

EFFECTIVE FEATURE EXTRACTION METHOD FOR SVM-BASED PROFILED ATTACKS

Ngoc Quy TRAN

Faculty of Electronics and Telecommunications

Academy of Cryptography Techniques

HaNoi, VietNam

e-mail: quytn@actvn.edu.vn, quang27269@gmail.com

Junbeom HUR

Department of Computer Science and Engineering

Korea University

Seoul 02841, Korea

e-mail: jbhur@korea.ac.kr

Hieu Minh NGUYEN*

Institute of Cryptographic Science and Technology

HaNoi, VietNam

e-mail: hieuminhmta@gmail.com

Abstract. Nowadays, one of the most powerful side channel attacks (SCA) is profiled attack. Machine learning algorithms, for example support vector machine, are currently used for improving the effectiveness of the attack. One issue when using SVM-based profiled attack is extracting points of interest, or features from power traces. So far, studies in SCA domain have selected the points of interest (POIs) from the raw power trace for the classifiers. Our work proposes a novel method for finding POIs that based on the combining variational mode decomposition (VMD) and Gram-Schmidt orthogonalization (GSO). That is, VMD is used to decompose

* Corresponding author

the power traces into sub-signals (modes) of different frequencies and POIs selection process based on GSO is conducted on these sub-signals. As a result, the selected POIs are used for SVM classifier to conduct profiled attack. This attack method outperforms other profiled attacks in the same attack scenario. Experiments were performed on a trace data set collected from the Atmega8515 smart card run on the side channel evaluation board Sakura-G/W and the data set of DPA contest v4 to verify the effectiveness of our method in reducing number of power traces for the attacks, especially with noisy power traces.

Keywords: Points of interest, profiled attack, side channel attack, support vector machine, variational mode decomposition, Gram-Schmidt orthogonalization

Mathematics Subject Classification 2010: 94A60

1 INTRODUCTION

Side channel attack is one of the most powerful cryptanalysis technique for revealing secret key or sensitive information stored on cryptographic devices. The conducting of SCA is based on the analyzing of unintended side channel leakages observed from the devices during cryptographic algorithms run on. There are so many forms of the observed leakages, but the time of operation, the power consumption of the devices, or electromagnetic radiation are the most common uses. SCAs based on the power consumption are known as the power analysis attacks first proposed by Kocher et al. in the late 1990s [1]. These attacks rely on the physical nature of instantaneous power consumption of a cryptographic device depends on the data being processed and the operation being executed. This dependency can be used to expose the data that contains secret key of a cryptographic device.

Depending on the knowledge of attacker about the device under attack as well as the statistical method of analysis and extraction of information from the power consumption traces, SCAs are classified in to two main classes: non-profiled attacks and profiled attacks. Differential Power Analysis (DPA) [1], Correlation Power Analysis (CPA) [2], Mutual Information Analysis (MIA) [3] attacks belong to the first class. These are considered as effective attack methods when the attacker has only an attack device and information of its implementation. The profiled attacks are used when the attacker has the same device as the attack device with full control over. By this device, the attacker is able to accurately characterize the power consumption of the device that results in attack efficiency is much higher than non-profiled attacks in the number of power consumption measurements needed for success in revealing secret key.

So far, there is a lot of attention on profiled attack in SCA research community. The first one is called template attack, as proposed in [4] by Chari et al., relies on assumption that power consumption characteristic follows multivariate

Gaussian distribution. However, in general, this assumption should be not met, machine learning techniques are introduced to conducting profiled attacks. Consequently, several works have applied machine learning techniques to profiled SCA attacks [5, 6, 7, 8, 9]. These works all indicate that machine learning based profiled attacks are more efficient and SVM is commonly used as machine learning algorithm.

Machine learning based profiled attacks relax the need for probability distributions of side channel leakage traces but still require specific extraction techniques to identify points of interest (POIs) on the traces or feature selection in machine learning domain. In SCA, POIs are time sample points from the power traces that correspond to the calculation of the sensitive variables being targeted and their values change according to those variables [10]. The POIs selection, as input features to machine learning algorithms is critical for two main reasons as follows:

1. the power traces are usually acquired by a measurement equipment with high sampling rates and so consist of a large amount of time samples. However, often only a relatively small range of these time samples is informative or statistically dependent on a sensitive target variable;
2. power traces are considered as highly multi-dimensional data that results in the curse of dimensionality issues with machine learning algorithms. That is, computational and runtime complexity for them to solve a task increase.

Therefore, POIs selection is critical to the effectiveness of the profiled attacks. The more precisely the POIs are selected, the better the ability to characterize the power consumption of a profiled device, resulting in increasing attack efficiency and vice versa. Our work focuses on a method for finding POI for SVM-based profiled attacks.

1.1 Related Works

Some studies in the side channel community focus on methods of finding POIs for profiled attacks, which can be classified into four classes: filter methods, dimensionality reduction method, wrapper and hybrid methods, and machine learning based methods. In filter methods, POIs selection process operates on the base of computation of some sample-wise statistics, whose aim is to quantify a sort of signal strength. The signal-strength estimates are derived from classical side channel attack distinguishers computed under the right key hypothesis, such as the Difference of Means [4] or Correlation Power Analysis (CPA) [11]. Other deployed estimates are the Sum of Squared Differences [12], the Signal-to-Noise Ratio [13, 14], and the Sum of Squared t-differences, corresponding to the t-test [12]. Once the chosen signal-strength estimate is computed, all time samples for which the signal strength is higher than a certain threshold are selected as POIs. Of these, the POIs selection method based on CPA estimates is the most common use.

Principal component analysis, as dimensionality reduction method, is another technique for POIs selection. The time samples on traces have the maximize the variability in the projection space of PCA are remained as POIs. So far, the effectiveness of PCA-based profiled attack is not clear, as reported in [15, 16, 17] works. Indeed, selecting the number of retained components as well as the threshold of determination in PCA process is also not an easy task.

Profiled attacks, were presented in [18], used the wrapper method for finding of POIs. In the wrapper method, subsets of time samples on the power traces are evaluated by the prediction performance of a classifier and the subset has the best performance is selected as POIs. To reduce the number of subsets of the wrapper, hybrid method is used. That is, candidate features are first selected by a filter then furthered refined by an accuracy wrapper. As claimed in [18], wrapper and hybrid methods gave slightly better results. The issue with this approach is that computational complexity and search space increase exponentially as the length of trace increases.

Because of the capability of machine learning algorithms in determining the most informative features from raw data inputs, machine learning algorithms can be used to finding POIs of power trace. In the first work in this approach [19], SVM has been trained and the sample points of trace which correspond to highly absolute value of weights are selected as POIs. This method is also called normal-based feature selection and strongly recommended by authors in [19].

1.2 Related Works

As yet, to our knowledge, there are only few works on finding POIs with noisy traces. Furthermore, there have been no more studies on feature engineering in the machine learning domain as applied to profiled attacks. For noisy traces, the authors in [10] claim that the goodness of POIs selections depends significantly on the noise level: as noise level increases the goodness of POIs selection decreases, while at the same noise level, CPA estimation based POIs selection method is the best. This drawback of the POIs selection method is confirmed by the authors in [18] regarding the wrapper and hybrid method.

Inspired by the success of VMD [21] in feature engineering in machine domain, in this work, we propose the method of combining VMD and GSO to find the POIs of power traces. That is, VMD is used to decompose a trace into sub-signals, or modes and POIs are selected from these modes by using GSO as the filter feature selection method. Then, the selected POIs are used for SVM-based profiled attack. We denote SVM_{VMD} for our proposed attack. To demonstrate the efficiency of our proposed attack method, we compare our method to two other SVM-based profiled attacks using the SVM classifier. The first attack uses CPA as the POIs selection method as in [5], so called SVM_{CPA} and is currently considered to be the best method, and the second one uses a normal-based feature selection method as in [19], so called SVM_{NB} . We also investigate the effectiveness of our method with noisy power traces, which often happens in the real attack scenarios. Ours contributions are as follows:

First, we investigate the ability of combining VMD and GSO for finding POIs of the power traces.

Second, we propose an SVM-based profiled attack method that uses our POIs selection method. This is a different approach for conducting profiled attack and it is efficient for noisy power traces.

The remainder of the paper is structured as follows: In Section 2, we describe the background to this research: the profiled attacks, variational mode decomposition and Gram-Schmidt Orthogonalization and SVM. In Section 3, we present our proposed SVM-based profiled attack. The experiments and their results are presented in Section 4. Finally, the main conclusions of our research are presented in Section 5.

2 BACKGROUND

2.1 Profiled Attack

For profiled attack, the attacker must have a device with full control over that is similar attack device. This device is called profiling device and used for leakage information characterization by the attacker. In this work, an attack device that runs a block cipher is used for our attack scenario and leakage is in the form of power consumption. The implementation of profiled attack consists of two phases: profiling phase on profiled device and attack phase on attack device.

In the profiling phase, a dataset of N_p profiling traces is acquired from the profiled device. The dataset is seen as the realization of the random variable $S_p \equiv \{(x_1, z_1), \dots, (x_{N_p}, z_{N_p}) \sim Pr[\mathbf{X}|Z]^{N_p}$, where x_i are the traces obtained from the device processing the respective intermediate values $z_i = \varphi(P, K)$. Based on S_p , a model is built to characterize the side channel leakage of the cryptographic device for each hypothetical value z_i . This can be modeled as $F(X|Z) : \mathcal{X} \rightarrow P(\mathcal{Z})$.

In the attack phase, a dataset of N_a attack traces are acquired from the target device. The dataset is seen as a realization of $S_a \equiv (k^* \{(x_1, z_1), \dots, (x_{N_a}, z_{N_a})\})$ such that $k^* \in \mathcal{K}$, and for all $i \in [1, N_a]$, $p_i \sim Pr[P]$ and $x_i \sim Pr[\mathbf{X}|Z = \varphi(p_i, k^*)]$. Subsequently, a prediction vector is computed for each attack trace, based on a previously built model: $y_i = F(x_i), \forall i \in [1, N_a]$. A score, for example the probability, is assigned to each trace for each intermediate value hypothesis z_j , with $j \in [1, |Z|]$. The j -value of y_i describes the probability of z_j according to the model when the attack trace is x_i . These scores are combined over all the attack traces to output a likelihood for each key hypothesis, and the candidate with the highest likelihood is predicted to be the correct key. The maximum likelihood score can be used for prediction. For every key hypothesis $k \in \mathcal{K}$, this likelihood score is defined by Equation (1) with the key assigned the highest score predicted as being the most likely.

$$d_{S_a} = \prod_{i=1}^{N_a} y_i[z_i] \quad \text{where} \quad z_i = \varphi(p_i, k). \quad (1)$$

2.2 Variational Mode Decomposition

VMD is a method used to decompose a real valued signal into narrowband sub-signals, also known as intrinsic mode functions (IMFs) or simply VMD modes [20] by Equation (2). In that $x(t)$ is the original signal and $u_k(t) = A_k(t) \cos(\phi_k(t))$, called the k^{th} mode, is the amplitude-modulation and frequency-modulation signal where $A_k(t)$ is the slowly varying, positive envelope and $\phi_k(t)$ is the phase. Each mode has a central frequency f_k that its instantaneous frequency $\phi'(k)$ varies around.

$$x(t) = \sum_{i=1}^K u_k(t). \tag{2}$$

The finding simultaneously a set of modes and their central frequencies by VMD is done by solving the optimization problem given by expression (3). This is the constrained minimization process of sum of all mode's bandwidth. The bandwidth of each mode is estimated by 3 steps:

1. compute the analytic signal of each mode by using Hilbert transform so its spectrum is positive;
2. multiply the analytic signal with a complex exponential for shifting its frequency spectrum to baseband;
3. compute the squared 2-norm of the gradient of the baseband signal.

$$\min_{u_k, f_k} \left\{ \sum_k \left\| \frac{d}{dt} \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j2\pi f_k t} \right\|_2^2 \right\} \quad \text{s.t.} \quad \sum_k u_k(t) = x(t). \tag{3}$$

The solution for (3) is equivalent to finding the optimal point of an unconstrained augmented Lagrangian given by (4) where α is the penalty factor and $\lambda(t)$ is Lagrangian multiplier. And this optimization could be solved by using alternate direction method of multipliers algorithm [21]. All modes of $x(t)$ are computed in frequency domain by (5) and (6) at each iteration of algorithm until the condition (7) is met.

$$\begin{aligned} \mathcal{L} \{u_k(t), f_k, \lambda(t)\} = & \alpha \sum_{k=1}^K \left\| \frac{d}{dt} \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j2\pi f_k t} \right\|_2^2 \\ & + \left\| x(t) - \sum_{i=1}^K u_k(t) \right\|_2^2 + \left\langle \lambda(t), x(t) - \sum_{k=1}^K u_k(t) \right\rangle, \end{aligned} \tag{4}$$

$$U_k^{n+1}(f) = \frac{X(f) - \sum_{i < k} U_k^{n+1}(f) - \sum_{i > k} U_k^n(f) + \frac{\Lambda^n}{2}(f)}{1 + 2\alpha \{2\pi (f - f_k^n)\}^2}, \tag{5}$$

$$f_k^{n+1} = \frac{\int_0^\infty |U_k^{n+1}(f)|^2 f df}{\int_0^\infty |U_k^{n+1}(f)|^2 df} \approx \frac{\sum f |U_k^{n+1}(f)|^2}{\sum |U_k^{n+1}(f)|^2}, \tag{6}$$

$$\sum_k \frac{\|u_k^{n+1}(t) - u_k^n(t)\|_2^2}{\|u_k^n(t)\|_2^2} < \epsilon. \tag{7}$$

After the convergence of this optimization, the inverse Fourier transform is applied to (5) to obtain the waveform of each mode. Because of the combination of Wiener filtering, Hilbert transform and ADMM in VMD, VMD modes are highly accurate in describing the different components of the original signal and robust to noise.

2.3 SVM

SVM algorithms [22] is used to construct classifiers. The basic form of SVM is the binary classifier which can classify two class by the largest-margin separating hyperplane between them. Let the training data set be $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the inputs data point and $y_i \in \{+1, -1\}$ is the corresponding label of x_i . This data set is assumed to be linearly separable. The hyperplane separates two class is given by expression (8).

$$f(\mathbf{x}) = \omega^T \mathbf{x} + b \tag{8}$$

where $\omega \in \mathbb{R}^d$ is the weights and $b \in \mathbb{R}$ is the bias. The margin is calculated as the minimum distance from all data points to this hyperplane, given by (9).

$$\min_n \frac{y_n(\omega^T \mathbf{x} + b)}{\|\omega\|_2}. \tag{9}$$

Because there are many hyperplanes separating two classes, the objective of SVM is finding one of hyperplanes that maximize the margin. So, the optimization of SVM is given by (10).

$$\operatorname{argmax}_{\omega, b} \left(\min_n \frac{y_n(\omega^T x_n + b)}{\|\omega\|_2} \right) = \operatorname{argmax}_{\omega, b} \left\{ \frac{1}{\|\omega\|_2} \min_n y_n(\omega^T x_n + b) \right\}. \tag{10}$$

This optimization problem can be converted to (11) because the margin does not change when scaling ω and b .

$$(\omega, b) = \operatorname{argmin}_{\omega, b} \frac{1}{2} \|\omega\|_2^2 \quad \text{s.t.} \quad 1 - y_n(\omega^T x_n + b) \leq 0, \forall n = 1, 2, \dots, N. \tag{11}$$

The quadratic programming should be used to solve (11) directly. However, the solution is complicated when the dimensionality and the number of data points increase. Then, the above problem can be usually solved by Lagrange multipliers method through its duality problem. The Lagrangian of (11) is given by (12), and its duality function is given by (13).

$$\mathcal{L}(\omega, b, \lambda) = \frac{1}{2} \|\omega\|_2^2 + \sum_{n=1}^N \lambda_n (1 - y_n (\omega^T x_n + b)) \tag{12}$$

where $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_N]^T$ and $\lambda_n \geq 0, \forall n = 1, 2, \dots, N$.

$$g(\lambda) = \min_{\omega, b} \mathcal{L}(\omega, b, \lambda), \text{ v } \lambda \geq 0,$$

$$g(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m y_n y_m x_n^T x_m. \tag{13}$$

Combining the dual function of the Lagrangian and the constraints of λ , the duality problem (11) is given by (14).

$$\lambda = \underset{\lambda}{\operatorname{argmax}} g(\lambda) \quad \text{s.t.} \quad \sum_{n=1}^N \lambda_n y_n = 0, \lambda > 0. \tag{14}$$

This is the convex optimization problem and satisfies the dual condition, so its solution will satisfy the Karush-Kuhn-Tucker (KKT) conditions with variables ω, b, λ . After finding the λ by KKT conditions, the pair (ω, b) is determined by the expression (15) and (16).

$$\omega = \sum_{\mathcal{S}} \lambda_m y_m \mathbf{x}_m, \tag{15}$$

$$b = \frac{1}{N_\delta} \sum_{n \in \delta} y_n - \omega^T x_n = \frac{1}{N_S} \sum_{n \in \delta} \left(y_n - \sum_{m \in \delta} \lambda_m y_m x_m^T x_n \right) \tag{16}$$

where $\mathcal{S} = \{n : \lambda_n \neq 0\}$ is the set of support vectors and N_S is the number of elements in \mathcal{S} . As the final, the class determination of a new data point is calculated by (17).

$$f(x) = \operatorname{sign} \left(\sum_{m \in \delta} \lambda_m y_m \mathbf{x}_m^T \mathbf{x} + \frac{1}{N_S} \sum_{n \in \delta} \left(y_n - \sum_{m \in \delta} \lambda_m y_m \mathbf{x}_m^T \mathbf{x}_n \right) \right). \tag{17}$$

In order to generalize well unseen data or work with linearly inseparable data, soft-margin SVM is introduced. The key idea is the using of slack variables ξ_i

to allow misclassification of difficult or noisy data points. Then, the optimization problem (4) can be rewritten by (18).

$$\begin{aligned}
 (\omega, b, \xi) = \operatorname{argmin}_{\omega, b, \xi} & \frac{1}{2} \|\omega\|_2^2 + C \sum_{n=1}^N \xi_n & (18) \\
 \text{s.t.} & 1 - \xi_n - y_n (\omega^T x_n + b) \leq 0, \forall n = 1, 2, \dots, N, \\
 & \xi_n \geq 0, \quad \forall n = 1, 2, \dots, N,
 \end{aligned}$$

where the parameter C is the penalty error that controls the trade-off between the training error and the margin width. To solve (16), the same above approach should be used.

In real application, the data set is almost nonlinear, SVM can be extended to work well by using a function $\phi(\cdot)$ that transforms the data point to higher dimensional data that is linear separable [23]. To avoid calculation of this function for all data points, the kernel function $k(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$ is used. Some popular kernel functions are listed in Table 1.

Type	$k(\mathbf{x}, \mathbf{z})$	Parameters
Linear	$\mathbf{x}^T \mathbf{z}$	
Polynomial	$(r + \gamma \mathbf{x}^T \mathbf{z})^d$	d : degree, r : coef0, γ : gamma
RBF (Radial Basis Function)	$e^{-\gamma \ \mathbf{x} - \mathbf{z}\ ^2}$	γ : gamma > 0

Table 1. Popular kernel functions

The output of an SVM classifier is the label of the input data point. However, in some cases, the probability that the data point x belongs to a label needs to be determined. To solve this problem, Platt [24] proposed a method of calculating this probability value by (19) using the sigmoid function.

$$\Pr(y = 1 \mid x) = \frac{1}{1 + \exp(Af(x) + B)}. \tag{19}$$

In which $f(x)$ is the hyperplane generated by SVM training process. Parameter pairs (A, B) are determined through minimal cross-entropy function by a different training data set than the one used to train SVM to avoid over-fitting.

The SVM classifier solves the binary classification problem and it can be extended to solve the multi-class classification problem with two main strategies as follows:

- One-vs-one strategy [24]: in this strategy multiple binary classifiers are constructed for each possible pair of classes. This results in $M(M - 1)/2$ classifiers for M -classes data set. All classifiers are used to predict the label of a new data point and the final decision is the label in which is the most voting.

- One-vs-all strategy [25]: In this strategy, it is necessary to construct M classifiers for M -classes data set in discriminating one class from the rest. The predicted label for a new data point is based on the outputs of M classifiers with winner takes all decision rule.

2.4 GSO-Based Feature Selection

As mentioned above, the finding of POI on the power trace is also known as the feature selection in machine learning domain. The method of feature selection is in two forms of wrapper and filter. In this paper, Gram-Schmidt orthogonalization based feature selection method is used to select the POI of the traces. This method is in the form of filter method, independent of the advance classifiers and is effective in ranking the features contained in the traces based on criteria computed directly from the traces. Indeed, this method allows the features to be determined without weighting all of features in the traces. It ranks features based on the correlation between features and the output target of a prediction model or the pre-assigned label of features. Let $x_k = [x_{k1}, x_{k2}, \dots, x_{kN}]^T$ be the k^{th} feature vector of N instances, $y = [y_1, y_2, \dots, y_N]^T$ be the output target and Q is the number of features. This results in (N, Q) matrix feature data set. To define the relation between each feature and output target, the correlation is calculated by (20) [26].

$$\cos(\alpha_k) = \frac{\langle \mathbf{x}_k, \mathbf{y} \rangle}{\|\mathbf{x}_k \cdot \mathbf{y}\|}. \quad (20)$$

In formula (20), x_k is a column vector containing N values of the k^{th} feature in all Q features, α_k is the angle of vectors x_k and y . If they are perpendicular to each other, the cosine of α_k equals 0 meaning there is no correlation between them, whereas when the angle between them becomes smaller, this correlation increases and the maximum value is 1 when they are completely correlated.

The GSO-based feature selection process uses the formula (20) to quantify the degree to which features are related to the output target. The first selected feature is the most correlated input features with the output target by the cosine calculation. The next features are selected according to the iteration process as follows until all input features are ranked, or until a stopping condition is met [26]:

1. the rest input features and output target are projected on the subspace orthogonal to the selected feature;
2. the cosine calculations are done on this subspace for all projected features and target output to find out the most correlated feature. This feature is added to selected feature list.

3 THE PROPOSED METHOD

In this part, we present our proposed SVM-based profiled attack that uses the combination of VMD and GSO for POIs selection of power traces.

3.1 SVM-Based Profiled Attacks

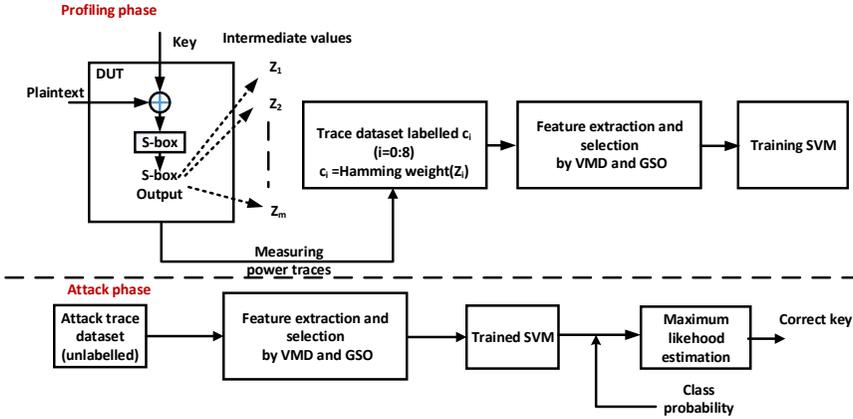


Figure 1. SVM-based profiled attacks framework

The proposed SVM-based profiled attack, shown in Figure 1, is carried out in two phases: a profiling phase and an attack phase. In the profiling phase, power traces are collected from the profiled device while it is executing a cryptographic algorithm to form a trace data set. This trace data set is labeled according to the Hamming weight of targeted value of the algorithm that needs to be profiled Z_1, \dots, Z_m . Usually these targeted values are taken at the output of the S-box. Because, they are 8-bit values that result in 9 Hamming weight classes from 0 to 8 denoted as: c_0, c_1, \dots, c_8 . This labeled set of traces is fed to the feature extraction and selection block for mapping traces into feature space and the best features are selected. These selected features are considered as POIs of the power trace in feature space that should describe the statistical dependency of the Hamming weight of the targeted value Z_i with the power consumption. In the final step of the profiled phase, POIs of all traces are used to train SVM to model the power consumption characteristic of the profiled device. For training SVM classifier, its parameters are selected as follows: the kernel function is RBF, the penalty factor and width of kernel of RBF are optimized by Grey wolf optimization algorithm presented in [27].

During the attack phase, unlabeled traces collected from the attack device are fed into the feature extraction and selection block to select POIs and they are next classified by the trained SVM model to determine the probabilities of the traces for classes c_0, \dots, c_8 . Finally, we compute the log likelihood for each hypothesis value

of the key byte that is used by attack device as follows:

$$\log L_k \equiv \log \prod_{i=1}^{N_a} P_{SVM}(X_i | c_i) = \sum_{i=1}^{N_a} \log P_{SVM}(X_i | c_i) \quad (21)$$

where k is a hypothesis key byte value, $c_i = \text{Hammingweight}(Sbox(p_i, k))$, p_i is the plaintext associated with trace X_i , and the number of attack traces is N_a . The key k_c that maximizes the log likelihood in (22), as given below, is predicted to be the correct key.

$$\underset{k_c}{\operatorname{argmax}} \log L_k. \quad (22)$$

3.2 Feature Extraction and Selection

Features or POIs selection is critical to the effectiveness of the profiled attacks. The more precisely the features are selected, the better the ability to characterize the power consumption of a profiled device, resulting in increasing attack efficiency and vice versa. This section presents a new method for finding features of power traces for SVM-based profiled attack. First, power trace characteristics are discovered as follows:

The power trace collected during the operation of a cryptographic device describes its power consumption. It consists of many components in which dynamic power dissipation is the most important [13]. This component depends on the processed data of the circuit and is useful information leakage for power analysis attack. The dynamic power dissipation mainly caused by the switching activity of logic gates in a circuit which is controlled by the operating clock frequency so the dynamic power consumption is driven by the clock frequency of circuit. Therefore, in spectrum of power trace, it is expected that the clock frequency component has significant magnitude compared to the other components. The information leakage is nearly in the form of a both amplitude and frequency modulation signal and the central frequency of its spectrum is the clock frequency. Generally, in a device, the different parts of its circuit are controlled by different operating frequencies through the clock division system, so the dynamic power dissipation is the combination of some amplitude – frequency modulation signals with different center frequencies. So, if it is possible to separate the dynamic power dissipation to amplitude – frequency modulation signals with different center frequencies, one of these signals contains significant information leakage related to target circuit part while other does not.

As a result, the feature extraction process from power trace should ensure:

1. the remaining features contain the most important information of the trace which is the dynamic power dissipation caused by the targeted circuit;
2. it could remove the other components of power traces;
3. it could reduce noise in the power traces.

Fortunately, these requirements can be fulfilled by using VMD method. That is because VMD decomposes a trace into different components and it is robust to sampling and noise.

In our proposed method, VMD is used for extracting features from power traces. VMD decomposes the signal into sub-signals, called VMD mode in this paper, which are amplitude-modulated frequency-modulated signals so each mode contains a specific frequency spectrum with different center frequency. So, the VMD mode which center frequency relates to clock frequency could be use as feature of the power trace. Indeed, VMD can discover signal changes more accurately so that features of power traces can be recognized more accurately. Moreover, VMD is robust to noise because of using Wiener filter technique. Thus, VMD should be useful for using noisy traces.

Unfortunately, VMD mode still contains redundant features which are not related to target variable that has been profiled. Therefore, they must be eliminated for increasing the generalization capability of the classifier and reducing the volume of training data. The elimination of redundant features is known as the feature selection. In previous related works, all features that are higher than a certain threshold are selected. In this paper, we recommend using GSO to selection feature of selected VMD mode.

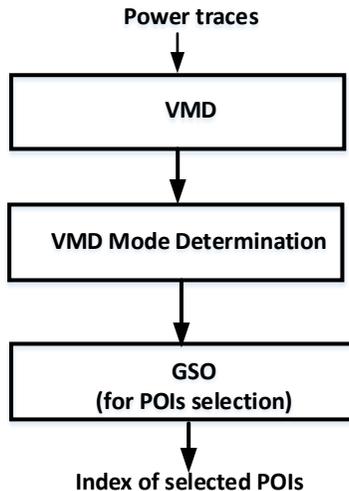


Figure 2. Feature selection procedure of power trace

To sum up, there are three phases in the proposed features extraction and selection method, as illustrated in Figure 2. First, VMD is used to decompose original traces to VMD modes. In the VMD process, it is necessary to set parameters. The number of decomposed modes K : VMD needs to preset the number of decomposed modes K . If K is too small, the decomposed modes are too few, and all the de-

composition modes cannot be captured; while if the value of K is too large, the interfering signal will be over decomposed such that the center frequencies of modes will be mixed. The penalty factor, α affects the bandwidth of the decomposed signal. To decompose traces by the VMD, the number of IMFs (K) and the quadratic penalty factor (α) should be determined beforehand. In this study, the parameters, K and α , were determined according to the following steps:

Step 1. Decompose a power trace into modes for different $K = [1, 20]$ and $\alpha = [5, 2000]$.

Step 2. Add up the modes for each of the K and a values to obtain the reconstructed power trace and estimate the values of Pearson correlation coefficient for the reconstructed and original power trace.

Step 3. Select the sets of K and a values for maximum of Pearson correlation coefficient.

For others input parameters of VMD: update rate, τ and convergence condition ϵ are selected by standardization values in range $0 : 1e-6$ [20].

In VMD mode selection phase, in order to determine which VMD mode has frequency range that relates to frequency of operating clock our attack device and this mode can be used as features of the power trace, we conduct correlation power analysis (CPA) attacks on all the VMD modes. Based on the results of CPA attacks, the VMD mode that has the largest correlation coefficient is selected.

In GSO feature selection phase, it is necessary to set number of selected features, N . Our principle of finding the value of N is to find a trade-off between the accuracy and the computation cost or execution time. So, the value of N that SVM has the highest accuracy together with the lowest execution time is selected.

4 EXPERIMENTS

In this section, we show the experimental results of implementing profiled attacks with the proposed new SVM_{VMD} approach, which is based on SVM and the combining of VMD and GSO for feature extraction and selection. We compared the effectiveness of the proposed method with the two profiled attacks based on SVM with points of interest selection by CPA in [5] called SVM_{CPA} , and the normal-based feature selection method in [19] called $SVMSVM_{NB}$. The following parameters are used to evaluate the effectiveness of an attack:

The ability to reveal the correct key: To confirm that our profiled attacks can reveal the correct key used by AES-128, we figure out the probability of each key being the actual key used. The key with highest probability is the most likely one.

Guessing entropy [28]: This score is also known as average rank of correct key is widely used to rate the effectiveness of side channel attack according to number

of attack traces. By conducting the attack several times independently, guessing entropy is calculated as follows:

1. the rank of correct key in all guessing keys are computed. This is the index of correct in list all ranked key;
2. calculate the average indexes of the correct key.

In the paper, this guessing entropy has estimated over 10 independent attacks.

4.1 Dataset

Dataset 1. The set consists of 60 000 traces collected while AES-128 processed intermediate values at S-box output. AES-128 was implemented on a Smartcard Atmega8515 run on Sakura G/W. A sample of one of the collected power traces has 2500 time-samples which is titled ‘Original trace’ in Figure 3.

Dataset 2. This data set consists of 100 000 traces downloaded from public DPA contest v4 website at: <http://www.dpacontest.org/v4>. There is 4000 time-samples in a trace of a first-order masked AES implementation which the output of S-box is $Sbox(P_i + k^*) * M$, where M is a mask [29]. When the mask values are known, this data set are considered as an unmasked case.

4.2 Results

4.2.1 Feature Selection Phase

In this section, we investigated the effect of the feature selection on the classification accuracy of the proposed method. First, VMD is used to decompose original traces to VMD modes. For VMD, two main parameters: the number of VMD modes (K) and penalty factor (α) are initialized with $K = 5$, $\alpha = 1000$ according to procedure as described in Section 3.2. The VMD modes of both Dataset 1 and Dataset 2 are depicted in Figures 3 and 4. As expected, VMD modes contain the different components of the original signal at different central frequencies. In order to determine which VMD mode has frequency range that relates to frequency of operating clock our attack device and this mode can be used as feature of the power trace, we conduct correlation power analysis (CPA) attacks on all the VMD modes. Based on the results of CPA attacks, the VMD mode that has the largest correlation coefficient is selected as the feature of the power trace. As results given by Table 2, VMD mode 1 is selected as extraction feature of power trace in Dataset 1, and with Dataset 2 it is VMD mode 2.

Table 3 represents the classification accuracy of SVM on Dataset 1 when extracted features are VMD mode 1 and the selected features are chosen by GSO. Table 4 represents the classification accuracy of SVM on Dataset 2 when extracted features are VMD mode 2 and the selected features are chosen by GSO. The selected features are put into an SVM classifier for the training phase. As the feature

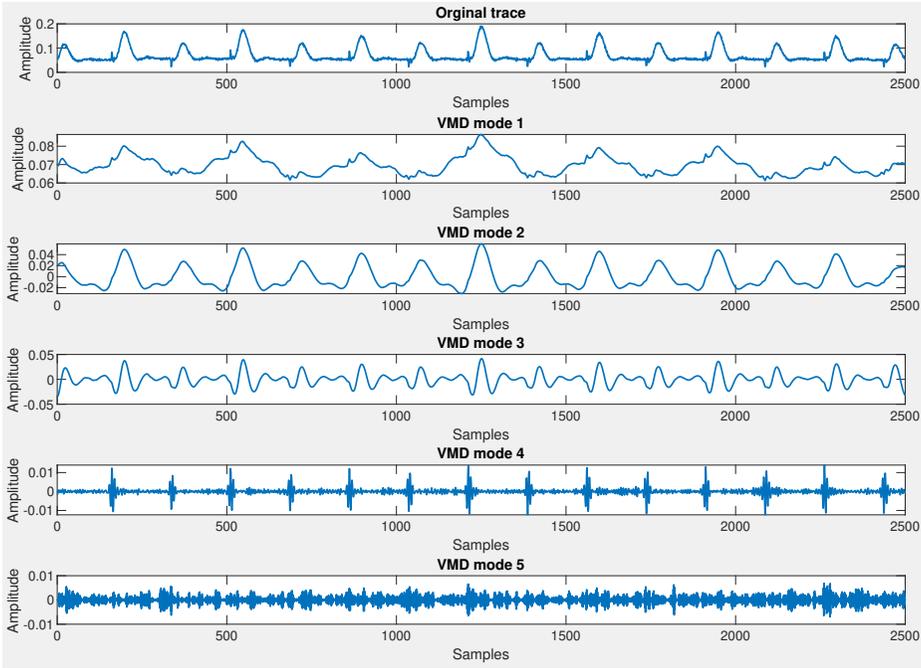


Figure 3. VMD mode of the power trace on Dataset 1

	Dataset 1		Dataset 2	
	Max correlation	Key found	Max correlation	Key found
VMD mode 1	0.64	63 (correct)	0.52	108 (correct)
VMD mode 2	0.62	63 (correct)	0.87	108 (correct)
VMD mode 3	0.54	63 (correct)	0.80	108 (correct)
VMD mode 4	0.37	255 (wrong)	0.37	188 (wrong)
VMD mode 5	0.35	246 (wrong)	0.34	135 (wrong)

Table 2. Results of correlation power attack on VMD modes

dimension increases, so does the accuracy of the classification, but with too many features the accuracy decreases because the features do not generalize the power consumption characteristic well when used by the classifier. Therefore, the subset of features with the highest accuracy and lowest feature dimensions are selected and shown in the bold font.

4.2.2 Key Recovery Phase

In order to verify our proposed SVM_{VMD} profiled attack has the ability to reveal secret key of attack device. In the attack phase, SVM_{VMD} is used to reveal the secret key when classifying 9 hamming weight classes of $S - box$ output. Instead of

Dimensions	Selected Features	Classification Accuracy (%)
2	1 036 509	18.2
4	1 036 509 2261 2262	30.12
6	1 036 509 2261 2262 2263 2260	50.31
8	1 036 509 2261 2262 2263 2260 2264 2265	81.56
10	1 036 509 2261 2262 2263 2260 2264 2265 2259 861	81.78
12	1 036 509 2261 2262 2263 2260 2264 2265 2259 861 2267 1038	89.22
14	1 036 509 2261 2262 2263 2260 2264 2265 2259 861 2267 1038 411 577	95.03
16	1 036 509 2261 2262 2263 2260 2264 2265 2259 861 2267 1038 411 577 886 1687	95.02
18	1 036 509 2261 2262 2263 2260 2264 2265 2259 861 2267 1038 411 577 886 1687 1211 1670	94.27
20	1 036 509 2261 2262 2263 2260 2264 2265 2259 861 2267 1038 411 577 886 1687 1211 1670 1576 216	92.84

Table 3. Acquired results considering extraction of features by VMD and selection by GSO on Dataset 1

Dimensions	Selected Features	Classification Accuracy (%)
2	1 804 3201	22.6
4	1 804 3201 1664 2389	31.89
6	1 804 3201 1664 2389 689 3231	60.38
8	1 804 3201 1664 2389 689 3231 1524 1556	80.24
10	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192	86.66
12	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192 2766 2282	90.35
14	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192 2766 2282 1244 852	95.68
16	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192 2766 2282 1244 852 2392 1797	96.62
18	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192 2766 2282 1244 852 2392 1797 2251 3113	94.58
20	1 804 3201 1664 2389 689 3231 1524 1556 3093 3192 2766 2282 1244 852 2392 1797 2251 3113 3108 1095	90.28

Table 4. Acquired results considering extraction of features by VMD and selection by GSO on Dataset 2

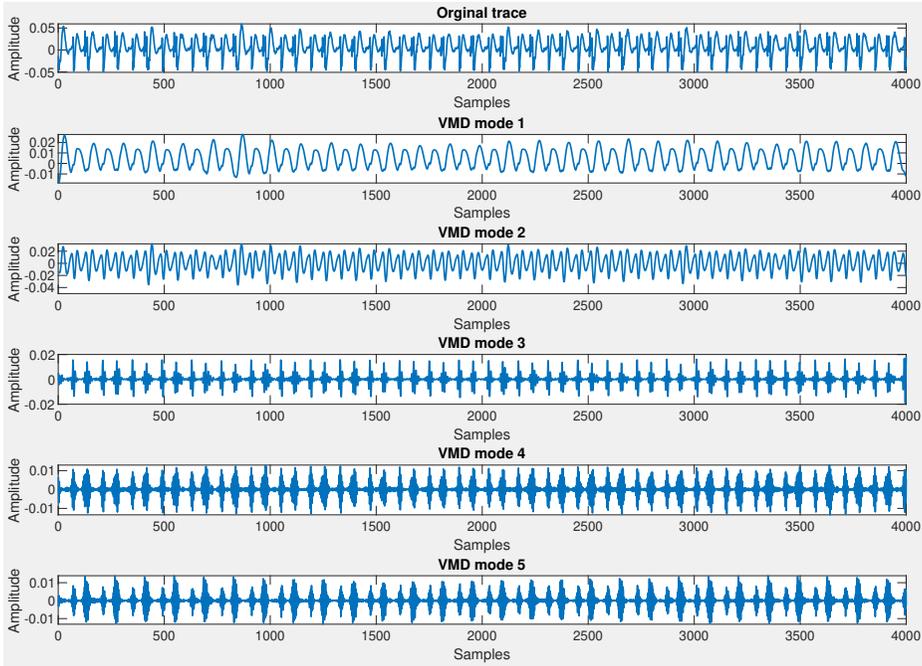


Figure 4. VMD mode of the power trace on Dataset 2

predicting the class HW of each trace, we gave the posterior conditional probability $P_{SVM}(X_i|c)$. The estimated probability of the hypothetical keys is determined by the maximum likelihood estimation. The correct key is defined as the key with the highest probability. For Dataset 1, which was collected in this experiment, the first byte of the AES-128 key is 63, and that is indeed assigned the largest probability value, as depicted in Figure 5. With Dataset 2, the recovery key is 108, identical to the key used to install AES in the DPA contest v4 (Figure 6). These results prove that our attack method could correctly recover the key used by AES-128.

Figures 7 and 8 report the GE corresponding to different numbers of traces used for attacks with Dataset 1 when SVM_{VMD} , SVM_{CPA} and SVM_{NB} are used to predict hamming weight classes. As expected, the GEs of all attacks decrease as the number of traces increases. Moreover, the larger the size of the training set, the lower the GE. The reason for this is that the performance of SVM is determined by its parameters, and the size of the training set is critical to finding the best parameters for the SVM. With Dataset 2, we performed the same experiments as for Dataset 1, and the GE calculated in the attack phases are presented in Figures 9 and 10. The overall performance of all the attacks are the same as those for Dataset 1. Again, SVM_{VMD} achieves the best GE values.

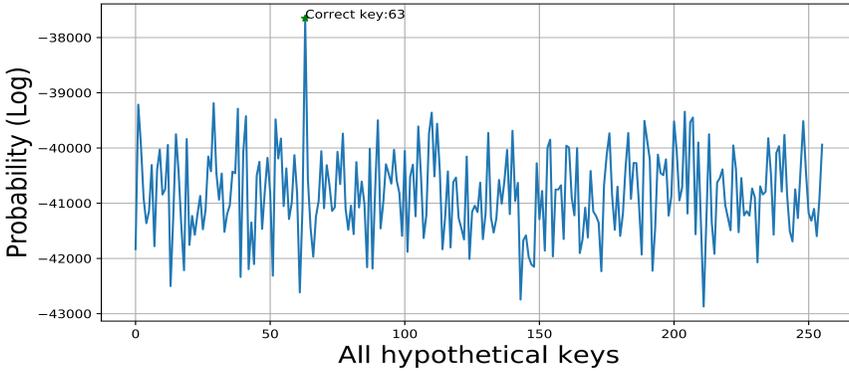


Figure 5. Probability of all hypothetical keys on Dataset 1

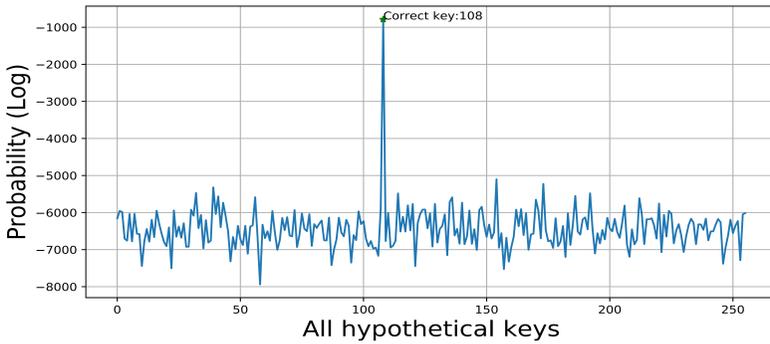


Figure 6. Probability of all hypothetical keys on Dataset 2

In Table 5, for each dataset we give the number of traces required by the profiled attacks based on SVM for guessing entropy to reach 0. SVM-based VMD feature extraction requires the minimum number of traces to recover the key, 10.2 and 5.3 traces on average, corresponding to 100 and 200 profiling traces, respectively. These empirical results indicate that the SVM-based profiled attack with the VMD feature extraction technique is more effective than the attacks with the CPA and normal-based feature extraction techniques. This can be explained by the VMD extraction technique allowing more effective selection of trace characteristics than the CPA and normal-based POI selection methods.

Number of Profiling Traces	Dataset 1			Dataset 2		
	SVM _{VMD}	SVM _{CPA}	SVM _{NB}	SVM _{VMD}	SVM _{CPA}	SVM _{NB}
100	10.2	18.1	17.6	10.3	19.2	18.3
200	5.3	9.2	8.7	4.7	9.4	7.3

Table 5. Number of traces used by the attacks to attain $GE = 0$

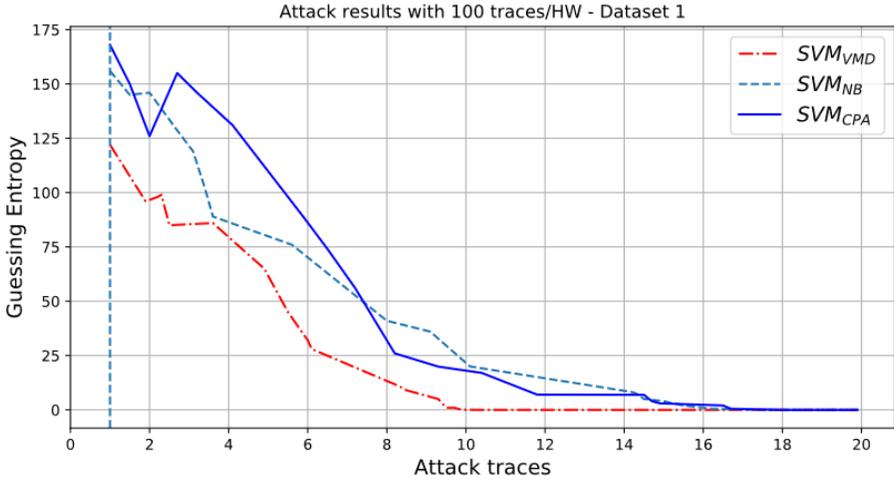


Figure 7. Attack performance with 100 traces/HW class on Dataset 1

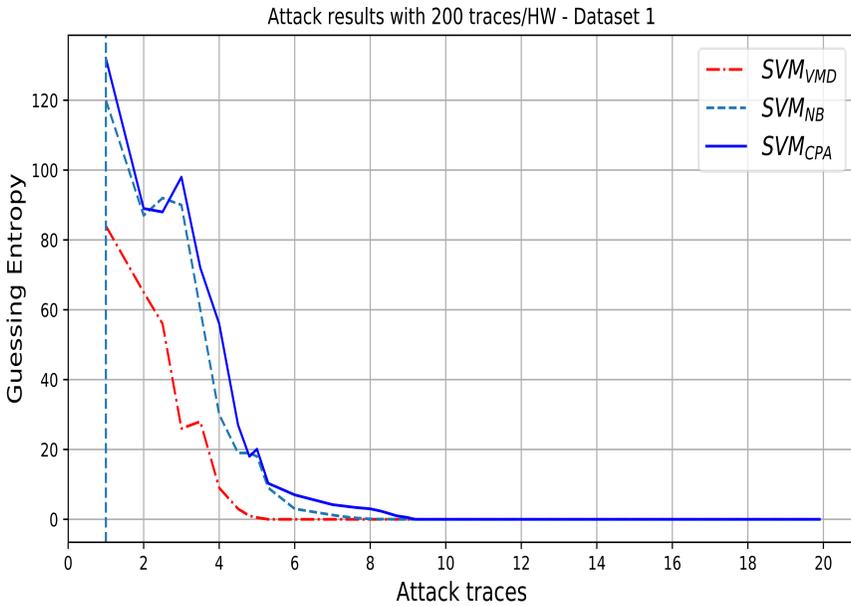


Figure 8. Attack performance with 200 traces/HW class on Dataset 1

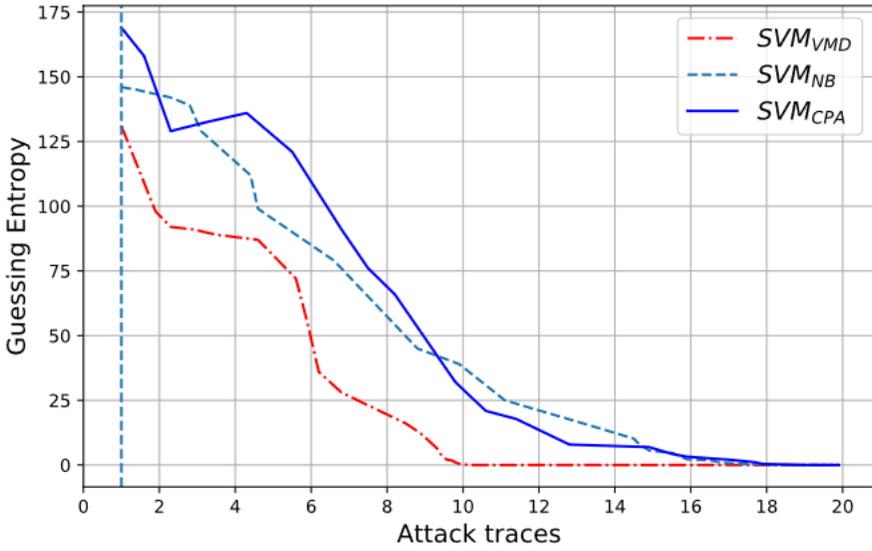


Figure 9. Attack performance with 100 traces/HW class on Dataset 2

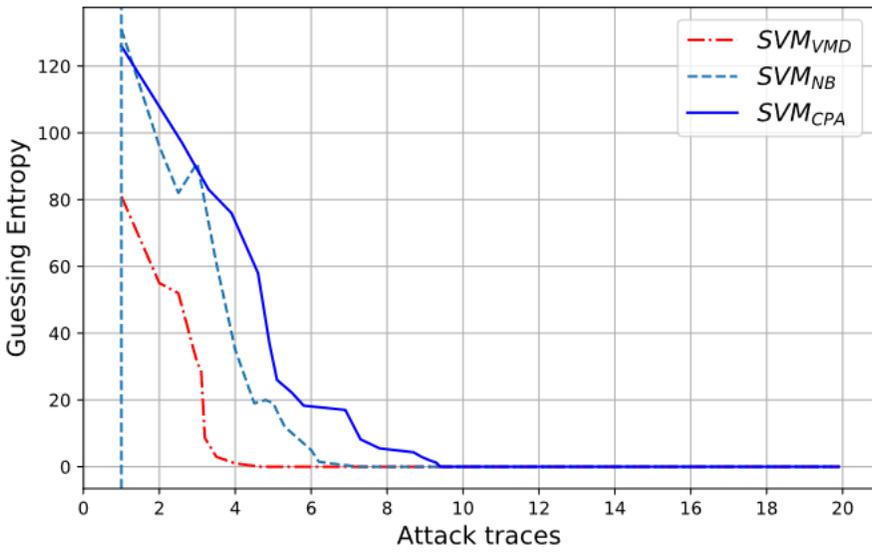


Figure 10. Attack performance with 200 traces/HW class on Dataset 2

4.2.3 Results in the Case of Noisy Traces

The power traces are usually polluted with noise in practice. To examine the effectiveness of our proposed SVM_{VMD} profiled attack in noisy condition, additive Gaussian noise is added to the power traces. In our experiments, two noise level of standard deviation $SNR_1 = 20$ and $SNR_2 = 10$ are added to both Dataset 1 and Dataset 2. In addition, different feature extraction techniques were used for the SVM-based profiled attacks to investigate their effects on the efficiency of the attacks in the presence of noise. Overall, the guessing entropy of all the attacks increase with the level of noise, but the attack based on SVM with combining of VMD and GSO is the least sensitive to noise. The results of our attacks with 200 profiling traces per Hamming weight class, presented in Figures 11, Figure 12, 13 and 14 and Table 6, show that out of SVM_{CPA}, SVM_{NB} and SVM_{VMD}, the proposed method, SVM_{VMD}, has the best performance at both noise levels while SVM_{CPA} and SVM_{NB} are comparable to each other. After adding noise to the power trace, the number of traces required for GE to reach 0 increased by only 25 % approximately with the proposed attack, while it increased by over 100 % for the other methods. This proves that the VMD signal is insensitive to noise so the SVM_{VMD} attack should work well under noisy conditions. This property is very useful in real attack scenarios where collected measurement traces invariably contain noise.

Noise Level	Dataset 1			Dataset 2		
	SVM _{VMD}	SVM _{CPA}	SVM _{NB}	SVM _{VMD}	SVM _{CPA}	SVM _{NB}
$SNR_1 = 20$ dB	7.4	19.0	17.0	6.7	18.8	14.7
$SNR_2 = 10$ dB	8.6	25.7	23.6	9.8	21.6	20.2

Table 6. Number of noisy traces used by the attacks to attain $GE = 0$

5 CONCLUSIONS

In this work, the combining of variational mode decomposition and Gram-Schmidt was proposed as a feature extraction and selection method of power traces. The VMD mode that has central frequency related to clock operation frequency of the attack device can be used as features of power traces and GSO can be used as a feature selection method. Experimental results show that an acceptable classification accuracy can be achieved when SVM classifier uses these selected features as its input. Compared to other SVM-based profiled attacks, the SVM_{VMD} required the minimum number of traces for successful key recovery. Furthermore, SVM_{VMD} is less sensitive to noise so can be used well with noisy power traces. In our opinion, this work suggests a new approach for feature extraction from power traces using variational mode decomposition, and this method should also be tested in combination with other feature selection method and learning algorithms for profiled attacks.

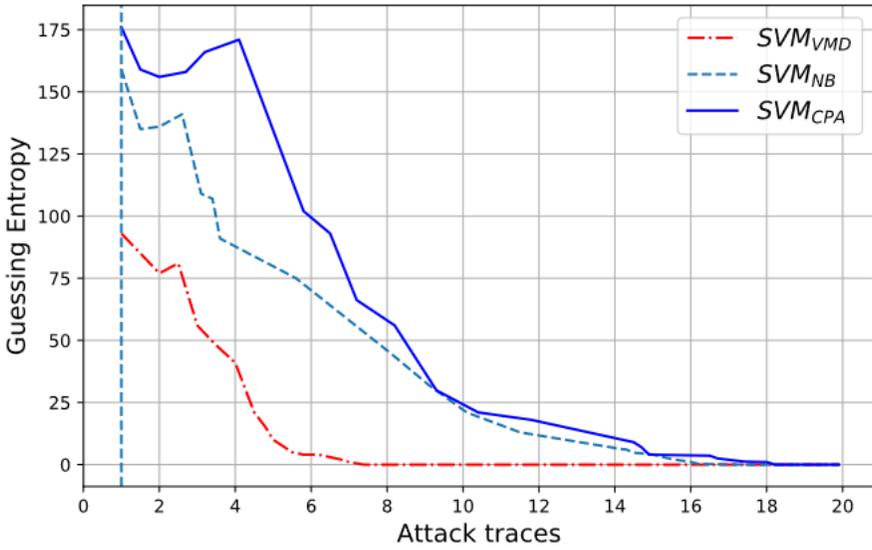


Figure 11. Attack results on Dataset 1 with $SNR_1 = 20$ noise added to power traces

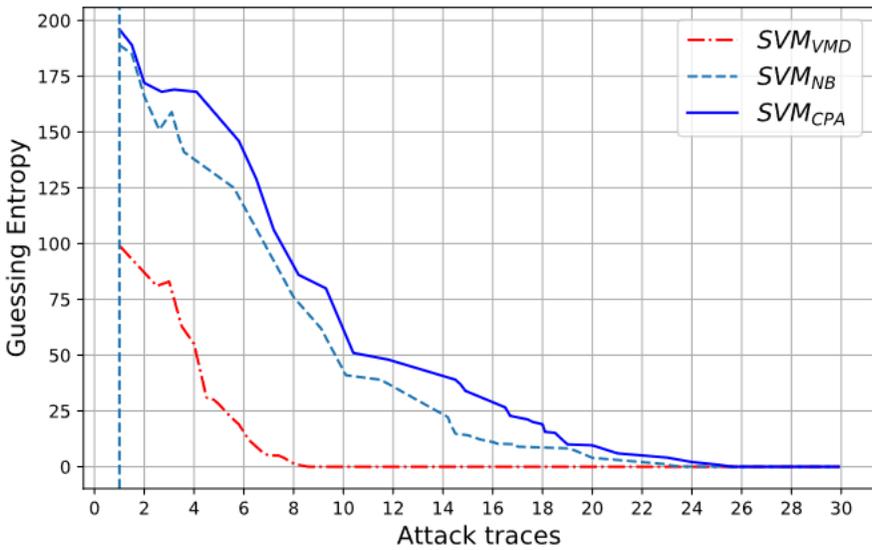


Figure 12. Attack results on Dataset 1 with $SNR_2 = 10$ noise added to power traces

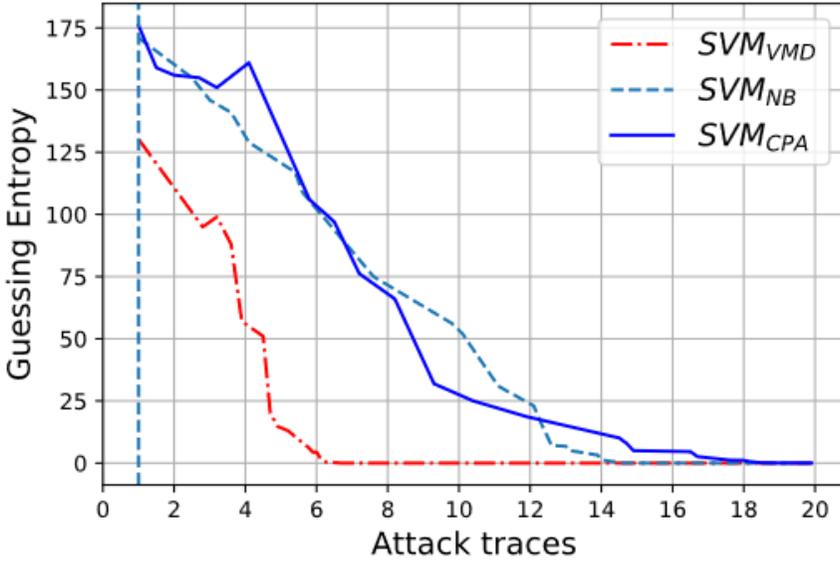


Figure 13. Attack results on Dataset 2 with $SNR_1 = 20$ noise added to power traces

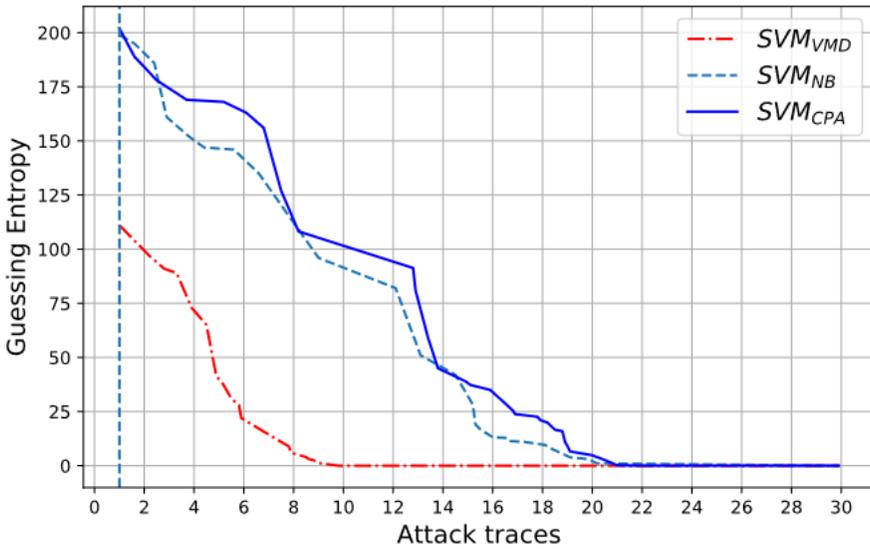


Figure 14. Attack results on Dataset 2 with $SNR_2 = 10$ noise added to power traces

REFERENCES

- [1] KOCHER, P.—JAFFE, J.—JUN, B.: Differential Power Analysis. In: Wiener, M. (Ed.): *Advances in Cryptology – CRYPTO 1999*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1666, 1999, pp. 388–397, doi: 10.1007/3-540-48405-1.25.
- [2] BRIER, E.—CLAVIER, C.—OLIVIER, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J. J. (Eds.): *Cryptographic Hardware and Embedded Systems – CHES 2004*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3156, 2004, pp. 16–29, doi: 10.1007/978-3-540-28632-5_2.
- [3] GIERLICH, B.—BATINA, L.—TUYLS, P.—PRENEEL, B.: Mutual Information Analysis. In: Oswald, E., Rohatgi, P. (Eds.): *Cryptographic Hardware and Embedded Systems – CHES 2008*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5154, 2008, pp. 426–442, doi: 10.1007/978-3-540-85053-3.27.
- [4] CHARI, S.—RAO, J. R.—ROHATGI, P.: Template Attacks. In: Kaliski, B. S., Koç, K., Paar, C. (Eds.): *Cryptographic Hardware and Embedded Systems – CHES 2002*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2523, 2002, pp. 13–28, doi: 10.1007/3-540-36400-5.3.
- [5] HEUSER, A.—ZOHNER, M.: Intelligent Machine Homicide. In: Schindler, W., Huss, S. A. (Eds.): *Constructive Side-Channel Analysis and Secure Design (COSADE 2012)*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7275, 2012, pp. 249–264, doi: 10.1007/978-3-642-29912-4.18.
- [6] HOSPODAR, G.—GIERLICH, B.—DE MULDER, E.—VERBAUWHEDE, I.—VANDEWALLE, J.: Machine Learning in Side-Channel Analysis: A First Study. *Journal of Cryptographic Engineering*, Vol. 1, 2011, Art.No. 293, doi: 10.1007/s13389-011-0023-x.
- [7] HOSPODAR, G.—DE MULDER, E.—GIERLICH, B.—VANDEWALLE, J.—VERBAUWHEDE, I.: Least Squares Support Vector Machines for Side-Channel Analysis. *Second International Workshop on Constructive Side-Channel (COSADE 2011)*, Darmstadt, 2011.
- [8] PICEK, S.—HEUSER, A.—JOVIC, A.—LUDWIG, S. A.—GUILLEY, S.—JAKOBOVIC, D.—MENTENS, N.: Side-Channel Analysis and Machine Learning: A Practical Perspective. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 4095–4102, doi: 10.1109/IJCNN.2017.7966373.
- [9] LERMAN, L.—STEPHANE, F. M.—BONTEMPI, G.—MARKOWITZ, O.: A Machine Learning Approach Against a Masked AES. In: Francillon, A., Rohatgi, P. (Eds.): *Smart Card Research and Advanced Applications (CARDIS 2013)*. Springer, Cham, Lecture Notes in Computer Science, Vol. 8419, 2013, pp. 61–75, doi: 10.1007/978-3-319-08302-5.5.
- [10] ZHENG, Y.—ZHOU, Y.—YU, Z.—HU, C.—ZHANG, H.: How to Compare Selections of Points of Interest for Side-Channel Distinguishers in Practice? In: Hui, L. C. K., Qing, S. H., Shi, E., Yiu, S. M. (Eds.): *Information and Communications Security (ICICS 2014)*. Springer, Cham, Lecture Notes in Computer Science, Vol. 8958, 2015, pp. 200–214, doi: 10.1007/978-3-319-21966-0.15.

- [11] RECHBERGER, C.—OSWALD, E.: Practical Template Attacks. In: Lim, C.H., Yung, M. (Eds.): Information Security Applications (WISA 2004). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3325, 2005, pp. 440–456, doi: 10.1007/978-3-540-31815-6_35.
- [12] GIERLICH, B.—LEMKE-RUST, K.—PAAR, C.: Templates vs. Stochastic Methods. In: Goubin, L., Matsui, M. (Eds.): Cryptographic Hardware and Embedded Systems – CHES 2006. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4249, 2006, pp. 15–29, doi: 10.1007/11894063_2.
- [13] MANGARD, S.—OSWALD, E.—POPP, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer US, 2007, doi: 10.1007/978-0-387-38162-6.
- [14] LOMNÉ, V.—PROUFF, E.—ROCHE, T.: Behind the Scene of Side Channel Attacks. In: Sako, K., Sarkar, P. (Eds.): Advances in Cryptology – ASIACRYPT 2013. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8269, 2013, pp. 506–525, doi: 10.1007/978-3-642-42033-7_26.
- [15] LERMAN, L.—BONTEMPI, G.—MARKOWITZ, O.: Side Channel Attack: An Approach Based on Machine Learning. Second International Workshop on Constructive Side-Channel (COSADE 2011), Darmstadt, 2011.
- [16] LIU, J.—ZHOU, Y.—HAN, Y.—LI, J.—YANG, S.—FENG, D.: How to Characterize Side-Channel Leakages More Accurately? In: Bao, F., Weng, J. (Eds.): Information Security Practice and Experience (ISPEC 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6672, 2011, pp. 196–207, doi: 10.1007/978-3-642-21031-0_15.
- [17] MAGHREBI, H.—PORTIGLIATTI, T.—PROUFF, E.: Breaking Cryptographic Implementations Using Deep Learning Techniques. In: Carlet, C., Hasan, M., Saraswat, V. (Eds.): Security, Privacy, and Applied Cryptography Engineering (SPACE 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 10076, 2016, pp. 3–26, doi: 10.1007/978-3-319-49445-6_1.
- [18] HEUSER, A.—PICEK, S.—JOVIC, A.—LEGAY, A.: On the Relevance of Feature Selection for Profiled Side-Channel Attacks. 2017, <https://hal.archives-ouvertes.fr/hal-01645128>.
- [19] BARTKEWITZ, T.—LEMKE-RUST, K.: Efficient Template Attacks Based on Probabilistic Multi-Class Support Vector Machines. In: Mangard, S. (Ed.): Smart Card Research and Advanced Applications (CARDIS 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7771, 2012, pp. 263–276, doi: 10.1007/978-3-642-37288-9_18.
- [20] DRAGOMIRETSKIY, K.—ZOSSO, D.: Variational Mode Decomposition. IEEE Transactions on Signal Processing, Vol. 62, 2014, No. 3, pp. 513–544, doi: 10.1109/TSP.2013.2288675.
- [21] BOYD, S.—PARIKH, N.—CHU, E.—PELEATO, B.—ECKSTEIN, J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends® in Machine Learning, Vol. 3, 2010, No. 1, pp. 1–122, doi: 10.1561/2200000016.
- [22] CORTES, C.—VAPNIK, V.: Support-Vector Networks. Machine Learning, Vol. 20, 1995, No. 3, pp. 273–297, doi: 10.1007/bf00994018.

- [23] PLATT, J. C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers*, MIT Press, 1999, pp. 61–74.
- [24] HASTIE, T.—TIBSHIRANI, R.: Classification by Pairwise Coupling. *The Annals of Statistics*, Vol. 26, 1998, No. 2, pp. 451–471, doi: 10.1214/aos/1028144844.
- [25] WESTON, J.—WATKINS, C.: Multi-Class Support Vector Machines. Technical Report CSD-TR-98-04, Royal Holloway, University of London, 1998.
- [26] STOPPIGLIA, H.—DREYFUS, G.—DUBOIS, R.—OUSSAR, Y.: Ranking a Random Feature for Variable and Feature Selection. *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1399–1414.
- [27] ESWARAMOORTHY, S.—SIVAKUMARAN, N.—SEKARAN, S.: Grey Wolf Optimization Based Parameter Selection for Support Vector Machines. *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, Vol. 35, 2016, No. 5, pp. 1513–1523, doi: 10.1108/COMPEL-09-2015-0337.
- [28] STANDAERT, F. X.—MALKIN, T. G.—YUNG, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: Joux, A. (Ed.): *Advances in Cryptology – EUROCRYPT 2009*. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5479, 2009, pp. 443–461, doi: 10.1007/978-3-642-01001-9_26.
- [29] NASSAR, M.—SOUISSI, Y.—GUILLEY, S.—DANGER, J.-L.: RSM: A Small and Fast Countermeasure for AES, Secure Against 1st and 2nd-Order Zero-Offset SCAs. *2012 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Dresden, 2012, pp. 1173–1178, doi: 10.1109/DATE.2012.6176671.



Ngoc Quy TRAN received his B.Sc. and M.Sc. degrees in electronics and communication from Vietnam National University, Hanoi (VNU), in 2003 and 2006, respectively. His research interests include hardware security, machine learning and side channel attack. He is now Ph.D. candidate at the Academy of Cryptography Techniques, Hanoi, Vietnam.



Junbeom HUR received his B.Sc. degree in computer science from Korea University, Seoul, South Korea, in 2001, and the M.Sc. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2005 and 2009, respectively. He was with the University of Illinois at Urbana–Champaign, Champaign, IL, USA, as Post-Doctoral Researcher from 2009 to 2011. He is currently Associate Professor with the Department of Computer Science and Engineering, Korea University. His current research interests include information security, cloud computing security, mobile security, and applied cryptography.



Hieu Minh NGUYEN is Vice Dean at the Institute of Cryptographic Science and Technology, Hanoi, Vietnam. He finished his Ph.D. at the Saint Petersburg Electrical Engineering University (2006). His current research interests include cryptography, database security, information security, and network security. He has authored or co-authored more than 85 scientific articles, book chapters, reports, and patents, in the areas of his research.