pre-processing technique is required for each specific attack in statistic-based attacks. It leads to a high-cost and time-consuming process of SCA evaluation. By applying DL techniques, pre-processing techniques are no longer required in DL-based attacks (DLSCA). However, research on non-profiled DLSCA is still quite a new field. The first non-profiled DLSCA, namely DDLA, was introduced by Timon at the TCHES conference in 2019. It has been demonstrated to break different SCA countermeasures, such as masking, de-synchronized, and noise generation. However, the DDLA technique is not optimized in terms of execution time because it requires a training process for each key hypothesis.

Most recently, Kwon *et al.* have introduced a parallel architecture[7] to mitigate the mentioned issue. Their proposed models can simultaneously predict a total of 256 hypothesis keys. Kwon's work can be considered a multi-label SCA approach. Despite being a very fast attack technique, the parallel architecture requires high memory usage. In addition, the success rate of attacks has not been investigated, especially in applying noise injection countermeasures.

In Viet Nam, there are only a few publications on the SCA domain, and most of them are published by the research teams from the Vietnam Academy of Cryptography Techniques, Vietnam National University, Hanoi, or Le Quy Don Technical University. In addition, their recent works focus only on the profiled attacks.

1.6 Summary

This chapter presents background knowledge of SCA data, the popular SCA methods, and the applications of SCA in the hardware security domain. In particular, it presents a comprehensive review of recent works on SCA attack methods as well as outlines some challenging issues that promote the contributions of this work.

[7] Kwon, D., Hong, S., & Kim, H. (2022). Optimizing Implementations of Non-Profiled Deep Learning-Based Side-Channel Attacks. IEEE Access, 10, 5957–5967

Table 4.1: The structure of reconstructed datasets.

Dataset	ASCAD		ChipWhisperer		CHES2018-CTF		Ground truth value
	No. of traces	Input	No. of traces	Input	No. of traces	Input	
Original	50000	700	10000	480	45000	2200	-
Dataset1	20000	700	-	-	-	-	LSB
Dataset2	20000	700	-	-	-	-	Identity
Dataset3	-	-	10000	480	-	-	LSB
Dataset4	-	-	10000	480	-	-	Identity
Dataset5	-	-	-	-	40000	2200	LSB
Dataset6	-	-	-	-	40000	2200	Identity

the Sbox function. Regarding the SHW label, it requires a different set of power traces corresponding to different key hypotheses. Therefore, SHW can not be applied in multi-output architecture. All multi-output datasets used in this chapter are constructed as depicted in Fig. 4.1.

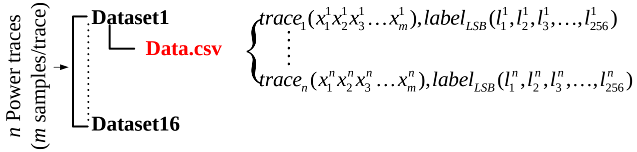

n Power traces
(*m* samples/trace)
Dataset1
Data.csv
Dataset16
$trace_1(x_1^1 x_2^1 x_3^1 \ldots x_m^1), label_{LSB}(l_1^1, l_2^1, l_3^1, \ldots, l_{256}^1)$
$trace_n(x_1^n x_2^n x_3^n \ldots x_m^n), label_{LSB}(l_1^n, l_2^n, l_3^n, \ldots, l_{256}^n)$

Figure 4.1: Structure of multi-output datasets used in this thesis.

4.3 Proposed multi-output classification neural networks

Overall, the proposed MOC networks have the same structure that consists of an input layer and shared layers followed by K branches of output layer corresponding to K hypothesis keys (for example, $K = 256$ in the case of 8-bit Sbox). The input layer of the proposed model has the same size as the number of samples in the power trace. The output consists of two nodes corresponding to the values of the LSB label. However, the structure of each branch is different and depends on which architecture is selected. In the case of MLP architecture, each branch contains an MLP architecture as same as MLP$_{\text{DDLA}}$ (except the input layer), as illustrated in Fig. 4.2.a. The number of layers and the number of nodes in each layer are similar

Chapter 4

MO-DLSCA: MULTI-OUTPUT DEEP LEARNING BASED NON-PROFILED SCA

4.1 Introduction

The results in the previous chapter show that DDLA is time-consuming since the requirement of repeating the training process. Most recently, the drawbacks of the original DDLA have been discussed and mitigated by Kwon *et al.* with parallel networks[1]. Their work is based on multi-label neural networks. Accordingly, they use a parallel architecture to predict a total of 256 hypothesis keys. The loss function used in the parallel network is the *binary cross-entropy*. The output layer consists of 256 nodes (corresponding to 256 key guesses). Their solution is using a custom function to calculate the accuracy of each key guess by separating the output and then matching the hypothesis values to count the number of corrected predictions. This method requires extra calculations to achieve the final results. Therefore, the attack time is not optimized. This chapter introduces new SCA techniques to deal with this issue based on multi-output multi-loss neural networks. Furthermore, the proposed techniques enhance the success rate of the attacks as well.

4.2 Data preparation

To apply a multi-output model in the SCA domain, the data input (power traces) should be labeled by the values corresponding to the model outputs. The proposed models aim to predict all key hypotheses in one training process. Therefore, the number of network outputs is K, which corresponds to K key guesses (0 to K). However, it is different from Kwon's proposals; the outputs of the proposed model are divided into K branches. In our case, the LSB labeling technique is selected as in previous works. Regarding the MOR model, the identity labeling technique is first selected to use in non-profiled DLSCA. In this case, the ID labels are the output values of

[1] Kwon, D., Hong, S., & Kim, H. (2022). Optimizing Implementations of Non-Profiled Deep Learning-Based Side-Channel Attacks. IEEE Access, 10, 5957–5967

Chapter 2

LOW COMPLEXITY CORRELATION POWER ANALYSIS ATTACKS

2.1 The complexity of CPA attacks

By using the secret key k_{cr} during the D (number of plaintexts) executions of the algorithm, the cryptographic device needs to calculate the intermediate values $\boldsymbol{v}_{cr} = f\left(d_i, k_{cr}\right)_{1\leq i\leq D}$. Therefore, the recorded traces depend on these intermediate values at same position. This position of the power traces, namely *"correct sample"*, is denoted as ct. By calculating hypothetical power consumption $\boldsymbol{h}_{cr} = HW/HD\left(v_{i,cr}\right)$, the correlation between $\boldsymbol{h}_{cr}$ and the column $\boldsymbol{t}_{ct}$ of power traces is the strongest. In fact, they lead to the highest value $r_{cr,ct}$ in $\boldsymbol{R}$. It is clear that the number of correlation coefficients of a CPA attack needs to calculate is the size of the matrix $\boldsymbol{R}$. Therefore, with K key hypotheses and the length S of the power traces, the complexity of CPA is proportional to $K * S$. As a result, the complexity of CPA depends on the length of the power traces.

2.2 Distribution of sampling correlation coefficients in CPA

The correlation coefficient ρ can be mapped to a random variable Z that has a normal distribution by applying Fisher's transformation, as in the equation (2.1). The mean of Z is then given by μ in (2.2) and the variance in (2.3).

$$R \mapsto Z = \frac{1}{2}\ln\frac{1+R}{1-R} \tag{2.1}$$

$$\mu = E(Z) = \frac{1}{2}\cdot\ln\frac{1+\rho}{1-\rho} \tag{2.2}$$

$$\sigma^2 = Var(Z) = \frac{1}{n-3} \tag{2.3}$$

In order to reveal the correct subkey k_{cr}, the number of power traces needs to be increased in an attack until a significant peak $\rho_{cr,ct}$ is visible in the correlation matrix $\boldsymbol{R}$, where $\rho_{cr,ct}$ denotes the correlation value that is

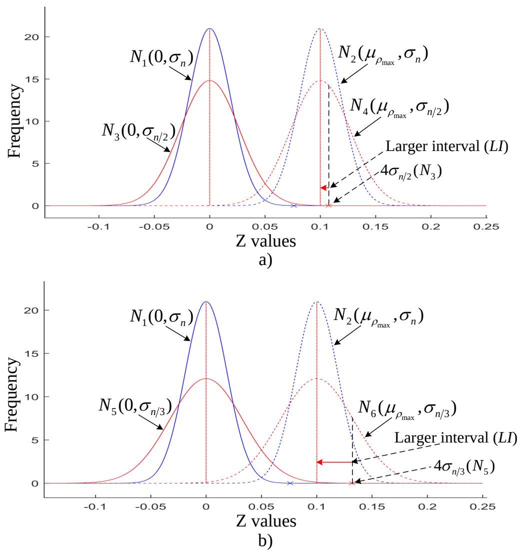


Figure 2.1: The probability density of Z values on different numbers of used power traces: a) n and $n/2$; b) n and $n/3$.

estimated by formula 1.1 at position t_{ct} of power traces and the intermediate value $\boldsymbol{h}_{cr}$ using the correct key k_{cr}. To take the peak, one usually assumes a very low correlation between correct and incorrect guesses. Under this assumption, which implies that the correlation coefficients ρ_k are almost null for every $k \neq k_{cr}$. Therefore, the peak is determined by the distance between the sampling distribution with $\rho = 0$ and $\rho = \rho_{\max}$.

2.3 Proposed improved correlation power analysis techniques

According to Mangard's work[1], the relationship between n (the minimum traces needed) and $\rho_{cr,ct}$ can be expressed as follows:

$$n \approx \frac{28}{\rho^2_{cr,ct}}. \tag{2.4}$$

In this case, $\rho_{cr,ct}$ equals to ρ_{max}. The authors also indicated that all the correlation coefficients are located in $4\sigma_n = \pm 4/\sqrt{n}$. If the number of traces

[1] Power Analysis Attacks- Revealing the Secrets of Smart Cards. Boston, MA: Springer US, 2007 [Online]

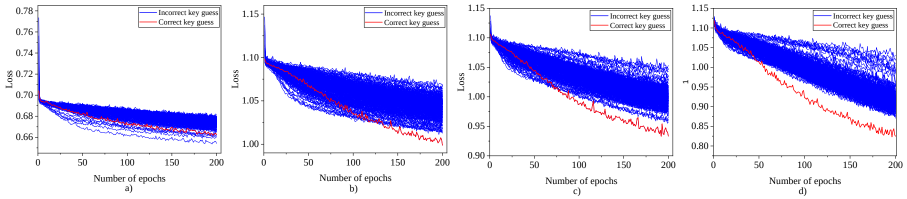


Figure 3.6: The attack results of CNN models using the LSB and SHW labeling technique on de-synchronized datasets with different numbers of filters. a) LSB label, 4 filters; b) SHW label, 4 filters; c) SHW label, 8 filters; d) SHW label, 16 filters;

Table 3.1: The execution time of DDLA attacks using different labeling techniques and architecture.

Target	Model	Label	No. of traces	No. of trace samples	No. of epochs	Execution time
CW no mask	MLP$_{\text{DDLA}}$	LSB	3000	10000	30	4 hrs 31 min 59 sec
	MLP$_{\text{proposed4}}$	SHW	3000	50	30	1 hr 39 min 30 sec
ASCAD masked	MLP$_{\text{DDLA}}$	LSB	20000	700	30	20 hrs 28 min 48 sec
	MLP$_{\text{DDLA}}$	SHW	20000	700	30	19 hrs 32 min 17 sec
CW-shifted	CNN$_{\text{LSB}}$	LSB	10000	480	200	10 hrs 31 min 12 sec
	CNN$_{\text{SHW}}$	SHW	10000	480	200	8 hrs 18 min 36 sec

training process (number of epochs, number of filters, activation, etc.), it can be seen that CNN$_{\text{LSB}}$ can not reveal the secret key, whereas the secret key can be clearly discriminated at epoch 150 by CNN$_{\text{SHW}}$. More interestingly, CNN$_{\text{SHW}}$ performs the attacks faster than CNN$_{\text{LSB}}$ by about 20% (8.31 hours compared to 10.52 hours as shown in Table 3.1) since the model using the SHW label reduces approximately 30% power traces needed for training. It is worth noting that by implementing various experiments, the results of CNN$_{\text{SHW}}$ mentioned above are achieved with a batch size equal to 200; other values provide poor results.

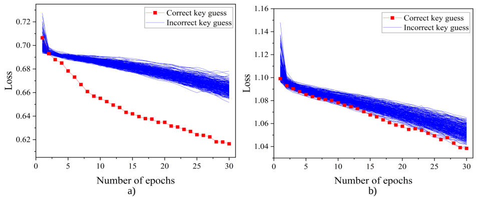


Figure 3.5: Experimental results on masked AES data using LSB and 3-HW labeling technique.

perfectly trained ($w_j \approx 0\,(j \neq l_1, l_2)$), the function $F\,(.)$ can be satisfied:

$$F\,(L\,(t_1)\,, L\,(t_2)) = |L\,(t_1) - L\,(t_2)| \tag{3.1}$$

in other words, the neural network can recombined the mask and the masked values following the *"absolute difference combining function*[1]*"* to perform a high-order attack (δ_1 be equal to δ_2).

b) Attack results

This part illustrates the effectiveness of the neural networks in performing the combining function. In this case, the HW model is used to estimate the relationship between the combining function and the leakage model. However, the proposed model uses the SHW label instead of using the HW label. The attack results are depicted in Fig. 3.5. It can be seen that the LSB label provides a clear distinction between correct and incorrect keys, whereas it is only a small gap between correct key and incorrect keys in the case of the SHW labeling method. It means that the correlation between the combined second-order leakage samples and the HW model is lower than the LSB model. However, this result indicates that SHW labeling is able to break a masking countermeasure. In addition, the proposed HW-based technique helps reduce the number of measurements for each training process by approximately 30% compared to the LSB labeling technique.

3.4.2 De-synchronized

The attack results of CNN$_{\text{LSB}}$ and CNN$_{\text{SHW}}$ using four filters are shown in Fig. 3.6.a and Fig. 3.6.b, respectively. With the same condition of the

[1] Prouff, E., Rivain, M., & Bévan, R. (2009). Statistical analysis of second order differential power analysis. IEEE Transactions on Computers, 58(6), 799–811

reduces from n to $n/2$, the standard deviation of the normal distribution $N_1(0, \sigma_n)$ will be changed from σ_n to $\sigma_{n/2}$. The shape of distribution $N_1(0, \sigma_n)$ will be changed to $N_3(0, \sigma_{n/2})$ as illustrated in Fig. 2.1.a. As a result, the correlation coefficients are located in new range $4\sigma_{n/2} = \frac{4\sqrt{2}}{\sqrt{n}}$. In addition, from Eqs. (2.4), we have $\rho_{\max} \approx \frac{\sqrt{28}}{\sqrt{n}}$. Therefore, it is easy to obtain:

$$4\sigma_{n/2} > \rho_{\max} \tag{2.5}$$

Similar result can be achieve, if the number of traces reduces from n to $n/3$. From Eqs. (2.5), it can be concluded that the correlation value ρ_{max} of $t_{cr,ct}$ is always small than the highest correlation value of the incorrect key when n reduces to $n/2$. Therefore, an interval (or group of samples) from the highest correlation value that contains the ρ_{max} can be taken. This interval is called "larger interval (LI)". As depicted in Fig. 2.1, by selecting a group of samples in the right tail of the incorrect key distribution, we can select the $t_{cr,ct}$ sample. The size of this group depends on the length of LI. The proposed technique is completed in Algorithm 2.

Algorithm 2 Auto-CPA based on partial correlation power analysis: P-CPA

```
Input: $trace^{n \times S}$, plaintext$^{n \times 16}$, attack byte $B$, $n' = 1/2n$, $\varepsilon = 100$
Output: $\boldsymbol{k}[B]$
$Plt_0$ = plaintext$^{n' \times 16}$
$Tr_0 = trace^{n' \times S}$
for $k$ from 0 to $K$ do
    $\boldsymbol{R} \leftarrow StandardCPA(Plt_0, Tr_0)$
end for
while $\varepsilon \leq 100$ do
    $\varepsilon = \varepsilon + 1$
    $\bar{S}[\varepsilon] = s_{max} \leftarrow argmax(\boldsymbol{R})$        ▷ $s_{max}$: the index of column containing the max value
    $s_{max} = 0$
end while
$Plt_1$ = plaintext$^{n \times 16}$
$Tr_1 = trace^{n \times \bar{S}}$        ▷ $\bar{S}$ has size of $(1 \times \varepsilon)$
for $k$ from 0 to $K$ do
    $\boldsymbol{R} \leftarrow StandardCPA(Plt_1, Tr_1)$
end for
$\boldsymbol{k}[B] = line_{max} \leftarrow argmax(\boldsymbol{R})$        ▷ $line_{max}$: the index of line containing the max value
```

Algorithm 3 Auto-CPA based on power trace biasing based partial correlation power analysis: BP-CPA

```
**Input:** $trace^{n\times S}$, plaintext$^{n\times 16}$, attack byte $B$, $n' = 0$, $\varepsilon = 250$
**Output:** $\boldsymbol{k}[B]$
$Plt_0 = \text{plaintext}^{n\times 16}$
**for** $k$ from 0 to $K$ **do**
    **for** i from 1 to $n$ **do**
        $h_{i,k} \leftarrow HW(SBOX((plaintext_{i,B}, k))$
        **if** $h_{i,k} = 0, 1, 2, 6, 7, 8$ **then**
            $n' = n' + 1$
            $trace_{n',S} = trace_{i,S}$
        **end if**
    **end for**
    $Tr_0 = trace^{n'\times S}$
    $\boldsymbol{R} \leftarrow StandardCPA(Plt_0, Tr_0)$
**end for**
**while** $\varepsilon \leq 250$ **do**
    $\varepsilon = \varepsilon + 1$
    $\bar{S}[\varepsilon] = s_{max} \leftarrow argmax(\boldsymbol{R})$        ▷ $s_{max}$: the index of column containing the max value
    $s_{max} = 0$
**end while**
$Plt_1 = \text{plaintext}^{n\times 16}$
$Tr_1 = trace^{n\times \bar{S}}$
**for** $k$ from 0 to $K$ **do**
    $\boldsymbol{R} \leftarrow StandardCPA(Plt_1, Tr_1)$
**end for**
$\boldsymbol{k}[B] = line_{max} \leftarrow argmax(\boldsymbol{R})$        ▷ $line_{max}$: the index of line containing the max value
```

In the case of n reduced to $n/3$, the LI is quite large, as illustrated in Fig. 2.1.b. It means that a huge number of POI need to be collected to ensure these samples contain t_{ct} sample. Our solution is to increase the value of μ_{max}. It will reduce the distance between μ_{max} and the 4σ position. In other words, the LI will be reduced and suitable to take the top-down correlation values as Algorithm 2. For each sample t_τ, the operation on all power traces is usually the same. Therefore, the variance of the operation-dependent power consumption $\sigma^2(t_o(\tau)) = 0$. Consequently, the SNR value can be simplified as:

$$SNR = \frac{\sigma^2(t_d(\tau))}{\sigma^2(t_{noise}(\tau))} \tag{2.6}$$

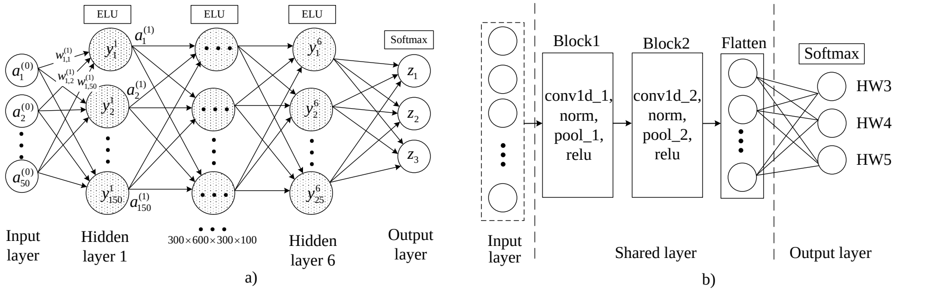


Figure 3.3: The proposed Multi-layer perceptron architecture.

In the case of CNN architecture, the proposed model is composed of an input layer and two 1-D convolutional (*Conv1d*) blocks in the middle, followed by the flattened layer and classification (output) layer. Each Conv1D block is formed by a Conv layer directly followed by a batch normalize (*norm*) layer and a pooling layer (*pool*) for selecting the informative and downsample the feature maps. The last layer of the *Conv1d* block is the activation layer, as described in Fig. 3.3.b. It is worth noting that the activation of the output layer is *Softmax* function. To achieve the model's output, the proposed models perform the same procedures. Firstly, a *forward propagation* procedure is performed to calculate the output (3 nodes) from the input layer. However, unlike in MLP models, where each neuron has a separate weight vector, neurons in CNN share the weights. Neurons perform convolutions on the data, with the convolution filter being formed by the weights. In our case, two *Conv1d* layers corresponding to two blocks are designed with the same number of filters (4, 8, and 16). Then, the *backward propagation* procedure is implemented to update the learning metrics. Concretely, to optimize the proposed models, SGDM and Adam optimizations are used for MLP and CNN models in training processes, respectively.

3.4 Validation experiments

3.4.1 Masking

a) Combining function in non-profiled DLSCA

In the case of masking countermeasure, every sensitive variable v is randomly split into d shared $m_1, \ldots, m_d$ in such a way that the relation $m_1 * m_2 \ldots * m_d = v$ is satisfied for a group operation $\circ$. If the model is

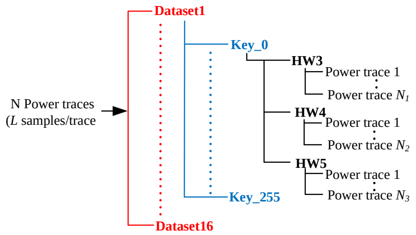


Figure 3.2: Structure of the new datasets: There are 16 folders (**Dataset1 to Dataset16**) corresponding to 16 bytes of the secret key, each folder contains 256 files in **.csv** format which correspond to 256 hypothesis keys. N original power traces (L samples/trace) are calculated to form N_1 new traces and labeled ($HW = \{3, 4, 5\}$). Each new trace contains 50 samples, which are the highest correlation values.

It means that if we use only three classes for training instead of nine classes, the DL model using the correct key still has better training metrics than the wrong key. As shown in Fig. 3.1, there are three significant HW values ($HW\ =\ 3,\ 4,\ 5$) that contain the most distribution of intermediate values. Moreover, the distribution of three significant HWs (SHW) is nearly balanced. Therefore, SHW is considered to use for classification in the non-profiled context.

To apply SHW in non-profiled DLSCA, new datasets are reconstructed as presented in Fig. 3.2.

3.3 Non-profiled DLSCA using significant HW labeling

This section introduces two models using SHW labels based on MLP and CNN architectures. In general, the proposed models have one input layer, shared layers, and an output layer. In which the input and output layers are the same for all proposed models. Concretely, the number of nodes in the input layer is assigned according to the number of samples in a power trace. The output layer of the proposed models consists of three nodes corresponding to three HW values. Regarding the shared layers, the proposed MLP network comprises six hidden layers. As depicted in Figure 3.3.a, all arrows represent the weights. Prior to the implementation training phase, the values of weights and bias are randomly chosen from a normal distribution using the Xavier scheme.

From Eqs. (2.6), it can be seen that SNR increase when the variance of t_d increase. To achieve higher SNR, our proposal is to select the power traces which correspond to the high-variance data input. Consequently, the plaintext corresponding to HW equal 1,2,3,6,7,8 will be selected, as illustrated in Fig. 3.1. The proposal is described in Algorithm 3. In this case, Algorithm 3 is different from Algorithm 2 in the process of phase 1, and the value of $\varepsilon = 250$. The proposal does further steps to take out $n' \approx \frac{n}{3}$ power traces based on the biasing technique (Step 6,7).

2.4 Validation experiments

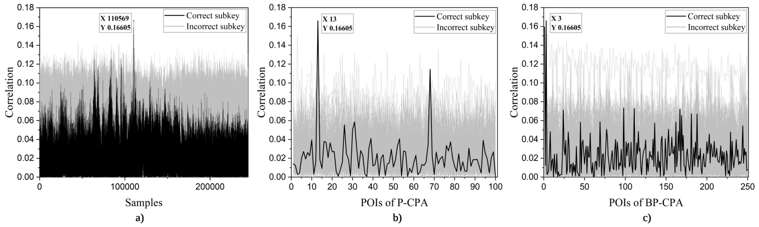


Figure 2.2: Attack results of standard CPA, P-CPA and BP-CPA methods on masking countermeasure: a) Standard CPA; b) P-CPA; c) BP-CPA.

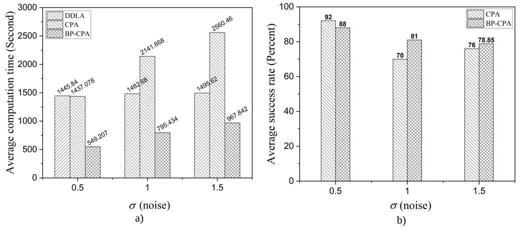


Figure 2.3: Average of computation time and success rate on different levels of Gaussian noise added ASCAD: a) Average of computation time; b) Average of success rate.

All experiments were performed with MATLAB software on a personal computer with an Intel Core i5-9500 CPU and DDR4 24GB memory. In the experiments, the average success rate and computation time are used as the metrics to evaluate the efficiency of the proposed methods.

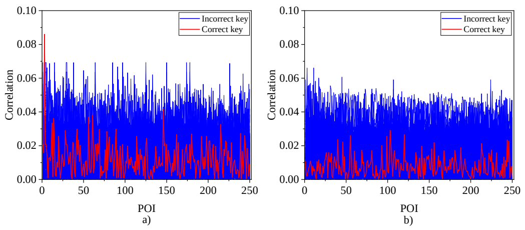


Figure 2.4: Experimental results of BP-CPA attack on de-synchronized countermeasure. a) Shifted value = 1; b) Shifted value = 5.

The results show that the proposed technique could select the POI as the analysis above. More importantly, the selected samples contain the correct sample t_{tc} as in conventional CPA ($\rho = 0.16605$), as depicted in Fig. 2.2. Since the POI is selected correctly, the execution time of CPA attacks reduces significantly (from 2560.46 seconds to 967.84 seconds, $\sigma = 1.5$). Consequently, the proposed techniques, such as P-CPA and BP-CPA, perform the attacks faster than conventional ones, as illustrated in Fig. 2.3.

In the case of hiding countermeasures, the random delay will cause a misaligned problem. Therefore, the POI extractor will not work correctly because it requires each operation of the cryptographic algorithm should be located at the same position in each power trace to find out the correlation, as depicted in Fig. 2.4.

2.5 Summary

This chapter introduces two improved CPA techniques based on the distribution of sampling correlation and biasing technique. The effectiveness of the proposed techniques has been clarified on different SCA countermeasures. However, the number of traces is limited because of the complexity of second-order leakage data processing. In addition, other power consumption models have not been investigated. The success rate of the proposed techniques is also increased compared to conventional CPA. However, BP-CPA can not break the de-synchronized countermeasure. This is also the solution to counteract BP-CPA on both hardware and software implementations of cryptographic devices.

Chapter 3

DIMENSIONALITY REDUCTION AND LABELING METHODS FOR EFFICIENT DEEP LEARNING BASED NON-PROFILED SCA

3.1 Reducing data dimension using P-CPA

In the previous chapter, the P-CPA technique is used to take the most relevant samples in the power trace by computing the correlation between real traces and their model. Additionally, P-CPA requires only 50% of given power traces for detecting the POI. Therefore, this method is suitable for power traces containing a large number of samples. Based on the advantages of P-CPA, this method is used to reduce the number of features of the data input in DDLA.

3.2 Significant HW Labeling

In the non-profiled context, HW model causes the imbalance dataset problem in the DLSCA. Indeed, by observing Fig. 3.1, it is obvious that the distribution of intermediate values on each HW is imbalanced and symmetric about HW4 in the case of 8-bit Sbox. To mitigate the imbalanced dataset issue in a non-profiled context, this section introduces a simple labeling technique based on the significant HW values. Accordingly, training with the correct key always has better learning ability than the incorrect ones.

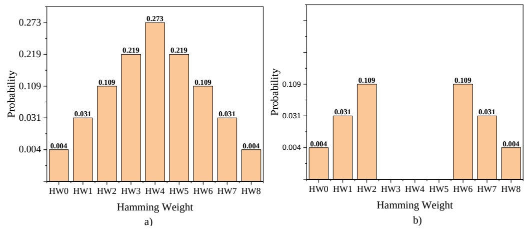


Figure 3.1: Probability distribution of the HW of a uniformly distributed 8-bit value. a) All HW; b) High variance HW.