Using More Points in One Clock Cycle to Achieve Better Performance of Template Attacks

Guangjun Fan[1], Yongbin Zhou[2], Hailong Zhang[2], Dengguo Feng[1]

[1] Trusted Computing and Information Assurance Laboratory,
Institute of Software, Chinese Academy of Sciences
guangjunfan@163.com, feng@tca.iscas.ac.cn
[2] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences
zhouyongbin@iie.ac.cn, zhanghailong@iie.ac.cn

Abstract. Template Attacks are widely accepted to be the most powerful side-channel attacks from an information theoretic point of view. For classical Template Attacks, several papers suggested that one should not choose more than one point as the interesting point per clock cycle when he conducts Template Attacks. Disobeying this constraint leads to poorer classification performance even if a higher number of interesting points is chosen. A more systematic approach, which relies on the data variability, is to choose the interesting points based on principal component analysis (PCA). In this paper, we present a new way of conducting Template Attacks when one uses more than one point as the interesting point per clock cycle. This new way has better classification performance compared with classical Template Attacks and PCA-based Template Attacks. Moreover, the computational price of the new way is low and practical. Therefore, we suggest that one should use this new way to better understand practical threats of Template Attacks when one want to use more than one point as the interesting point per clock cycle.

Keywords: Side-Channel Attacks, Power Analysis Attacks, Template Attacks.

1 Introduction

As an important method of Power Analysis Attacks, Template Attacks were firstly proposed by S. Chari et al. in 2002 [1]. Under the assumption that one has a reference device identical or similar to the target device, and thus be well capable of characterizing power leakages of the target device, Template Attacks are widely accepted to be the strongest side-channel attacks from an information theoretic point of view [1].

Principally, Template Attacks consist of two stages. The first stage is the profiling stage and the second stage is the extraction stage. In the profiling stage, one can accurately characterize signals and noises in different time samples and builds templates for each key-dependent operation with the reference device. In the extraction stage, one can exploit a small number of power traces measured from the target device and the templates to classify the correct (sub)key. We note that, Template Attacks are also important tools to evaluate the physical security of a cryptographic device.

Contributions Depending on the measurement setup and the data acquisition strategy, captured traces can be quite big (i.e. the number of sampled points is high). For Template Attacks to be practical, it is paramount that not all points of a trace are part of the templates. To reduce the number of points, one needs to choose some interesting points in traces. The interesting points are those points that contain the most information about the characterized key-dependent operation(s). For classical Template Attacks, the paper [2] suggested that the minimum distance between these points should be approximately a clock cycle or more. This constraint is used to avoid numerical problems when inverting the covariance matrix, since additional points in the same clock cycle *do not* provide additional information. Disobeying this constraint leads to poorer classification performance even if a higher number of interesting points is chosen [2]. Some other papers [3,4,15] also recommended to choose at most *one* interesting point per clock cycle. According to this guideline, the number of interesting points is rather limited and depends on the number of clock cycles which are correspond to the key-dependent operation.

A more systematic approach, which relies on the data variability, is to choose the interesting points based on principal component analysis (PCA). PCA-based Template Attacks were investigated in [3]. However, this kind of Template Attacks is inefficient [2] due to its high computational requirements and may not improve the classification performance [7]. Therefore, PCA-Based Template Attacks are not used widely in practice.

In this paper, we present a new way of conducting Template Attacks. In this new way, one also uses more than one point as the interesting point per clock cycle. The advantages of the new way are as follows:

– Using this new way, one can achieve better classification performance[1] compared with classical Template Attacks and PCA-based Template Attacks.
– The computational price of the new way is low and practical.

Therefore, we suggest that one should use this new way to better understand practical threats of Template Attacks when one want to use more than one point as the interesting point per clock cycle to conduct this kind of attacks.

Related Work Template Attacks were firstly introduced in [1]. The paper [2] provided answers to some basic and practical issues of Template Attacks, such as how to choose interesting points in an efficient way and how to preprocess noisy data. LDA-based Template Attacks were introduced in [12]. However, this kind of Template Attacks depends on the condition of equal covariances. Therefore, it is not a better choice compared with PCA-based Template Attacks in

[1] In this paper, we use success rate of attacks [6] as a metric about classification performance.

most settings [5] and we ignore this kind of attacks here. The paper [15] presented a variation of Template Attacks that can be applied to block ciphers when the plaintext and ciphertext used are unknown. In [8], Template Attacks were used to attack a masking protected implementation of a block cipher. In [9], an efficient leakage characterization method was introduced to efficiently characterize power leakages of the target device. Recently, a simple pre-processing technique of Template Attacks, normalizing the sample values using the means and variances was evaluated for various sizes of test data [7]. In [10], the assumption of Template based DPA was relaxed with machine learning techniques. Also, the paper [11] relaxed the assumption made in Template Attacks by using a method based on a semi-supervised learning strategy.

Organization of This Paper The rest of this paper is organized as follows. In section 2, we review classical Template Attacks as well as PCA-based Template Attacks. In section 3, we introduce and analyze our new way. The new way was verified by practical experiments which are introduced in section 4. In section 5, we conclude the whole paper.

2 Preliminaries

In this section, we briefly review classical Template Attacks and PCA-based Template Attacks.

2.1 Classical Template Attacks

We will introduce the two stages of classical Template Attacks in the following.

2.1.1 The Profiling Stage In the profiling stage, one has a reference device identical or similar to the target device. One can use power traces measured from the reference device to characterize power leakages of the target device.

Let us assume that there exist K different (sub)keys $key_i, i = 0, 1, \ldots, K-1$ which need to be classified. Also, there exist K different key-dependent operations O_i, $i = 0, 1, \ldots, K-1$. Usually, one will generate K templates, one for each key-dependent operation O_i. One can exploit advanced techniques [12, 13] to choose N interesting points $(P_1, P_2, \ldots, P_N)$. Each template is composed of a mean vector and a covariance matrix. Specifically, the mean vector is used to estimate the data-dependent portion of side-channel leakages. It is the average signal vector $M_i = (M_i[P_1], \ldots, M_i[P_N])$ for each one of the key-dependent operations. The covariance matrix is used to estimate the probability density of the noises at different interesting points. It is assumed that noises at different interesting points approximately follow the multivariate normal distribution. A N dimensional noise vector $n_i(S)$ is extracted from each trace $S = (S[P_1], \ldots, S[P_N])$ representing the template's key dependency O_i as $n_i(S) = (S[P_1] - M_i[P_1], \ldots, S[P_N] - M_i[P_N])$. One computes the $(N \times N)$ covariance matrix C_i from these noise vectors. The probability density of the noises occurring under key-dependent operation O_i is given by the N dimensional multivariate Gaussian distribution $p_i(\cdot)$, where the probability of observing a noise vector $n_i(S)$ is:

$$p_i(n_i(S)) = \frac{1}{\sqrt{(2\pi)^N |C_i|}} exp\Big(-\frac{1}{2} n_i(S) C_i^{-1} n_i(S)^T \Big) \quad n_i(S) \in \mathbb{R}^N. \qquad (1)$$

In equation (1), the symbol $|C_i|$ denotes the determinant of C_i and the symbol C_i^{-1} denotes its inverse.

2.1.2 The Extraction Stage

In the extraction stage, one tries to classify the correct (sub)key with a small number of traces obtained from the target device.

Assume one obtains t traces (denoted by $S_1, S_2, \ldots, S_t$) in the extraction stage. For example, when the traces are statistically independent, one will apply maximum likelihood approach on the product of conditional probabilities [14], i.e.

$$key_{ck} = argmax_{key_i} \Big\{ \prod_{j=1}^{t} \mathsf{Pr}(S_j | key_i), i = 0, 1, \ldots, K-1 \Big\},$$

where $\mathsf{Pr}(S_j|key_i) = p_{f(S_j,key_i)}(n_{f(S_j,key_i)}(S_j))$. The key_{ck} is considered to be the correct (sub)key. The output of the function $f(S_j, key_i)$ is the index of a key-dependent operation. For example, when the output of the first S-box (denoted by $Sbox$) in the first round of AES-128 is chosen as the target intermediate value, one builds templates for each output of the S-box. In this case, $f(S_j, key_i) = Sbox(m_j \oplus key_i)$, where m_j is the plaintext corresponding to the power trace S_j.

2.2 PCA-based Template Attacks

PCA-based Template Attacks [3] exploit a *continual* point fragment correspond to the target intermediate value in traces (We assume the length of the continual point fragment is N.). One first computes the empirical covariance matrix, which is given by

$$ECM = \frac{1}{K} \sum_{i=0}^{K-1} (M_i - \bar{M})(M_i - \bar{M})^T.$$

The quantity $\bar{M} = \Sigma_{i=0}^{K-1} M_i / K$ is the average of the mean vectors. Let us denote the matrixes of eigenvectors and eigenvalues of ECM by U and Δ, i.e.

$$ECM = U \Delta U^T.$$

The principal directions $\{w_i\}_{i=1}^{L}$ are the columns of U that correspond to the L largest eigenvalues of Δ. The corresponding matrix of principal directions is denoted $W \in \mathbb{R}^{N \times L}$. One uses projected mean vectors $\{W^T M_i^T\}_{i=0}^{K-1}$ and projected covariance matrices $\{W^T C_i W\}_{i=0}^{K-1}$ to conduct the attacks. Specifically speaking, the probability of observing a noise vector when one assumes the key-dependent operation is O_i is computed by

$$p_i(n_i(S)) = \frac{exp(-\frac{1}{2}n_i(S)W(W^T C_i W)_i^{-1}(n_i(S)W)^T)}{\sqrt{(2\pi)^L |W^T C_i W|}} \quad n_i(S) \in \mathbb{R}^N. \qquad (2)$$

One classifies the correct (sub)key based on the probability computed by equation (2).

3 Our New Way

In this section, for the purposes of comparison, we will introduce three different strategies to conduct Template Attacks. The first strategy and the second strategy are the classical way of conducting Template Attacks but with different number of interesting points per clock cycle. The third strategy is our new way of conducting Template Attacks. Finally, we evaluate the computational prices of the second strategy, the third strategy, and PCA-based Template Attacks.

Assume that, in one trace, there is a *continual* point fragment $(P_0, P_1, \ldots, P_{N-1})$ which is correspond to the key-dependent operation and has length N. We also assume that these N points are in c continual clock cycles. Therefore, there are N/c points per clock cycle. Let the symbol $P_{(i,j)}$ denotes the j^{th}, $j \in \{1, \ldots, N/c\}$ interesting point in the i^{th}, $i \in \{1, \ldots, c\}$ clock cycle.

Note that, there are many methods about how to choose interesting points. For example, difference of means based method [1], sum of squared differences based method [10], Signal to Noise Ratio based method [14], SOST [10], and DPA based method [14] etc. The DPA based method is considered to be the most efficient method to choose interesting points for classical Template Attacks. However, in this paper, we do not investigate the question about how to choose a point as the interesting point. In other words, we assume one can choose interesting points efficiently and effectively.

3.1 Strategy 1

In this strategy, one uses only one point as the interesting point per clock cycle and chooses c points

$$\left\{P_{(1,1)}, P_{(2,1)}, \ldots, P_{(c,1)}\right\}$$

from the N continual points as the c interesting points. Then, one conducts classical Template Attacks with templates which are built with the c interesting points. We call the attack with this strategy as "ATTACK-1".

3.2 Strategy 2

In this strategy, one uses more than one point as the interesting point per clock cycle. In order to show this strategy more clearly, we take the simplest case as an example, i.e. we assume that one uses two points as the interesting point per clock cycle. Therefore, $2c$ points are chosen from the N continual points as the interesting points:

$$\{(P_{(1,1)}, P_{(1,2)}), (P_{(2,1)}, P_{(2,2)}), \ldots, (P_{(c,1)}, P_{(c,2)})\}.$$

Then, one conducts classical Template Attacks with templates which are built with the $2c$ interesting points. This means that one needs to compute a $(1 \times 2c)$ mean vector and a $(2c \times 2c)$ covariance matrix for each template. We call the attack with Strategy 2 as "ATTACK-2".

Note that, the success rate of Strategy 2 will reduce when one uses more points as the interesting point per clock cycle to conduct Template Attacks [5]. Our experiments in the next section also verified this fact.

3.3 Strategy 3 (Our New Way)

Strategy 3 is our new way of conducting Template Attacks. In our new way, during the profiling stage, one uses more than one point as the interesting point per clock cycle. In order to show our new way more clearly, we also take the simplest case as an example, i.e. we assume that one uses two points as the interesting points per clock cycle. Therefore, $2c$ points are chosen from the N continual points as the interesting points:

$$\{(P_{(1,1)}, P_{(1,2)}), (P_{(2,1)}, P_{(2,2)}), \ldots, (P_{(c,1)}, P_{(c,2)})\}.$$

One divides the $2c$ interesting points into two sets. In the first set, there are c interesting points:

$$\mathsf{Set1} = \{P_{(1,1)}, P_{(2,1)}, \ldots, P_{(c,1)}\}.$$

The rest c interesting points are in the second set:

$$\mathsf{Set2} = \{P_{(1,2)}, P_{(2,2)}, \ldots, P_{(c,2)}\}.$$

Note that, in each set, any two points of the c interesting points are not in the same clock cycle. But the two points $(P_{(i,1)}, P_{(i,2)}),\ i = 1, 2, \ldots, c$ are in the same clock cycle and contain very similar information. In the following, one builds templates in the same way as classical Template Attacks with the c interesting points in $\mathsf{Set1}$ and obtains a group of templates denoted by $\mathsf{G1}$. Similarly, with the same traces used for obtaining $\mathsf{G1}$, one builds templates with the c interesting points in $\mathsf{Set2}$ and obtains another group of templates $\mathsf{G2}$. At this point, the profiling stage is finished.

In the extraction stage, one first computes a sequence

$$\{\mathsf{Pr}(1,0), \mathsf{Pr}(1,1), \ldots, \mathsf{Pr}(1, K-1)\}$$

using $\mathsf{G1}$ with some traces obtained from the target device in the same way as classical Template Attacks. The value $\mathsf{Pr}(1,i)$ represents the probability of the i^{th} (sub)key is the correct (sub)key (In the example of section 2.2, $\mathsf{Pr}(1,i)$ equals to

$\prod_{j=1}^{t} \mathsf{Pr}(S_j|key_i).$). Then, one sorts the sequence $\{\mathsf{Pr}(1,0), \mathsf{Pr}(1,2), \ldots, \mathsf{Pr}(1, K-1)\}$ in descending order and computes $\mathsf{Index}(1,i),\ i = 0, 1, \ldots, K-1$ for each (sub)key. The value $\mathsf{Index}(1,i)$ represents the sequence number of $\mathsf{Pr}(1,i)$ in the sorted sequence. Similarly, one computes another sequence

$$\big\{\mathsf{Pr}(2,0), \mathsf{Pr}(2,1), \ldots, \mathsf{Pr}(2, K-1)\big\}$$

using $\mathsf{G2}$ with the same traces obtained from the target device. Then, he computes $\mathsf{Index}(2,i),\ i = 0, 1, \ldots, K-1$ for each (sub)key. The candidate value of the correct key is computed by

$$key_{ck} = argmin_i\big\{\mathsf{Index}(1,i) + \mathsf{Index}(2,i), i \in \{0, 1, \ldots, K-1\}\big\}.$$

We call the attack with our new way as “ATTACK-3”.

Discussions In our new way, we do not build templates with interesting points in the same clock cycle simultaneously. Therefore, the new way avoids the numerical problems when inverting the covariance matrix. The new way will has higher success rate of attacks because it exploits information from more points in a trace in spite of the points in the same clock cycle provide very *similar* information. The paper [2] claims that *“additional points in the same clock cycle do not provide additional information”*. Our experiments in the next section show that the claim of that paper are not correct. If additional points in the same clock cycle do not provide additional information, the success rate of our new way should approximately equal to that of ATTACK-1. However, evaluation results show that the success rate of our new way is higher than that of ATTACK-1.

There are two generalizations about our new way of conducting Template Attacks. The first generalization is to use more points as the interesting point per clock cycle to conduct our new way of Template Attacks. Assume that one uses s $(2 < s \leq N/c)$ points as the interesting point per clock cycle, he will divide the cs interesting points into s sets and build s groups of templates in the profiling stage similarly to the way introduced above. In the extraction stage, one classifies the correct (sub)key by computing

$$key_{ck} = argmin_i\big\{\mathsf{Index}(1,i) + \ldots + \mathsf{Index}(s,i), i \in \{0, 1, \ldots, K-1\}\big\}.$$

Note that, if one uses all the points per clock cycle, he will not need to choose special points as the interesting point.

The second generalization is as follows. One can use s $(2 \leq s \leq N/c)$ points as the interesting point per clock cycle. For a fixed number of points used as the interesting point per clock cycle, one uses more than s sets of points to build templates and conducts our new way similarly as long as any two points in each set are not in the same clock cycle.

The success rate of our new way will be higher when the two generalizations are used but the computational price will also be higher. In this paper, we only consider the first generalization and do not further consider the second generalization.

3.4 Computational Price

We evaluate the computational prices of ATTACK-2, ATTACK-3, and PCA-based Template Attacks (short for PCA-TA) in this subsection. For Template Attacks, the computational price mainly depends on the size of the mean vector and the covariance matrix. Therefore, we compare the sizes of the mean vectors and the covariance matrixes to show the computational prices of the three attacks. We respectively show the sizes of the mean vectors and the covariance matrixes of the profiling stage and the extraction stage for the three attacks in Table 1 and Table 2. For ATTACK-2 and ATTACK-3, we assume one uses s ($2 \leq s \leq N/c$) points as the interesting point per clock cycle. Hence, for ATTACK-3, one needs to compute s $(1\times c)$ mean vectors and s $(c\times c)$ covariance matrixes.

From Table 1 and Table 2, we find that the computational price of our new way (ATTACK-3) is much lower than that of classical Template Attacks (Attack-2) both in the profiling stage and the extraction stage, especially when the value of c is large. The reason is that the size of the covariance matrix grows quadratically with the number of interesting points. In the profiling stage, the computational price of our new way is much lower than that of PCA-based Template Attacks. But the computational price of our new way is higher than that of PCA-based Template Attacks in the extraction stage. To sum up, the global computational price of our new way is low and practical.

Table 1. The Sizes of Mean Vectors And Covariance Matrixes of The Profiling Stage

	the sizes of mean vectors	the sizes of covariance matrixes
ATTACK-2	$1 \times sc$	$sc \times sc$
ATTACK-3	$s \times (1 \times c)$	$s \times (c \times c)$
PCA-TA	$1 \times N$	$N \times N$

Table 2. The Sizes of Mean Vectors And Covariance Matrixes of The Extraction Stage

	the sizes of mean vectors	the sizes of covariance matrixes
ATTACK-2	$1 \times sc$	$sc \times sc$
ATTACK-3	$s \times (1 \times c)$	$s \times (c \times c)$
PCA-TA	$1 \times L$	$L \times L$

4 Experiments

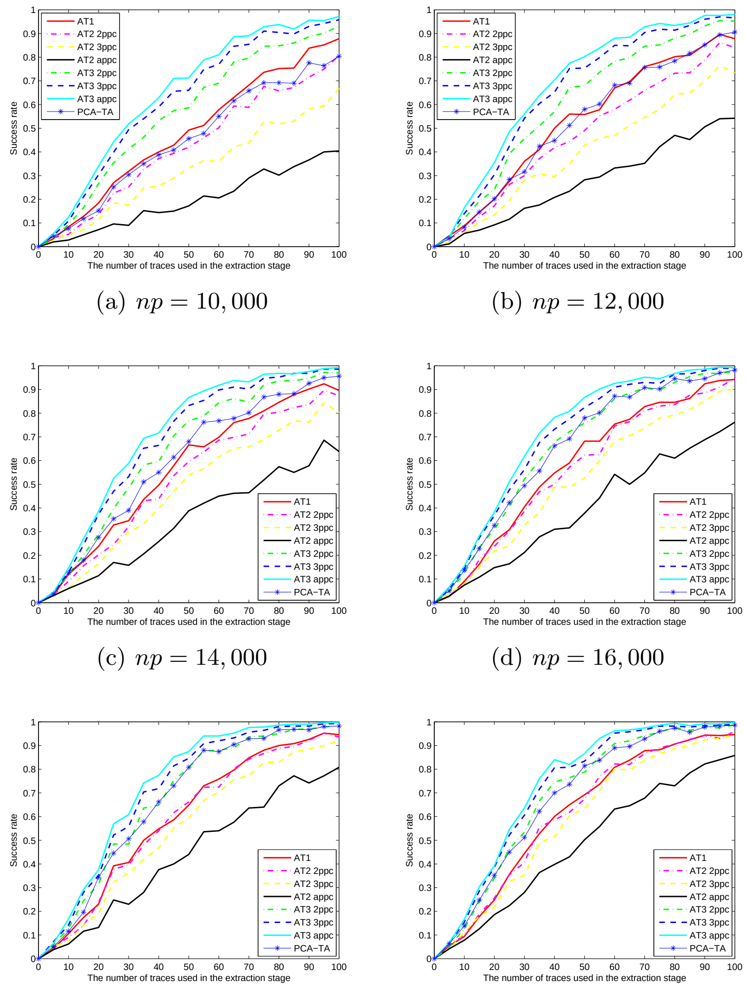
In this section, we experimentally evaluate the three strategies introduced in Section 3 as well as PCA-based Template Attacks. We tried to attack the output of the first S-box in the first round of unprotected AES-128 software implementation on an typical 8-bit microcontroller STC89C58RD+ whose operating frequency is 11MHz as an example. The real power traces were acquired with a sampling rate of 50MS/s. The average number of real power traces during the sampling process was 10 times. In all the practical experiments, we chose the continual point fragment using classical DPA based method [14]. The correlation coefficient of each point in the continual point fragment was larger than 0.7. We chose the points with high correlation coefficient as the interesting point per clock cycle.

Let "AT1" denote Attack-1. Let "AT2 2ppc" and "AT3 2ppc" denote case of ATTACK-2 and ATTACK-3 using the same two points as the interesting point per clock cycle. Let "AT2 3ppc" and "AT3 3ppc" denote case of ATTACK-2 and ATTACK-3 using the same three points as the interesting point per clock cycle. Let "AT2 appc" and "AT3 appc" denote case of ATTACK-2 and ATTACK-3 using all the points as the interesting point per clock cycle. Let "PCA-TA" denote PCA-based Template Attacks using six principal directions[1] (i.e. $L = 6$ Please see Section 2.2 for more details.). We conducted the eight attacks with same traces both in the profiling stage and the extraction stage. For simplicity, let np denote the number of traces used in the profiling stage and let ne denote the number of traces used in the extraction stage.

We used 10,000, 12,000, 14,000, 16,000, 18,000, and 20,000 traces to build the 256 templates respectively. The traces were generated with a fixed main key and random plaintext inputs. We generated additional 20,000 traces with another fixed main key and random plaintext inputs. The 20,000 traces were used in the extraction stage. We tested the success rates of the eight attacks when one uses ne traces in the extraction stage as follows. We repeated the eight attacks 500 times. For each time, we chose ne traces from the 20,000 traces uniformly at random and the eight attacks were conducted with the same ne traces. We respectively recorded how many times the eight attacks can successfully recover the correct subkey. The success rates of the eight attacks are shown in Figure 1.

From Figure 1, we can see that the success rate of ATTACK-3 is higher than those of ATTACK-1, ATTACK-2, and PCA-based Template Attacks. Therefore, we prove that PCA-based Template Attacks is not optimal. We also note that, when more points are used as the interesting point per clock cycle, the success rate of our new way will be higher while the success rate of classical Template Attacks will be lower. One cannot expect that the success rate of ATTACK-2 will be much higher than those of ATTACK-1 and PCA-based Template Attacks. The reasons are as follows. First, ATTACK-2 only uses very similar information from the additional points in the same clock cycle. Second, essentially, our new way does not depend on more advanced method of information extraction compared with classical Template Attacks.

[1] For our device, six principal directions are sufficient to ensure the success rate of PCA-based Template Attacks. The next few principal directions only *slightly* increase the power of this kind of attacks.



(a) $np = 10,000$ (b) $np = 12,000$

(c) $np = 14,000$ (d) $np = 16,000$

(e) $np = 18,000$ (f) $np = 20,000$

Fig. 1. The Practical Experiments Results

For other S-boxes in the first round of the unprotected AES-128 software implementation, similar evaluation results were obtained by us. These evaluation results show that one can not ignore the additional information provided by the additional points in the same clock cycle because the additional information can also be exploited to achieve better classification performance of Template Attacks.

5 Conclusion and Future Work

In this paper, we introduce a new way of conducting Template Attack when one want to use more than one point as the interesting point per clock cycle. This new way achieves better classification performance compared with classical Template Attacks and PCA-based Template Attacks. Moreover, the computational price of the new way is low and practical. Therefore, we suggest that one should use this new way to better understand practical threats of Template Attacks when one want to use more than one point as the interesting point per clock cycle to conduct this kind of attacks. In the future, it would be interesting to find quantitative factors about why classical Template Attacks have poorer classification performance when one uses more than one point as the interesting point per clock cycle. It is also very necessary to further verify our new way in other devices such as FPGA and ASIC.

References

[1] Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. CHES2002, LNCS 2523, pp.13-28, 2003.

[2] Rechberger, C., Oswald, E.: Practical Template Attacks. WISA2004, LNCS 3325, pp.440-456, 2004.

[3] Archambeau, C., Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Template Attacks in Principal Subspaces. CHES2006, LNCS 4249, pp.1-14, 2006.

[4] Bär, M., Drexler, H., Pulkus, J.: Improved Template Attacks. COSADE2010, 2010.

[5] Choudary, O., Kuhn, M.G.: Efficient Template Attacks. CARDIS2013, 2013.

[6] Standaert, F.-X., Malkin, T.G., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. EUROCRYPT2009, LNCS 5479, pp.443-461, 2009.

[7] Montminy, D.P., Baldwin, R.O., Temple, M.A., Laspe, E.D.: Improving cross-device attacks using zero-mean unit-variance mormalization. Journal of Cryptographic Engineering, Volume 3, Issue 2, pp.99-110, June 2013.

[8] Oswald, E., Mangard, S.: Template Attacks on Masking—Resistance Is Futile. CT-RSA2007, LNCS 4377, pp.243-256, 2007.

[9] Zhang, H., Zhou, Y., Feng, D.: An Efficient Leakage Characterization Method for Profiled Power Analysis Attacks. ICISC2011, LNCS 7259, pp.61-73, 2011.

[10] Lerman, L., Bontempi, G., Markowitch, O.: Side Channel Attack: An Approach Based On Machine Learning. COSADE2011, pp.29-41, 2011.

[11] Lerman, L. Medeiros, S.F., Veshchikov, N., Meuter, C., Bontempi, G., Markowitch, O.: Semi-Supervised Template Attack. COSADE2013, LNCS 7864, pp.184-199, 2013.
[12] Standaert, F.-X., Archambeau, C.: Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. CHES2008, LNCS 5154, pp.411-425, 2008.
[13] Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. Stochastic Methods A Performance Analysis for Side Channel Cryptanalysis. CHES2006, LNCS4249, pp.15-29, 2006.
[14] Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer 2007.
[15] Hanley, N., Tunstall, M., Marnane, W.P.: Unknown Plaintext Template Attacks. WISA2009, LNCS 5932, pp.148-162, 2009.