Wavelet Support Vector Machine Algorithm in Power Analysis Attacks

Shourong HOU, Yujie ZHOU, Hongming LIU, Nianhao ZHU

Dept. of Electronic Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, China

ruihou@sjtu.edu.cn, mlscagroup@163.com

Submitted April 24, 2017 / Accepted July 15, 2017

Abstract. *Template attacks and machine learning are two powerful methods in the field of side channel attacks. In this paper, we aimed to contribute to the novel application of support vector machine (SVM) algorithm in power analysis attacks. Especially, wavelet SVM can approximate arbitrary nonlinear functions due to the multidimensional analysis of wavelet functions and the generalization of SVM. Three independent datasets were selected to compare the performance of template attacks and SVM based on various kernels. The results indicated that wavelet SVM successfully recovered the offset value of the masked AES implementation for each trace, which was obviously 5 to 8 percentage points higher than SVM-RBF. And also, the time required was almost reduced by 40% when using the optimal parameters of wavelet SVM. Moreover, wavelet SVM only required an average of 5.4 traces to break the secret key for the unmasked AES implementation and less than 7 traces for the masked AES implementation.*

Keywords

Power analysis attacks, template attacks, support vector machine, wavelet analysis, kernel function

1. Introduction

Modern cryptographic devices generally implement the cryptographic algorithm and store the corresponding secret key. The cryptographic device must remain the secret key regardless of whether the algorithm itself is public or not. Therefore, it's important that the cryptographic algorithm does not reveal key-related information in the process of execution. Unfortunately, none of the cryptographic devices can eliminate the relevant information about the secret key from various side channels. An attacker may invade the entire security system and break the key through side channel information, which is called side channel attacks (SCAs). Typical SCAs contain power analysis attacks [1], timing attacks [2], acoustic cryptanalysis key extraction attacks [3], electromagnetic attacks [4], and their combinations [5]. In these attacks, power analysis attacks have attracted the attention of industry and academia.

Power analysis attacks were first proposed by Kocher et al. [1], which exposed the fact that the instantaneous power consumption of a cryptographic device depends on the data being processed and operations being performed. Many power analysis attacks methods have sprung up since then, such as Differential Power Analysis (DPA) [1], Template Attacks (TA) [6], [7], Correlation Power Analysis (CPA) [8], Mutual Information Analysis (MIA) [9], and Stochastic Model based Power Analysis (SMPA) [10]. From the viewpoint of engineering, power analysis attacks include two types, namely profiling and non-profiling attacks. For profiling attacks, a basic assumption is that the adversary is free to access target devices and profiling devices. Profiling attacks consist of two phases, called profiling and attacking. During the profiling phase, the adversary analyzes the profiling device by multiple power traces so that the key of the target device can be recovered when the attacking phase is performed. However, non-profiling attacks are single-step attacks that are performed directly on the target device. Over the past decade, various countermeasures against power analysis attacks have been proposed in hardware or software implementations. In general, almost all strategies are divided into two categories, hiding and masking. The masking scheme is very popular due to low cost and high performance. A comprehensive summary of power analysis attacks and countermeasures can refer to the book [11].

Rivest [12] first studied the intersection of machine learning and cryptography. Machine learning is a discipline whose purpose is to build a probability model based on the given data to predict the final result, which includes unsupervised learning, supervised learning and reinforcement learning. Roughly, the purpose of supervised learning is to learn a general function that maps the input space to the desired output space. The application of machine learning algorithms in power analysis attacks has just begun in recent years. Heyszlet et al. [13] used the k-means algorithm to attack the public key cryptosystem. Martinasek, Z. et al. [14] presented power analysis attacks based on multi-layer perceptron (MLP), and the authors [15] improved the MLP approach. The unsupervised clustering algorithm was proposed by Whitnall et al. [16], which was used to build the power consumption leakage model. Zhang et al. [17] researched

DPA attack based on genetic algorithm (GA). In work [18], the generalized CPA based on the k-Nearest Neighbors (k-NN) algorithm was briefly mentioned. Later, Martinasek, Z. et al. [19] improved power analysis attacks based on k-NN.

Hospodar et al. [20], [21] first applied Least Squares SVM (LS-SVM) in power analysis attacks. Although no real attack is performed, it provides a novel perspective on how LS-SVM is used for power analysis attacks. He et al. [22] presented that SVM-based attack recovered the entire secret key of DES performed on an 8-bit Atmel smartcard in a Stanford course project. Profiling attacks based on machine learning algorithms were introduced by Lerman et al. [23], [24]. They first compared TA and learning algorithms, namely Random Forest (RF), SVM, and Self-Organizing Maps (SOM), and then proposed an enhanced brute force algorithm to break the key. Heuser et al. [25] analyzed multiple bits of the key based on the Hamming weight model by using multi-class SVM. The authors divided the intermediate power consumption into several classes and then calculated the probability of belonging to each class. Finally, they adopted the maximum likelihood method to get the correct key. Their results have demonstrated that the performance of SVM is more stable than TA. This probabilistic multi-class SVM approach was later improved by Bartkewitz et al. [26]. More precisely, the authors assumed that the side channel leakage information could split interesting points into a strict order. Lerman et al. [27] presented the attack based on machine learning algorithm against the masked AES implementation. The results declared that SVM required 26 power traces to recover the key and had smaller computational complexity than TA. Hence, it was assumed that sample points of traces may not follow multivariate Gaussian distribution [28].

SVM is the most popular machine learning algorithm in power analysis attacks, while other algorithms have been proved to be feasible [14–19]. The previous work [23–27] focused more on how SVM translates a problem of breaking the key into the classification of machine learning. There is no systematic literature to study the elements that influence the performance of SVM in power analysis attacks. The kernel function (kernel method, kernel trick), hyperparameters (penalty factor, gamma of RBF kernel), feature selection, and other elements have significant effects on the performance of SVM. Wavelet analysis is a powerful tool for signal processing, which is often used to approximate the target function [29]. Although wavelet analysis has been used to process noisy power traces [30–32], the combined effects of wavelet analysis and power analysis attacks through kernel functions have been not yet explored. In order to enhance the sparsity of wavelet approximation and the generalization of SVM, Zhang et al. [33] first proposed a variant SVM algorithm based on wavelet kernel, known as wavelet SVM. It has been widely applied in financial, medical, industrial control, computer vision and other fields.

This paper aims to explore the application of wavelet SVM in power analysis attacks. Our attacks were implemented on three public datasets, including the offset recovery phase and the key recovery phase. We selected the success rate as a measure of the offset recovery phase and the guessing entropy as a metric of recovering the secret key. Furthermore, the results indicate that wavelet SVM is one of the most effective and efficient algorithms in power analysis attacks. This paper attempts to answer the following questions:

- Is wavelet SVM more suitable for power analysis attacks than TA and SVM based on other kernels?
- What is the impact of the optimal parameters of SVM on the classification results?
- What are the effects of the number of power traces (or the number of interesting points) on the performance of SVM based on various kernels?

2. Background

This section provides all the necessary knowledge about the principle and fast implementation of SVM, followed by the brief introduction to TA.

2.1 Binary-Class SVM

Cortes and Vapnik [34] proposed SVM in 1995 to address the binary classification with high generalization. The most basic model of SVM is a linear binary classifier, which aims to determine the separating hyperplane between two classes. Let

$$D_M = \{(\mathbf{X}_i, y_i) \,|\, \mathbf{X}_i \in R^N, y_i \in \{-1, +1\}, i = 1, 2, ..., M\} \quad (1)$$

represent a training set, where $\mathbf{X}_i$ is a training vector in the feature space, and y_i is the class label of $\mathbf{X}_i$. The separating hyperplane is:

$$f(\mathbf{X}) = \omega^T \phi(\mathbf{X}) + b \quad (2)$$

where $\omega \in R^N$, $b \in R$. By the nonlinear mapping function $\phi(\cdot)$, $\mathbf{X}$ is mapped into a feature space. There are many possible hyperplanes for an SVM classifier. A reasonable choice for the optimal hyperplane is to find the maximum separation (margin) between two classes. Accordingly, the maximum margin classifier can be rewritten as a constrained optimization problem:

$$\begin{gathered} \min_{\omega,b,\xi} \left(\frac{1}{2}\|\omega\|^2 + C \sum_{i=1}^{M} \xi_i \right), \\ \text{s.t. } y_i(\omega^T \phi(\mathbf{X}_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, 3, ..., M \end{gathered} \quad (3)$$

where ξ_i is the training error for vector $\mathbf{X}_i$, and C>0 is the regularization parameter which determines a trade-off between training error and margin size, also known as penalty factor.

By introducing the Lagrange multiplier, the optimization problem with constraints in (3) is simplified a dual problem:

$$\begin{gathered} \min_{\alpha} \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j K(\mathbf{X}_i, \mathbf{X}_j) - \sum_{i=1}^{M} \alpha_i, \\ \text{s.t. } \sum_{i=1}^{M} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, ..., M \end{gathered} \quad (4)$$

where the kernel function is $K(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i)^T \phi(\mathbf{X}_j)$, and α_i are Lagrange multipliers. The optimal solution to this problem is $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_M^*)^T$. The ω^* is given as follows:

$$\omega^* = \sum_{i=1}^{M} \alpha_i^* y_i \phi(\mathbf{X}_i) \tag{5}$$

and then a Lagrange multiplier α_j^* that satisfies $0 < \alpha_j^* < C$ is chosen to calculate:

$$b^* = y_j - \sum_{i=1}^{M} \alpha_i{}^* y_i K(\mathbf{X}_i, \mathbf{X}_j) \tag{6}$$

where $K(\mathbf{X}_i, \mathbf{X}_j)$ is the kernel function. Finally, the decision function of hyperplane is [35]:

$$f(\mathbf{X}) = \text{sign}\left(\sum_{i=1}^{M} \alpha_i^* y_i K(\mathbf{X}_i, \mathbf{X}) + b^* \right). \tag{7}$$

2.2 Multi-Class SVM

By adding several extensions, the binary-class SVM can be used to construct multi-class SVM. The mainstream strategies include one-against-one [36], one-against-all [37], directed acyclic graphs (DAG) [38], and error correction output coding [38]. For the sake of training time and accuracy, the machine learning community adopts the one-against-one strategy [39] to train a binary-class SVM classifier for each pair of possible classes. That is, for L classes, $(L-1)L/2$ binary-class SVM classifiers are required to be trained. The prediction results of all binary-class SVM classifiers are combined into the multi-class SVM classifier output, and then the class with the most votes is chosen. For more details, please refer to [36], [39].

2.3 Probabilistic SVM

In order to obtain the maximum likelihood estimate in the key recovery phase, the probability output of the class label c is necessary. Considering the sparsity of SVM, the logistic sigmoid function is usually used to approximate the outputs $y(\mathbf{X}_i)$ of all binary-class SVM classifiers [40]. The posterior conditional probability is given as follows:

$$p\,(c = 1|\mathbf{X}_i) = \frac{1}{1 + \exp(A \cdot y(\mathbf{X}_i) + B)} \tag{8}$$

where vector $\mathbf{X}_i$ belongs to the class $c = 1$. Obviously, $p(c = -1|\mathbf{X}_i) = 1 - p(c = 1|\mathbf{X}_i)$. The parameters A and B are computed by the optimization of cross-entropy error as follows:

$$\arg\min_{A,B} -\sum_{i=1}^{M} t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \tag{9}$$

where $t_i = (1 + sign[y(\mathbf{X}_i)])/2$ and $p_i = p(c = 1|\mathbf{X}_i)$. The numerical problems of optimization are introduced in [41].

So far, many fast implementations of SVM have been proposed to compute the globally optimal solution. One of the most popular is sequential minimal optimization (SMO), proposed by Platt [42] in 1998. The SMO algorithm decomposes the optimization problem into many smaller-scale problems, which requires only two Lagrange multipliers at one time. This strategy makes it possible to obtain the objective function value of quadratic programming by means of the analytical method, which significantly accelerates the training speed of SVM. The specific implementation of SMO algorithm and numerical problems to be noted, please refer to [42].

2.4 Template Attacks

TA is the most powerful power analysis attack in an information theory sense. The classical TA is based on the multivariate Gaussian distribution $N(\mathbf{t}; (\mathbf{m}, \mathbf{C}))$ as follows:

$$(2\pi)^{-\frac{N}{2}} |\mathbf{C}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\mathbf{t} - \mathbf{m})^T \mathbf{C}^{-1} (\mathbf{t} - \mathbf{m}) \right) \tag{10}$$

where $\mathbf{t}$ represents a N-dimensional vector, $\mathbf{m}$ is the mean vector, $\mathbf{C}$ is the covariance matrix.

In parametric estimation theory, suppose that the number of traces is P_k, the given operation is expressed as O_k, and a power trace is recorded as $\mathbf{t}_{p_k}|O_k$. The estimated template consists of a set of mean vectors $\{\mathbf{m}_k\}$ and a set of covariance matrices $\{\mathbf{C}_k\}$, $k = 1, \ldots, K$. In the maximum likelihood approach, the parameters that maximize the likelihood are selected. Maximizing the likelihood is equal to the log likelihood maximization, which is given by:

$$\log L_k = \log \prod_{p_k=1}^{P_k} p(\mathbf{t}_{p_k}|O_k) = \sum_{p_k=1}^{P_k} \log N\left(\mathbf{t}_{p_k}|\mathbf{m}_k, \mathbf{C}_k\right) \tag{11}$$

where $p(\mathbf{t}_k|O_k)$ is the likelihood probability of power traces $\mathbf{t}_{p_k}$ under the operation O_k performed on the cryptographic device.

In terms of SCAs, where an erroneous environment is assumed, an attacker is more interested in the probability of an instance $\mathbf{X}_i$ belonging to the class c. Hence, instead of predicting the class c, we predict the posterior conditional probability $P_{\text{SVM}}(\mathbf{X}_i|c)$ of each class c. Since the probability estimate of multi-class SVM is a very specialized discipline, we refer to [41] for the knowledge of how to calculate $P_{\text{SVM}}(\mathbf{X}_i|c)$. The log likelihood of each possible key k is as follows:

$$\log L_k \equiv \log \prod_{i=1}^{M_k} P_{\text{SVM}}(\mathbf{X}_i|c) \equiv \sum_{i=1}^{M_k} \log P_{\text{SVM}}(\mathbf{X}_i|c) \tag{12}$$

where M_k is the number of power traces belonging to the key k. The key k^* that maximizes the log likelihood in (11) or (12) is chosen, which is given as follows:

$$\arg\max_{k^*} \log L_{k^*}. \tag{13}$$

3. Wavelet Kernel for Power Analysis

As we all know, the kernel function has been applied in many pattern recognition and machine learning algorithms. By introducing the kernel function, SVM avoids the problem of processing data in high dimensional space and even theoretically infinite dimensional space. The kernel function also maintains the reasonable computational complexity of SVM in the feature space.

3.1 Common Kernel Functions

In general, the kernel function consists of a linear kernel, a polynomial kernel, and an RBF kernel, which must satisfy the Mercer theorem [43]. The Linear kernel function (Linear kernel):

$$K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{X}_i^T \mathbf{X}_j. \tag{14}$$

The Polynomial kernel function (Poly kernel):

$$K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i^T \mathbf{X}_j + 1)^d. \tag{15}$$

The Gaussian kernel function (RBF kernel):

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp\left(-\gamma \left\|\mathbf{X}_i - \mathbf{X}_j\right\|^2\right), \gamma > 0 \tag{16}$$

where $d \geq 1$ is the order of polynomials, γ is the hyperparameter of RBF kernel, and the notation $\|\cdot,\cdot\|$ represents the Euclidean distance between two vectors.

The kernel function is equivalent to a similarity function in some feature space. Given two objects, the kernel outputs a similarity score. As long as the kernel functions know how to compare them, the objects can be anything. For the Linear kernel in (14), the similarity is the product of the length of $\mathbf{X}_i$ and the projection length of $\mathbf{X}_j$ in the direction of $\mathbf{X}_i$. The similarity of RBF kernel between two vectors in (16) is reweighted by the hyperparameter γ. A small γ will result in low bias and high variance while a large γ will get higher bias and low variance [35]. Accordingly, when choosing the appropriate kernel function and its hyperparameters to solve practical problems, the expertise in the relevant areas of problems is necessary.

3.2 Wavelet Kernel Functions

From a perspective of signal analysis, one power trace is also a continuous signal in time domain. The traditional signal analysis theory is based on Fourier analysis, which has many deficiencies in the non-stationary signal. Compared with Fourier analysis, wavelet analysis processes signal simultaneously in time domain and frequency domain, which extracts information more effectively from processed signals. Wavelet analysis adopts fast attenuation, known as the wavelet to represent signal waveforms, which can arbitrarily scale and shift the input signal. The wavelet function is:

$$h_{a,b}(x) = \frac{1}{\sqrt{a}} h\left(\frac{x-b}{a}\right) \tag{17}$$

where a is a dilation factor and b is a translation factor. A detailed introduction of wavelet analysis is given in [44], [45]. Zhang et al. [33] proved the method of constructing wavelet kernel. Let $h(x)$ be a wavelet function, respectively $a, b_i, b'_i, x_i, x'_i \in R$, $\mathbf{X}, \mathbf{X}' \in R^N$, and then the wavelet kernel is:

$$K(\mathbf{X}, \mathbf{X}') = \prod_{i=1}^{N} \left(h\left(\frac{x_i - b_i}{a}\right) \cdot h\left(\frac{x'_i - b'_i}{a}\right)\right) \tag{18}$$

where $\mathbf{X} = (x_1, x_2, ..., x_N)$ and $\mathbf{X}' = \left(x'_1, x'_2, x_3 ..., x'_N\right)$ are N dimensional vectors. and the translation-invariant wavelet kernel is:

$$K(\mathbf{X}, \mathbf{X}') = \prod_{i=1}^{N} h\left(\frac{x_i - x'_i}{a}\right). \tag{19}$$

The Morlet wavelet function is given as follows:

$$h(x) = \cos(1.75x) \cdot \exp(-\frac{x^2}{2}). \tag{20}$$

Thus, the wavelet kernel based on Morlet wavelet function is [33]:

$$K(\mathbf{X}, \mathbf{X}') = \prod_{i=1}^{N} \left(\cos\left(\frac{1.75(x_i - x'_i)}{a}\right) \exp\left(-\frac{\|x_i - x'_i\|^2}{2a^2}\right)\right). \tag{21}$$

Later, many wavelet kernel functions including Gaussian wavelet kernel function [46], [47] were proposed. The Gaussian wavelet function is defined as follows:

$$h(x) = (-1)^{\frac{p}{2}} C_p(x) \exp\left(-\frac{1}{2}x^2\right) \tag{22}$$

where p is a positive even integer, $C_p(x)\exp(-\frac{1}{2}x^2)$ is the pth step's differential coefficient of Gaussian function. The form of Gaussian wavelet function varies with the p value. When p is zero, $C_p(x)$ is 1, which is actually a type of Gaussian function. When the value of p is 2, $C_p(x) = x^2 - 1$ is called the Mexican hat wavelet function. When the value of p is 4, $C_p(x) = x^4 - 6x^2 + 3$ is a four order polynomial that is unstable in the numerical theory. Therefore, we chose Gaussian function and Mexican hat wavelet function as kernel functions. The RBF kernel based on Gaussian function is shown in (16). The wavelet kernel based on Mexican hat wavelet function is:

$$K(\mathbf{X}, \mathbf{X}') = \prod_{i=1}^{N} \left(\left(1 - \frac{(x_i - x'_i)^2}{a^2}\right) \exp\left(-\frac{\|x_i - x'_i\|^2}{2a^2}\right)\right). \tag{23}$$

For the purpose of theoretical completeness and paying tribute to J. Fourier, here the Fourier kernel [48], [49] is given as follows:

$$K(\mathbf{X}, \mathbf{X}') = \prod_{i=1}^{N} \frac{1 - q^2}{2(1 - 2q\cos(x_i - x'_i) + q^2)}. \tag{24}$$

The wavelet kernel approximates the non-stationary signal with high precision, which is impossible for the traditional kernels. The traditional kernel functions such as Gaussian function are related and even redundant. However, the wavelet function is orthonormal, which almost approximates any function in continuous space, thus the generalization of wavelet SVM is improved. Meanwhile, the sparse wavelet kernel accelerates the training speed of SVM. Although the wavelet kernel requires more time to process power traces than other kernels, the overall training time of wavelet SVM is significantly decreased. Consequently, we creatively proposed an assumption that wavelet SVM has better stability and fewer iterations than SVM based on others kernels in power analysis.

4. Experiments

Prof. Lin Chih-Jen of Taiwan University has developed a widely used SVM kit, known as LIBSVM (Library

for Support Vector Machines) [50]. LIBSVM is an integrated software for distribution estimation (one-class SVM), C-support vector classification (C-SVC), nu-support vector classification (nu-SVC), epsilon-support vector regression (epsilon-SVR), and nu-support vector regression (nu-SVR). It supports different SVM formulations (multi-class SVM, weighted SVM for unbalanced data), cross-validation, model selection, various kernels, probability estimates, etc. The one-aginst-one strategy is used to predict the probability output P_{SVM}. The highly optimized C-SVC makes it easy to set parameters for the given classification problem.

4.1 How to Set Parameters

It is theoretically possible to test an infinite number of parameters, but in practice, it makes no sense. According to article [51], we selected the regularization parameter C from 0.01 to 10, epsilon (tolerance of termination criterion) from 0.01 to 1 and various kernels including Linear kernel, Poly kernel, RBF kernel, Fourier kernel and wavelet kernels, together it was 3600 of combinations. Besides, TA was selected as a comparison. The Hamming weight model assumes the intermediate power consumption of the entire byte instead of only multiple bits, thus it is selected as the hypothetical power leakage model in this paper.

The first dataset (DS1) aims to break the 16th byte of the first round key of the unmasked AES algorithm. TeSCASE Group has [53] implemented this algorithm on the Sasebo-GII board [52] provided by RCIS [54]. This board has a mechanism to provide users various means to access the reconfiguration function of FPGA. Only 900 power traces were selected to locate interesting points in this dataset. We computed the Pearson correlation between each instant of power traces and the Hamming weight of the S-Box output and then selected the 32 highest correlated points as interesting points. An example of power traces is shown in Fig. 1. The first peak is the plaintext loaded into the register, and the next 10 peaks correspond to 10 rounds of the unmasked AES algorithm.

The second dataset (DS2), which includes 1000 power traces, is prepared for the masked AES implemented in software. Power traces are freely available on the DPA Contest v4 (DPACv4) website [56]. The masking scheme, known as RSM [55], is an additive Boolean masking countermeasure with 16 masked S-Boxes. The mask values are rotated according to the offset value. All power traces were measured during the first round and the beginning of the second round of AES algorithm. The label value corresponds to the offset value (0 to 15). The Pearson correlation between the offset value and each instant of power traces was used to locate interesting points. We selected the two highest correlated points for each mask value, thus the number of interesting points was 32. The offset value is highly correlated with sample points except for the central part of each trace in Fig. 2.

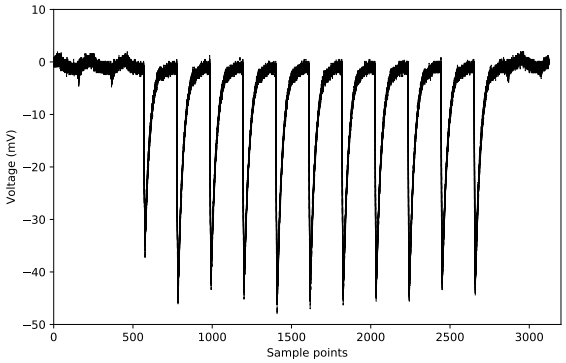

Fig. 1. Power traces of the unmasked AES in DS1.

The third dataset (DS3) concentrates on the second byte of the first round key of the masked AES algorithm. We selected the Hamming weight of the second S-Box (SBox1) output as the label of an SVM classifier. That is, the label value corresponds to the Hamming weight value of the output byte (0 to 8). This dataset includes 3600 power traces, but only 1800 traces are randomly selected in each test. The reason is that the number of traces corresponds to each label may be not the same, namely SVM with imbalanced data. We calculated the Pearson correlation between each instant of 3600 power traces and the Hamming weight of the SBox1 output to locate interesting points. Besides, the 32 highest correlated points were selected as interesting points. Only a particular interesting points have larger power leakage of the S-Box output as illustrated in Fig. 3.

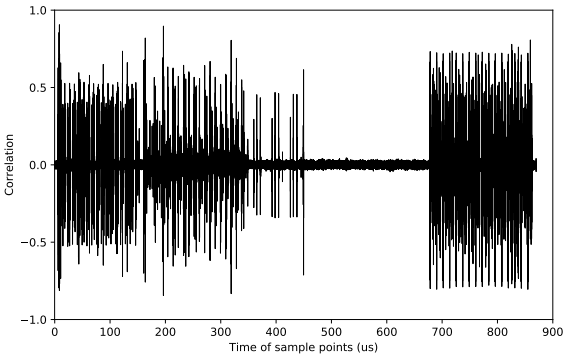

Fig. 2. Correlation between the Hamming weight of the offset value and the power consumption in the 1st round of the masked AES in DS2.

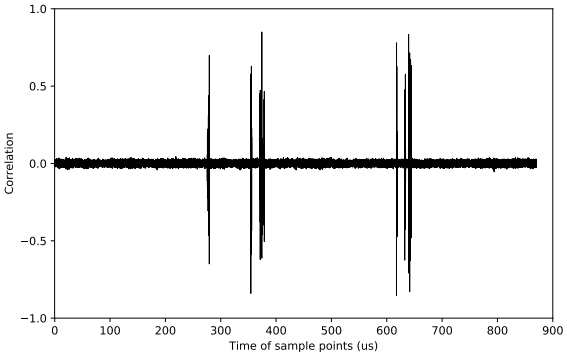

Fig. 3. Correlation between the Hamming weight of the Sbox1 output and the power consumption in the 1st round of the masked AES in DS3.

Here only the Pearson correlation was used for feature selection. In addition, we recommend using other methods, such as the minimum redundancy maximum correlation [57], principal component analysis [58], etc.

4.2 How to Compare Performance

In order to compare TA and several SVM algorithms, we performed different experiments on three datasets. We assumed that the attacker had complete control over the cryptographic device, who measured enough power traces to recover the secret key.

Our experimental methodology was as follows: Given a dataset, a random two-third was reserved as the learning set and the remaining one-thirds was used as the test set. The learning set generated training and validation sets by using 5-fold cross validation. The validation sets of all folds were used to optimize hyperparameters of SVM. The optimal hyperparameter (the one that has the highest average accuracy on the validation folds) was used to train the final SVM algorithm on the training set.

The most popular evaluation method is the accuracy (success rate) of the independent test set. Consequently, we selected the success rate as a measure in the offset recovery phase. In order to make the results more reliable, each experiment was repeated 10 times, and then the success rate of the corresponding test set was recorded. The final result was the average of all success rates.

The success rate is adequate when the number of power traces corresponding to each predicted class is the same. However, for the key recovery phase, which assumes multiple traces, the success rate is not suited as a measure. The problem is that the most likely Hamming weight class has the largest number of power traces when the success rate is selected as a metric. The guessing entropy [59] was selected to evaluate the number of remaining keys. The guessing entropy is defined as follows: let g include the descending probability ranking of all possible keys and i represent the position of the correct key in g. After performing s experiments, one gets a matrix $[g_1, g_2, ..., g_s]$ and a corresponding vector $[i_1, i_2, ..., i_s]$. In other words, the guessing entropy represents the average number of power traces required to recover the correct key. Thus, GE^1 (a guessing entropy of 1) was selected as a measure in the key recovery phase.

5. Results and Analysis

We recovered the offset value of the masked AES by using DS2, especially DS1 and DS3 were used for the key recovery phase. Each attack extracted the offset value before performing the key recovery of DS3. All our experiments were performed on Asus laptop with 2.50GHz Intel Core (TM) i5, 8GB 1067MHz DDR3 (Windows7 x64 Ultimate). The attack lasted about 5 weeks without considering the time to create three datasets.

5.1 Finding the Offset Value

This section explores the performance of different approaches that expose the offset value by using power traces of DS2. We randomly selected 500 traces as a training set and 250 traces as a test set. Moreover, 3~32 interesting points were used in our experiments, which were the most correlated with the offset value. We compared TA and various SVM algorithms such as SVM-Linear, SVM-Poly, SVM-RBF, SVM-Fourier, and wavelet SVM from the success rate and the time required. The impact of the training set size on the success rate was also discussed.

The first experiment explored the effect of various kernels on the success rate of SVM when the training set size was different. Figures 4 and 5 describe the success rate of different numbers of interesting points when using SVM-Linear and SVM-Poly to recover the offset value. First, the performance of SVM-Linear or SVM-Poly is basically not affected by the training set size, which is mainly due to the fact that the feature space of traces is not linearly separable. Furthermore, SVM-Linear and SVM-Poly require a lot of iterations to find the appropriate hyperplane, resulting in very low training efficiency. Second, the number of interesting points per trace significantly determines the success rate. In general, the more interesting points, the higher the success rate. Interestingly, the success rate of 100% size of the training set is obviously lowest for SVM-Poly when the number of interesting points is 16 to 28. One possible explanation is that SVM-Poly appeared overfitting when the training set size is small. We did not give the results of SVM-Linear and SVM-Poly due to the poor performance.

Figures 6, 7, 8, and 9 reveal the corresponding success rates for different numbers of interesting points when SVM-RBF, SVM-Fourier, SVM-Morlet, and SVM-Mexican are used to predict the offset value. As expected, the success rate of SVM increases as the number of interesting points increases. Moreover, the larger training set size, the higher the success rate. This can be explained that the performance of SVM is determined by its parameters, and the training set size is critical to the best parameters of SVM. When the training set size is expanded from 25% to 50%, the success rate of SVM increases significantly, but when the training set size is expanded from 75% to 100%, the success rate of SVM is not obviously improved. Wavelet SVM and SVM-Fourier obtain much higher success rates than SVM-RBF due to the powerful approximation capability.

The purpose of the second experiment was to study the success rate of TA and compare the efficiency of TA and SVM based on various kernels. Figure 10 illustrates the relationship between the success rate of TA and the size of the training set when the number of interesting points is 3 to 32. Generally, the larger number of interesting points, the higher the success rate of TA. However, for the small training set (25%~50% size), the success rate of TA is reduced when the number of interesting points exceeds a certain value. The reason is that when the number of interesting points is

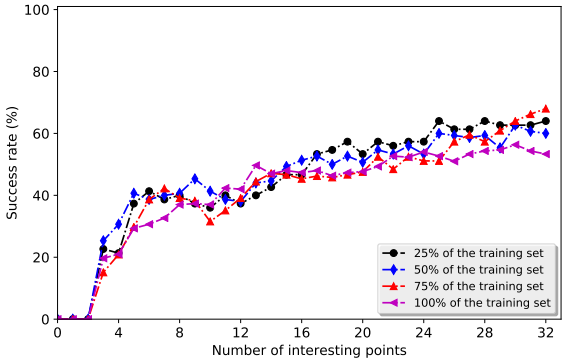

Fig. 4. Success rate of finding the offset value based on SVM-Linear by using power traces of DS2.

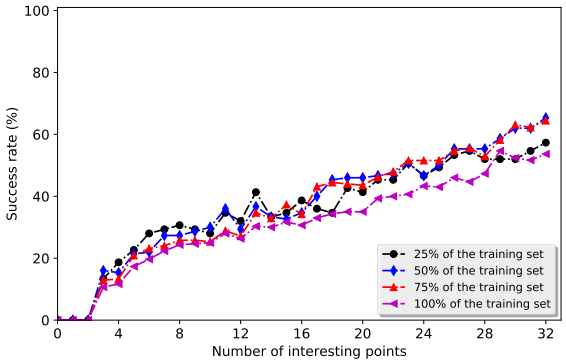

Fig. 5. Success rate of finding the offset value based on SVM-Poly (with degree 2) by using power traces of DS2.

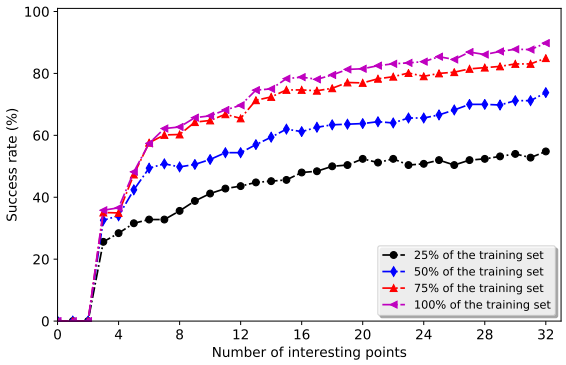

Fig. 6. Success rate of finding the offset value based on SVM-RBF by using power traces of DS2.

too large, the covariance matrix may be an ill-conditioning matrix [11]. The results illustrate that SVM extracts more information of the offset value than TA. The performance of TA is equivalent to wavelet SVM when the training set size exceeds 75%, but the computational complexity of TA is higher than SVM based on various kernels (see Fig. 11). More precisely, the success rate of TA is very good while its prediction time increases exponentially with the number of interesting points. Although the classical TA does not work well in terms of efficiency, it is still selected for comparison in later experiments.

The third experiment was used to find the optimal dilation factor a in (21) and (23) by comparing the success rate of wavelet SVM. If a is very large, then even if two vectors are quite similar, the kernel function will output a small value.

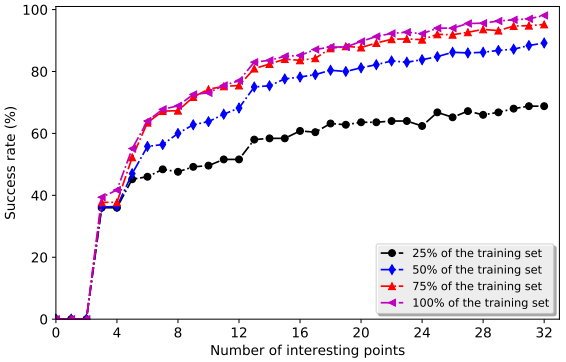

Fig. 7. Success rate of finding the offset value based on SVM-Fourier by using power traces of DS2.

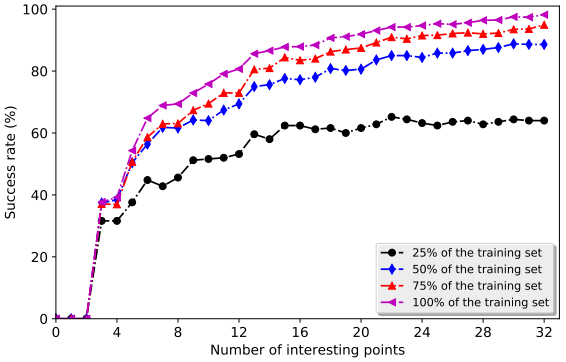

Fig. 8. Success rate of finding the offset value based on SVM-Morlet by using power traces of DS2.

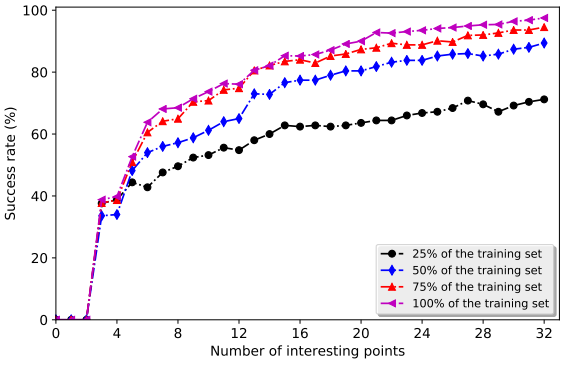

Fig. 9. Success rate of finding the offset value based on SVM-Mexican by using power traces of DS2.

In other words, the support vectors obtained by a training set have little impact on the classification of the test set, which causes the model may be prone to overfitting. When the value of a is small, support vectors have a great effect on the classification. This means that you may not be able to obtain a complex decision boundary. The optimal value of a is 3.2 in our experiments. We used the wavelet function and the value of a to construct a new symbol that represents the type of wavelet kernel. For example, the symbol Mexican1 represents the wavelet function is Mexican hat wavelet function and the value of a is 1. From the perspective of success rate, wavelet SVM is 5~8% higher than SVM-RBF in Fig.12. Note that the success rates of SVM-Morlet1 and SVM-Mexican1 are very low when the number of interesting points exceeds a certain value. This can be interpreted as

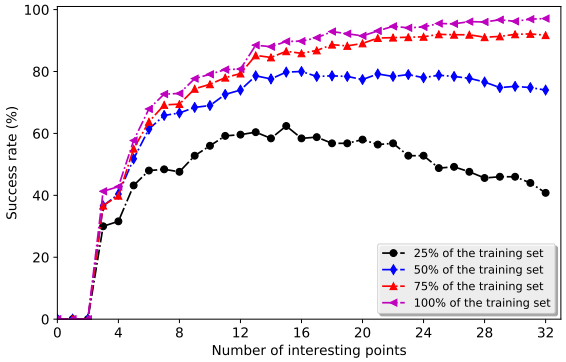

Fig. 10. Success rate of finding the offset value based on TA by using power traces of DS2.

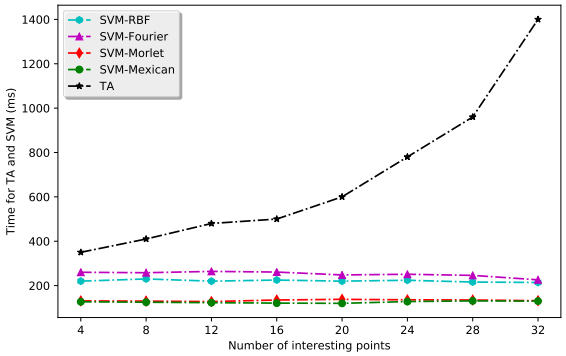

Fig. 11. The training time required of SVM vs TA by using 500 power traces of DS2.

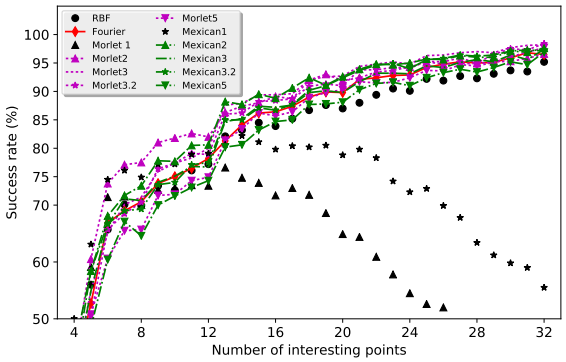

Fig. 12. Success rate of finding the offset value based on different kernels by using 500 power traces of DS2.

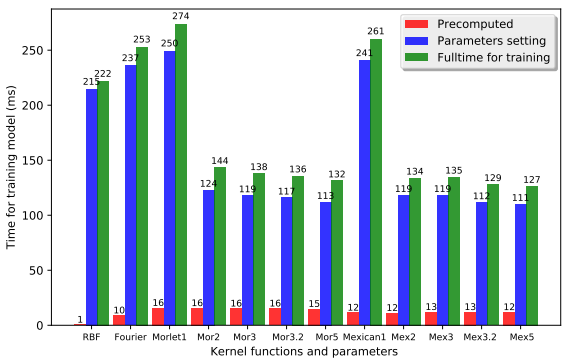

Fig. 13. The efficiency of finding the offset value based on different kernels by using 500 power traces of DS2 [Mor = Morlet, Mex = Mexican].

	C=0.3	C=0.7	C=1	C=7	C=10
RBF	70.3	85.3	90.4	95.6	91.2
Fourier	78	96.8	97.1	98.6	97.1
Morlet1	18.4	35.6	36.7	40.1	40.3
Morlet2	82.6	93.7	95.9	96.7	95.6
Morlet3	86.2	95.8	96.7	97.5	96.5
Morlet3.2	85.8	96.4	97.5	98.4	97.6
Morlet5	78.9	93.5	96.6	97.1	97.6
Mexican1	17.8	58.4	60.3	62.3	60.2
Mexican2	82.7	96.1	97.9	98	97.8
Mexican3	85.6	96.4	97.1	97.8	96.2
Mexican3.2	82.6	97.6	98.8	99.2	97.0
Mexican5	80.1	94.6	95.4	97.3	96.8
TA	83.5	95.3	96.2	97.5	95.1

Tab. 1. The effect of parameters C on success rate of SVM based on different kernels by using power traces of DS2 when the number of interesting points is 32, epsilon=0.32.

an inappropriate value of a. When the value of a is too small, the simple decision boundary deteriorates the generalization of wavelet SVM.

In the fourth experiment, a fixed number of power traces were selected to compare the efficiency of SVM based on various kernels. The efficiency of SVM was evaluated by measuring the time required to use the kernel function to process interesting points (Precomputed), the time required to train parameters of SVM (Parameters setting), and the time required to perform a complete training (Fulltime for training). The results indicate that when the value of a is appropriate, the overall time of wavelet SVM is less than SVM based on other kernels (see Fig. 13). However, wavelet SVM requires more time to process interesting points. Hence, the value of a is crucial for the efficiency of wavelet SVM. When the value of a is near 3.2, the overall training time of wavelet SVM is greatly reduced. Although the Precomputed time is negligible, the overall time of SVM-RBF is still more than wavelet SVM. In the view of time cost, SVM-Fourier and SVM-RBF are similar, but the Precomputed time of SVM-Fourier is equal to wavelet SVM. When the value of a is appropriate, the overall time of wavelet SVM is almost reduced by 30% to 40% compared with SVM-RBF. Consequently, the wavelet kernel accelerates the convergence speed of setting parameters and ultimately reduces the overall training time.

The last experiment was conducted to verify the effect of penalty factor C on the success rate of SVM. The penalty factor controls the cost of misclassification on the training set, which indicates the importance of misclassification to SVM. The large value of C implies the high cost of misclassification (hard margin), which allows SVM to increase the number of iterations to optimize the separating hyperplane. In other words, the generalization of SVM drops due to the large value of C. When the value of C is small, the cost of misclassification is low (soft margin), allowing some misclassifications. The small value of C makes SVM tend to accelerate the speed of training, resulting in a decrease in success rate. The best C is to find a balance between hard margin and soft margin. When the value of C is 0.3, 0.7, 1,

7, and 10, the success rates of SVM based on various kernels are given in Tab. 1. The final results show that the optimal value of C is 7 when the value of epsilon is 0.32.

5.2 Key Recovery Phase

The experiments were carried out to recover the secret key by using power traces of DS1 and DS3. Here, the number of power trace was not limited to only one, thus we made use of various methods to recover the key. In this paper, the Hamming weight leakage model is based on the entire intermediate value of the S-Box output rather than a single bit. Therefore, we adopted the probabilistic multi-class SVM algorithm to distinguish nine different classes. We combined the prediction results of a binary-class SVM classifier from N power traces X_i belonging to the class c by using the posterior probability output $P_{\mathrm{SVM}}(X_i|c)$. We performed the maximum likelihood estimate for each possible key and then selected the key that maximizes the likelihood in (13) by using multiple power traces. Ultimately, a guessing entropy of 1 (GE^1) was selected to measure and compare the performance of different algorithms in the key recovery phase.

In the first experiment, the 16th byte of the last round key of the unmasked AES was extracted by using power traces of DS1. We randomly selected 400 power traces as a training set, 200 power traces as a test set. As can be seen from Figure 14, the success rate of the Hamming weight of the S-Box output increases as the number of interesting points increases. When the value of a is appropriate, the performance of wavelet SVM is still better than SVM-RBF in the key recovery phase. However, with the increase in the number of interesting points, the success rate of SVM-RBF decreases significantly and becomes very unstable. The reason may be that sample points of power traces of DS1 don't obey the multivariate Gaussian distribution. In order to improve the performance of SVM, we adopted cross validation and grid search algorithms to optimize hyperparameters. Overall, the success rate of SVM based on various kernels is maintained at 60~90%. Especially, when the value of a is 2, the success rates of SVM-Morlet and SVM-Mexican reach about 90%. Note that SVM-Fourier obtains a fairly good success rate when the number of interesting points exceeds a certain value, but the time required is more than wavelet SVM. Therefore, wavelet SVM has advantages over TA and SVM based on other kernels when using power traces of DS1 in the key recovery phase.

The second experiment was aimed at the masked AES implementation in terms of the assumption that the offset value of each trace is known. The distribution of the Hamming weight of the S-Box output is not uniformly distributed, which has little effect on the experimental results. After all, the prediction results of SVM are independent of the distribution of the Hamming weight class when the guessing entropy is selected as a measure. As long as the dataset ensures that the number of power traces per Hamming weight class is sufficient, SVM for the unbalanced data is also competent. Here, we performed the maximum likelihood estimate for each possible key by using probabilistic multi-class SVM. A fixed value of the guessing entropy was selected to assess how many power traces of DS3 were required. Figures 15, 16, 17 and 18 describe the maximum likelihood probability of all possible guessing keys (0x00~0xff, and the correct key is 0xec) when using 1 to 20 power traces. SVM-RBF requires the maximum number of power traces to obtain the secret key. Also, SVM-Mexican requires 6 or so power traces to guess the correct key. SVM-Morlet works slightly

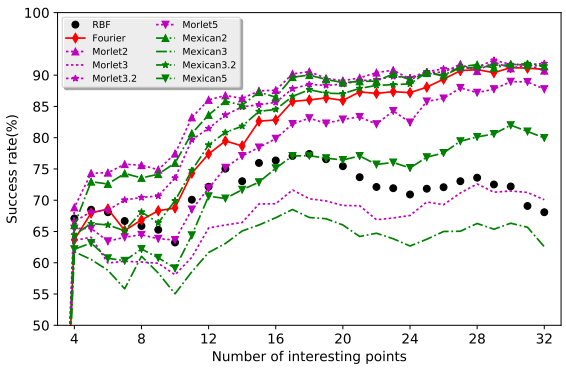

Fig. 14. Success rate of the Hamming weight of the S-Box output based on various kernels by using power traces of DS1.

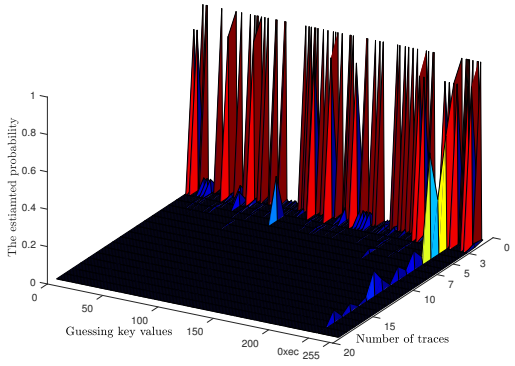

Fig. 15. The maximum likelihood estimation of guessing entropy based on SVM-RBF by using traces of DS3.

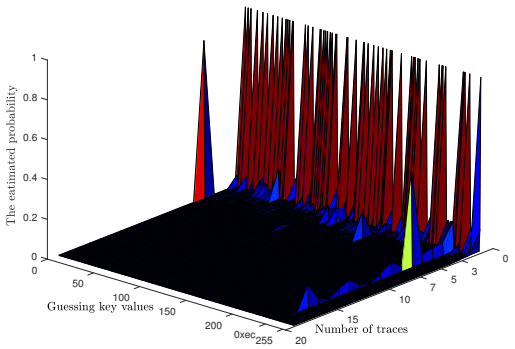

Fig. 16. The maximum likelihood estimation of guessing entropy based on SVM-Fourier by using traces of DS3.

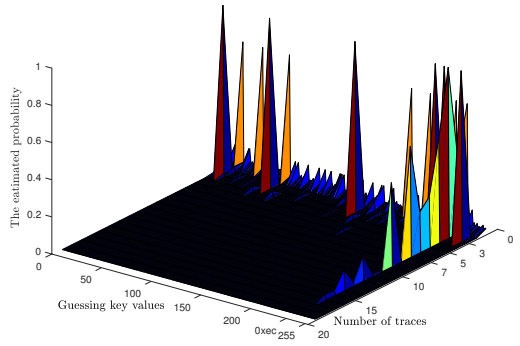

Fig. 17. The maximum likelihood estimation of guessing entropy based on SVM-Morlet by using traces of DS3.

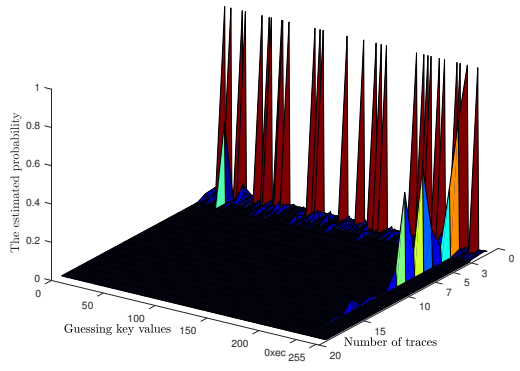

Fig. 18. The maximum likelihood estimation of guessing entropy based on SVM-Mexican by using traces of DS3.

	DS1	DS3
SVM-RBF	9.8	7.2
SVM-Fourier	9.7	6.7
SVM-Morlet	6.7	5.4
SVM-Mexican	6.3	5.3
TA	76	58

Tab. 2. The number of power traces required by SVM and TA when the guessing entropy is set to 1 (GE^1).

better than SVM-Mexican, which needs about 5 power traces. The performance of SVM-Fourier has some instability, which requires over 10 power traces. The results of SVM based on various kernels further confirmed the superiority of wavelet SVM in the key recovery phase.

In the third experiment, TA and SVM were used to recover the secret key by using power traces of DS1 or DS3. For fairness considerations, we repeated hundreds of similar experiments by using TA and SVM based on various kernels and then calculated the average as the final results. The results were recorded in Tab. 2. TA requires about 58 traces to break the key of the unmasked AES implementation by using power traces of DS3. In contrast, wavelet SVM needs a smaller number of power traces for the key recovery of DS3 (5.4 traces in average for SVM-Morlet, 5.3 traces in average for SVM-Mexican). SVM-RBF requires 9.8 traces (using DS1) and 7.2 traces (using DS3) when the guessing entropy is set to 1 (GE^1). The performance of SVM-Fourier is much better than TA, slightly inferior to wavelet SVM. Similar results can be obtained by using power traces of DS1. The results confirm that wavelet SVM is very suitable for the key recovery of the unmasked or masked AES algorithm.

5.3 Comparison with Other Work

In Tab. 3, we summarized the previous work of using non-SVM learning algorithms we have discussed in Sec. 1. We described the results of learning algorithms such as MLP, k-NN, and RF in detail.

The MLP algorithm recovered the key based on one power trace [14]. It achieved 85% empirical success rate and 80% theoretical success rate. They focused on the first byte of the secret key. The authors [15] later proposed averaging of power traces as the preprocessing method, which improved the success rate to 96%. However, they did not give a specific feature selection method and training time. The training process of MLP is very time-consuming in practice.

The k-NN algorithm exhibited great potential in power analysis attacks [19]. The standard CPA and Pearson correlation were used to locate interesting points. They chose 50 sample points for each trace on three datasets as interesting points. The success rate of k-NN ($k = 5$) was 94.97%. The time required to perform one 10-fold cross validation was less than 1 s. The training time of k-NN is much less than other learning algorithms, but the k nearest neighbor search for all training instances is very time-consuming in the testing phase. Moreover, the k-NN algorithm is very sensitive to neighbor instances. If the neighbor instance happens to be noise, the prediction results will go wrong [35].

The goal was to attack a single bit of 3DES by using the RF algorithm, and its performance went beyond TA [23]. Besides, they used many feature selection methods such as Ranking, PCA, Minimum redundancy maximum relevance (mRMR), and SOM. They selected 20 sample points as interesting points in all experiments. The RF algorithm increased the probability of recovering one byte of the key from 5.80% (TA) to 15.33%. Lerman et. al [27] presented the machine learning attack against the masked AES algorithm by using 1500 power traces of DPACv4. The Pearson correlation coefficient between the offset value and sample points of each trace was used to locate interesting points. They chose 50 points that are most correlated with the offset value as intereting points. Due to the strategy of feature selection, interesting points may be too concentrated to extract more power consumption leakage information. For the RF algorithm, the success rate of recovering the offset value was about 80%. The time to process one trace in the learning phase was less than 1 ms.

In our work, wavelet SVM successfully recovered the offset value of the masked AES algorithm for each power trace, which was obviously 5~8% higher than SVM-RBF and the time required was almost reduced by 40% when

Ref.	Machine learning	Algorithm	No. of traces	Performance
[13]	k-means	ECC	9 traces	recovers the secret scalar from single execution attack
[14]	MLP	AES	2560 traces	the theoretical and empirical success rates are 80% and 85%
[15]	Improved MLP	AES	2560 traces	recovers the key from one trace with accuracy $\geq$ 96%
[17]	GA	DES	2000 (time), 7000 (freq.)	reduces no. traces by 60% by attacking multiple S-Boxes
[19]	k-NN	AES	2560 traces	recovers the secret offset with accuracy 94.97%
[23]	RF	3DES	400 (a byte of the key)	performs binary bit classification better than TA
[27]	RF	AES	1500 traces	finds the offset value with accuracy 80%

Tab. 3. Machine learning in power analysis attacks.

using the optimal hyperparameters. In order to break the secret key, wavelet SVM only required in average 5.4 traces for the unmasked AES algorithm and less than 7 traces for the masked AES algorithm. Considering the training time and success rate, we firmly believe that the performance of other learning algorithms is slightly inferior to SVM.

6. Conclusion and Future Work

As can be seen from the above description, power analysis attacks are viewed as the classification problems. Power analysis attacks and machine learning create templates (features) to describe power traces of a training set and then calculate the similarity between templates (features) and power traces of a test set. Finally, the results are given with a certain probability. Generally, power analysis attacks assume that sample points of power traces are approximated by a set of finite normal distributions. However, machine learning assumes that sample points are independent and identically distributed, but not restrict to a certain distribution.

TA assumes that sample points of each power trace follow a multivariate Gaussian distribution. Moreover, TA not only describes the power consumption information but also inevitably describes the noise, which makes it need a lot of traces to improve the signal to noise ratio. However, the strategy adopted by SVM is quite different from TA. The key of SVM is to quickly find the separating hyperplane of the offset values or the Hamming weights. TA aims to simulate the real power consumption distribution by considering all sample points of traces. However, SVM focuses on the separation of classes, using only support vectors. Consequently, TA requires more power traces than SVM in the learning (profiling) phase.

Wavelet analysis can approximate any function, which is a powerful tool to process nonlinear and multidimensional signals. SVM is very suitable for solving the classification problems of small-scale dataset. SVM-RBF based on Gaussian function is the most commonly used in engineering, but not necessarily the optimal solution to solve all classification problems. In light of our experiments, wavelet SVM significantly improves the success rate of finding the offset value, which requires less power traces than SVM-RBF in the key recovery phase. Accordingly, wavelet SVM show excellent performance and efficiency in power analysis attacks.

The application of machine learning in power analysis attacks has not been fully explored. An important direction of research is to use the learning algorithms for feature selection. Furthermore, the development of a customized learning algorithm for power analysis attacks will be a challenging area of future work.

References

[1] KOCHER, P. C., JAFFE, J., JUN, B. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*. London (UK), 1999, p. 388–397. DOI: 10.1007/3-540-48405-1_25

[2] KOCHER, P. C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. Santa Barbara (USA), 1996, p. 104–113. DOI: 10.1007/3-540-68697-5_9

[3] GENKIN, D., SHAMIR, A., TROMER, E. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin (Germany), 2014, p. 444–461. DOI: 10.1007/978-3-662-44371-2_25

[4] GANDOLFI, K., MOURTEL, C., OLIVIER, F. Electromagnetic analysis: Concrete results. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2001*. Paris (France), 2001, p. 251–261. DOI: https://doi.org/10.1007/3-540-44709-1_21

[5] STANDAERT, F., ARCHAMBEAU, C. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2008*. Washington, D.C (USA), 2008, p. 411–425. DOI: 10.1007/978-3-540-85053-3_26

[6] CHARI, S., RAO, J., ROHATGI, P. Template attacks. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2002*. Redwood Shores (USA), 2002, p. 13–28. DOI: 10.1007/3-540-36400-5_3

[7] CHOUDARY, O., KUHN, M. G. Efficient template attacks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin (Germany), 2014, p. 253–270. DOI: 10.1007/978-3-319-08302-5_17

[8] BRIER, E., CLAVIER, C., OLIVIER, F. Correlation power analysis with a leakage model. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2004*. Cambridge (USA), 2004, p. 16–29. DOI: 10.1007/978-3-540-28632-5_2

[9] GIRELICHS, B., BATINA, L., TUYLS, P., et al. Mutual information analysis. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2008*. Washington, D.C. (USA), 2008, p. 426–442. DOI: 10.1007/978-3-540-85053-3_27

[10] SCHINDLER, W., LEMKE, K., PAAR, C. A stochastic model for differential side channel cryptanalysis. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2005*. Edinburgh (UK), 2005, p. 30–46. DOI: 10.1007/11545262_3

[11] MANGARD, S., OSWALD, E., POPP, T. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. 1st ed. Secaucus (USA): Springer US, 2007. ISBN: 978-0-387-30857-9

[12] RIVEST, R. L. Cryptography and machine learning. In *Proceedings of the International Conference on the Theory and Applications of Cryptology: Advances in Cryptology - ASIACRYPT'91*. 1991, vol. 739, p. 427–439. DOI: 10.1007/3-540-57332-1_36

[13] HEYSZL, J., IBING, A., MANGARD, S., et al. Clustering algorithms for non-profiled single-execution attacks on exponentiations. In *Proceedings of the 12th International Conference Smart Card Research and Advanced Applications CARDIS 2013*. Berlin (Germany), 2013, p. 79–93. DOI: 10.1007/978-3-319-08302-5_6

[14] MARTINASEK, Z., ZEMAN, V. Innovative method of the power analysis. *Radioengineering*, 2013, vol. 22, no. 2, p. 586–594. ISSN: 1210-2512

[15] MARTINASEK, Z., HAJNY, J., MALINA, L. Optimization of power analysis using neural network. In *Proceedings of the 12th International Conference Smart Card Research and Advanced Applications CARDIS*. Berlin (Germany), 2013, p. 94–107. DOI: 10.1007/978-3-319-08302-5_7

[16] WHITNALL, C., OSWALD, E. Robust profiling for DPA-style attacks. In *Proceedings of the International Workshop Cryptographic Hardware and Embedded Systems - CHES 2015*. Saint-Malo (France), 2015, p. 3–21. DOI: 10.1007/978-3-662-48324-4_1.

[17] ZHANG, Z., WU, L., WANG, A., et al. Improved leakage model based on genetic algorithm. In *IACR Cryptology ePrint Archive*. 2014.

[18] AUMONIER, S. Generalized correlation power analysis. In *Proceedings of the Ecrypt Workshop Tools for Cryptanalysis*. 2007, vol. 518.

[19] MARTINASEK, Z., ZEMAN, V., MALINA, L., et al. k-Nearest neighbors algorithm in profiling power analysis attack. *Radioengineering*, 2016, vol. 25, no. 2, p. 365–382. DOI: 10.13164/re.2016.0365

[20] HOSPODAR, G., GIERLICHS, B., DE MULDER, E., et al. Machine learning in side-channel analysis: A first study. *Journal of Cryptographic Engineering*, 2011, vol. 1, no. 4, p. 293–302. DOI: 10.1007/s13389-011-0023

[21] HOSPODAR, G., DE MULDER, E., GIERLICHS, B., et al. Least squares support vector machines for side-channel analysis. In *Proceedings of the Second International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2011)*. Darmstadt (Germany), 2011, p. 293–302.

[22] HE, H., JAFFE, J, ZOU, L. Side channel cryptanalysis using machine learning: Using an SVM to recover DES keys from a smart card. 2012, Stanford University.

[23] LERMAN, L., BONTEMPI, G., MARKOWITCH, O. Power analysis attack: An approach based on machine learning. *International Journal of Applied Cryptography*, 2014, vol. 3, no. 2, p. 97–115. DOI: https://doi.org/10.1504/IJACT.2014.062722

[24] LERMAN, L., BONTEMPI, G., MARKOWITCH, O. Side channel attack: An approach based on machine learning. In *Proceedings of the Second International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2011)*. Darmstadt (Germany), 2011, p. 29–41. DOI: 10.1504/IJACT.2014.062722

[25] HEUSER, A., ZOHNER, M. Intelligent machine homicide - breaking cryptographic devices using support vector machines. In *Proceedings of the Third International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*. Darmstadt (Germany), 2012, p. 249–264. DOI: 10.1007/978-3-642-29912-4_18

[26] BARTKEWITZ, T., LEMKE-RUST, K. Efficient template attacks based on probabilistic multi-class support vector machines. In *Proceedings of the Smart Card Research and Advanced Applications*. Graz (Austria), 2013, p. 263–276. DOI: 10.1007/978-3-642-37288-9_18

[27] LERMAN, L., MEDEIROS, S. F., BONTEMPI, G., et al. A machine learning approach against a masked AES. In *Proceedings of the 12th International Conference of Smart Card Research and Advanced Applications CARDIS*. Berlin (Germany), 2013, p. 61–75. DOI: 10.1007/978-3-319-08302-5_5

[28] SAEEDI, E., KONG, Y. Side channel information analysis based on machine learning. In *Proceedings of the 8th International Conference on Signal Processing and Communication Systems (ICSPCS)*. 2014, p. 1–7. DOI: 10.1109/ICSPCS 2014.7021075

[29] CHUI, C. K. *Wavelets: A Tutorial in Theory and Applications (Wavelet Analysis and its Applications)*. San Diego, CA (USA): Academic Press, 1992. ISBN: 0323139744, 9780323139748

[30] SOUISSI, Y., ELAABID, M. A., DEBANDE, N., et al. Novel applications of wavelet transforms based side-channel analysis. In *Proceedings of the Non-Invasive Attack Testing Workshop*. 2011.

[31] PARK, A., RYOO, J., HAN, D. G. CPA performance comparison based on wavelet transform. In *Proceedings of the IEEE International Carnahan Conference on Security Technology*. 2012, p. 201–206. DOI: 10.1109/CCST.2012.6393559

[32] DEBANDE, N., SOUISSI, Y., ABDELAZIZ, M., et al. Wavelet transform based pre-processing for side channel analysis. In *Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture Workshops*. Vancouver (Canada), 2012, p. 32–38. DOI: 10.1109/ MICROW.2012.15

[33] ZHANG, L., ZHOU, W., JIAO, L. Wavelet support vector machine. *IEEE Transactions on Systems, Man, & Cybernetics, Part B (Cybernetics)*, 2004, p. 34–39. DOI: 10.1109/TSMCB.2003.811113

[34] CORTES, C., VAPNIK, V. Support-vector networks. *Machine Learning*, 1995, p. 273–297. DOI: 10.1007/BF00994018

[35] HANG, L. *Statistical Learning Method*. 1st ed. Beijing (China): Tsinghua University Press, 2012. ISBN: 978-7-302-27595-4

[36] FRANC, V., HLAVAC, V. Multi-class support vector machine. In *Proceedings of the 16th International Conference on Pattern Recognition*. Quebec (Canada), 2002, vol. 2, p. 236–239. DOI: 10.1109/ICPR.2002.1048282

[37] DIETTERICH, T. G., BAKIRI, G. Solving multiclass learning problems via error correcting output codes. *Journal of Artificial Intelligence Research*, 1994, vol. 2, no. 1, p. 263–286. DOI: 10.1.1.72.7289

[38] HASTIE, T., TIBSHIRANI, R. Classification by pairwise coupling. In *Proceedings of the Conference on Neural Information Processing Systems*. 1998, vol. 26, p. 451–471. DOI: 10.1214/aos/1028144844

[39] HSU, C. W., LIN, C. J. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 2002, p. 415–425. DOI: 10.1109/72.991427

[40] PLATT, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, p. 61–74. DOI: 10.1.1.41.1639

[41] LIN, H. T., LIN, C. J., WENG, R. C. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 2007, p. 267–276. DOI: 10.1007/s10994-007-5018-6

[42] PLATT, J. C. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, Cambridge (USA): MIT Press, 1999, p. 185–208. ISBN: 0-262-19416-3

[43] MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. DOI: 10.1098/rsta.1909.0016.

[44] WICKERHAUSER, M. V. Adapted wavelet analysis from theory to software. *SIAM Review*, 1996. DOI: 10.1137/1038018

[45] TORRENCE, C., COMPO, G. P. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 1998, p. 61–78.

[46] PARIKH, U. B., DAS, B., MAHESHWARI, R. P. Combined wavelet-SVM technique for fault zone detection in a series compensated transmission line. *IEEE Transactions on Power Delivery*, p. 1789–1794. DOI: 10.1109/TPWRD.2008.919395

[47] DU, P., TAN, K., XING, X. Wavelet SVM in reproducing kernel Hilbert space for hyperspectral remote sensing image classification. *Optics Communications*. 2010, p. 4978–4984. DOI: 10.1016/j.optcom. 2010.08.009

[48] RUPING, S. SVM kernels for time series analysis. *Universitätsbibliothek Dortmund*, 2001. DOI: 10.1.1.23.9841

[49] EDUARD, G. B., LI, F., SMINCHISESCU, C. Fourier kernel learning. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2012, p. 459-–473. DOI:10.1007/978-3-642-33709-3_33

[50] CHANG, C. C., LIN, C. J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011, vol. 2, no. 27. DOI: 10.1145/1961189.1961199

[51] HSU, C. W., CHANG, C.C., LIN, C. J. A practical guide to support vector classification. *BJU International*, 2003.

[52] RESEARCH INSTITUTE FOR SECURE SYSTEMS. *Side-Channel Attack Standard Evaluation Board Sasebo-gii Specification*. [Online] Cited 2017-04-08. Available at: http://www.rcis.aist.go.jp/files/special/SASEBO/SASEBO-GII-en/SASEBO-GII_Spec_Ver1.01_English.pdf

[53] NUEESS LAB NORTHEASTERN UNIVERSITY. *TeSCASE - Testbed for Side Channel Analysis and Security Evaluation*. [Online] Cited 2017-04-08. Available at: http://tescase.coe.neu.edu/?current_page=homepage

[54] RESEARCH INSTITUTE FOR SECURE SYSTEMS. *Evaluation Environment for Side-Channel Attacks*. [Online] Cited 2017-04-08. Available at: https://www.risec.aist.go.jp/project/sasebo/

[55] NASSAR, M., SOUISSI, Y., GUILLEY, S., et al. RSM: A small and fast countermeasure for AES, secure against 1st and 2ndorder zero-offset SCAs. In *Proceedings of the Conference Exhibition on Design, Automation Test in Europe (DATE)*. Dresden (Germany), 2012, p. 1173–1178. DOI: 10.1109/DATE.2012.6176671

[56] GUILLEYHO, S. *DPA contest v4*. [Online] Cited 2017-04-08. Available at: http://www.dpacontest.org/v4/rsm_doc.php

[57] PENG, H., LONG, F., DING, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, p. 1226–1238. DOI: 10.1109/TPAMI.2005.159

[58] GIERLICHS, B., LEMKE-RUST, K., PAAR, C. Templates vs. stochastic methods. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2006*. Yokohama (Japan), 2006, p. 15–29. DOI: 10.1007/11894063_2

[59] STANDAERT, F. X., MALKIN, T. G., YUNG, M. A unified framework for the analysis of side-channel key recovery attacks. In *Proceedings of the 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques*. Berlin (Germany), 2009, p. 443–461. DOI: 10.1007/978-3-642-01001-9_26

About the Authors . . .

Shourong HOU was born in Shandong, China. He received his B.Eng. from Xidian University in 2015. His research interests include machine learning and side channel attack. He is now a Ph.D. candidate at Shanghai Jiao Tong University.

Yujie ZHOU was born in Zhejiang, China. She received his Ph.D. from University of Science and Technology of China in 1997. Her research interests include digital integrated circuit design, embedded system, trusted computing and digital copyright protection.

Hongming LIU was born in Jiangxi, China. He received his Ph.D. from Shanghai Jiao Tong University in 2014. His research interests include cryptographic chip design, blockchain and machine learning.

Nianhao ZHU was born in Jiangshu, China. She received his Ph.D. from Shanghai Jiao Tong University in 2014. His research interests include cryptographic chip design and side channel attack.