Template Attacks Based On Priori Knowledge

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

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

Abstract. Template Attacks consist of two stages, the profiling stage and the extraction stage. In order to improve the key-recovery efficiency of Template Attacks, a feasible way is to characterize signals and noises accurately. Under the assumption that a reference device is available, in the profiling stage, one can operate the reference device as many times as possible and samples a large number of power traces to help accurately characterize signals and noises at different interesting points. However, in some practical scenarios, it is not always the case and one can only record a *limited* number of power traces. In this paper, we show that one can still make Template Attacks practical and more powerful in the above scenario if he could obtain some kind of priori knowledge about the reference device. For example, the priori knowledge is some kind of priori distribution of the signal component in the instantaneous power consumption for fixed operation on fixed data. Evaluation results show that the priori knowledge poses potential threat to the physical security of cryptographic devices and this kind of threat can not be neglected.

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

1 Introduction

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

Template Attacks consist of two stages. The first stage is the profiling stage and the second stage is the extraction stage. In the profiling stage, one has a reference device identical or similar to the targeted device and builds templates for each key-dependent operation with the reference device. In the extraction stage, one can exploit a small number of power traces measured from the targeted device and the templates to classify the correct (sub)key.

Now, let's focus on the practical attack scenario. In order to improve the key-recovery efficiency of Template Attacks, a feasible and usual way is to characterize signals and noises accurately. Under the assumption that a reference device is fully controlled by the attacker, in the profiling stage, the attacker can operate the reference device as many times as possible and samples a large number of power traces to help accurately characterize signals and noises at different interesting points. However, in some practical scenarios, it is not always the case and the attacker can only record a *limited* number of power traces. For example, a common countermeasure is used to limit the number of invocations that the reference device can perform in certain time interval, or that the reference device performs under one key for limited number of invocations and then the key is refreshed. In these cases, the attacker can only record limited number of power traces. Furthermore, the signals and noises may not be characterized accurately enough if the attacker uses traditional method of building templates with limited number of power traces.

Motivations Although the attacker can not obtain enough power traces to characterize signals and noises accurately enough in the above cases, it is still possible for him to possess some kind of priori knowledge about the reference device (as well as the targeted device) in practical. Specifically speaking, the priori knowledge may be some kind of priori distribution of the signal component in the instantaneous power consumption for fixed operation on fixed data. We show three examples about this case. *Example 1:* When the attacker has a device similar to the reference device and the device is under his full control, he can obtain priori knowledge about the reference device from this device. *Example 2:* The attacker may obtain priori knowledge about the reference device from his previous experiments of conducting Template Attacks. *Example 3:* For a sophisticated attacker, after obtaining power traces from the reference device, he may use the power traces to obtain the interval estimation of the signal component (may be not very accurate) and guesses the prior distribution of the signal component is some kind of distribution over the interval by his instincts. To sum up, for a seasoned attacker, it is very difficult to guarantee that he does not possess any priori knowledge about the reference device from a practical point of view.

Therefore, we need to answer two natural and important questions when the attacker can not obtain enough power traces but has priori knowledge about the reference device. The first question is that how can the attacker exploit the priori knowledge in a correct and reasonable way to improve the classification performance of Template Attacks? The second question is that whether or not the priori knowledge (even if may not be very accurate) will make Template Attacks practical and more powerful (achieve higher classification performance)? In this paper, we try to answer these two important questions.

Contributions Main contributions of this paper are two-folds. Firstly, based on the method of Bayes estimation [17], we theoretically give out a correct and reasonable way of exploiting priori knowledge when the attacker conducts Template Attacks with limited number of power traces in the profiling stage. Secondly, we verify the way of exploiting priori knowledge by both simulated and practical experiments. Evaluation results show that Template Attacks will be practical and more powerful if the attacker can possess the priori knowledge. What's more, more accurate the priori knowledge is, more powerful Template Attacks will be. This discovery enables us to realize that the attacker may be more powerful than we previously think if he obtains some kind of priori knowledge about the reference device.

Related Work The paper [2] provided answers to some basic and practical issues of Template Attacks, such as how to choose interesting points in an efficient way and how to preprocess noisy data. The paper [4] proposed efficient methods to avoid several possible numerical obstacles when implementing Template Attacks. The paper [12] presented a variant of Template Attacks that can be applied to block ciphers when the plaintext and ciphertext used are unknown. In [8], Template Attacks were used to attack a masking protected implementation of a block cipher. Recently, a simple pre-processing technique of Template Attacks, normalizing the sample values using the means and variances was evaluated for various sizes of test data [7]. Principal Component Analysis (PCA)-Based Template Attacks were investigated in [3]. However, this kind of Template Attacks may not improve the classification performance [7]. Therefore, PCA-Based Template Attacks are not used widely in practice and are ignored in this paper. LDA-based Template Attacks were introduced in [9]. This kind of Template Attacks depends on the condition of equal covariances [4] (Please see Section 2.1.1 for more details.), which does not hold in most settings. Therefore, it is not a better choice compared with PCA-based Template Attacks in most settings [4].

Organization of This Paper The rest of this paper is organized as follows. In Section 2, we review the basic concept of Template Attacks and the method of Bayes estimation. In Section 3, we give out a correct and reasonable way of exploiting priori knowledge to make Template Attacks practical and more powerful. In Section 4, we verify the way of exploiting the priori knowledge by simulated and practical experiments. In Section 5, we conclude the whole paper.

2 Preliminaries

Template Attacks capture two subcategories: Classical Template Attacks [1] and Reduced Template Attacks [23]. In this section, we briefly review Classical Template Attacks, Reduced Template Attacks, and the method of Bayes estimation.

2.1 Classical Template Attacks

We will introduce the two stages of Classical Template Attacks: the profiling stage and the extraction stage.

2.1.1 The Profiling Stage Assume that there exist K different (sub)keys $key_i, i = 0, 1, \ldots, K-1$ which need to be classified. Also, there exist K different key-dependent operations O_i, $i = 0, 1, \ldots, K-1$. Usually, one will generate K templates, one for each key-dependent operation O_i. One can exploit some methods to choose N interesting points $(P_0, P_1, \ldots, P_{N-1})$. The interesting points are those time samples that contain the most information about the characterized key-dependent operations. Each template is composed of a mean vector and a covariance matrix. The mean vector is used to estimate the signal component of side-channel leakages. It is the average signal vector $\mathbf{M}_i = (M_i[P_0], \ldots, M_i[P_{N-1}])$ for each one of the key-dependent operations. The covariance matrix is used to estimate the probability density of the noise component at different interesting points. It is assumed that noises at different interesting points approximately follow the multivariate normal distribution. A N dimensional noise vector $\mathbf{n}_i(\mathbf{S})$ is extracted from each power trace $\mathbf{S} = (S[P_0], \ldots, S[P_{N-1}])$ representing the template's key dependency O_i as $\mathbf{n}_i(\mathbf{S}) = (S[P_0] - M_i[P_0], \ldots, S[P_{N-1}] - M_i[P_{N-1}])$. One computes the $(N \times N)$ covariance matrix $\mathbf{C}_i$ from these noise vectors. The probability density of the noises occurring under key-dependent operation O_i is given by the N dimensional multivariate Gaussian distribution $p_i(\cdot)$, where the probability of observing a noise vector $\mathbf{n}_i(\mathbf{S})$ is:

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

In equation (1), the symbol $|\mathbf{C}_i|$ denotes the determinant of $\mathbf{C}_i$ and the symbol $\mathbf{C}_i^{-1}$ denotes its inverse. We know that the matrix $\mathbf{C}_i$ is the estimation of the true covariance $\mathbf{\Sigma}_i$. The condition of equal covariances [4] means that the leakages from different key-dependent operations have the same true covariance $\mathbf{\Sigma} = \mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \cdots = \mathbf{\Sigma}_{K-1}$. In most settings, the condition of equal covariances does not hold. Therefore, in this paper, we only consider the devices in which the condition of equal covariances does not hold.

2.1.2 The Extraction Stage Assume that one obtains t power traces (denoted by $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_t$) from the targeted device in the extraction stage. For example, when the power traces are statistically independent, one will apply maximum likelihood approach on the product of conditional probabilities [11], i.e.

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

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

2.2 Reduced Template Attacks

In order to avoid numerical problems with the inversion of the covariance matrix $\mathbf{C}_i$, one can set the covariance matrix equal to the identity matrix. This essentially means that one does not take the covariances between the interesting points into account. A template that only consists of a mean vector is called a *reduced template* [23]. Naturally, Template Attacks based on reduced templates are called Reduced Template Attacks. In Reduced Template Attacks, the probability density of the noises occurring under key-dependent operation O_i is given by the N dimensional multivariate Gaussian distribution $p'_i(\cdot)$, where the probability of observing a noise vector $\mathbf{n}_i(\mathbf{S})$ is:

$$p'_i(\mathbf{n}_i(\mathbf{S})) = \frac{1}{\sqrt{(2\pi)^N}} exp\Big(- \frac{1}{2}\mathbf{n}_i(\mathbf{S})\mathbf{n}_i(\mathbf{S})^T \Big) \quad \mathbf{n}_i(\mathbf{S}) \in \mathbb{R}^N.$$

The space of this paper does not allow to show Reduced Template Attacks in details (Please see [23] for more details about Reduced Template Attacks.).

2.3 The Method of Bayes Estimation

In the following, we briefly introduce the method of Bayes estimation [17]. We first introduce the definition of Bayes estimators. Then, we introduce how to compute a Bayes estimator.

Definition 1. *An estimator is a real-valued function δ defined over the sample space. It is used to estimate an estimand, $g(\theta)$, a real-valued function of the parameter θ [17].*

Suppose an unknown parameter θ is known to have a prior distribution Λ. Quite generally, suppose that the consequences of estimating $g(\theta)$ by a value $\delta(X)$ (based on some measurements X) are measured by $L(\theta, \delta(X))$. Of the *loss function* L, we shall assume that

$$L(\theta, \delta(X)) \geq 0 \text{ for all } \theta \text{ and } \delta(X),$$

and

$$L[\theta, g(\theta)] = 0 \text{ for all } \theta,$$

so that the loss is zero when the correct value is estimated. The accuracy, or rather inaccuracy, of an estimator δ is then measured by the *risk function*

$$R(\theta, \delta) = E_\theta\{L[\theta, \delta(X)]\},$$

the long-term average loss resulting from the use of $\delta(X)$. This defines the risk function as a function of $\delta(X)$. An estimator $\delta(X)$ minimizing

$$r(\Lambda, \delta) = \int R(\theta, \delta) d\Lambda(\theta)$$

is called a *Bayes estimator* with respect to the prior distribution Λ. Note that, the distribution Λ is a probability distribution of the parameter θ, that is,

$$\int d\Lambda(\theta) = 1.$$

Now, we will introduce how to compute a Bayes estimator of an unknown parameter θ. Let $\lambda(\theta)$ denote the prior probability density of the parameter θ. The prior probability density of the population (or discrete probability function) is denoted by $f(X, \theta)$. If one extracts n samples $(X_1, X_2, \ldots, X_n)$ from the population, then the probability density of this group of samples is

$$f(X_1, \theta)f(X_2, \theta) \cdots f(X_n, \theta).$$

Thereby, we can compute the marginal density

$$p(X_1, X_2, \ldots, X_n) = \int \lambda(\theta)f(X_1, \theta)f(X_2, \theta) \cdots f(X_n, \theta)d\theta.$$

Then, the following posterior probability density is computed:

$$\lambda(\theta|X_1, \ldots, X_n) = \lambda(\theta)f(X_1, \theta) \cdots f(X_n, \theta)/p(X_1, X_2, \ldots, X_n). \tag{2}$$

In general, the Bayes estimator of the parameter θ is set to be the mean value of $\lambda(\theta|X_1, \ldots, X_n)$ [17].

3 The Usage of The Prior Knowledge For Template Attacks

In this section, we introduce the usage of prior knowledge for Template Attacks. The usage is the same for both Classical Template Attacks and Reduced Template Attacks.

It is well known that the power consumption of each point of a power trace (PC_{total}) can be modeled as the sum of an operation-dependent component PC_{op}, a data-dependent component PC_{data}, electronic noise $PC_{el.noise}$, and a constant component PC_{const} [18], i.e.

$$PC_{total} = PC_{op} + PC_{data} + PC_{el.noise} + PC_{const}.$$

The value $PC_{op}+PC_{data}$ (or $PC_{op}+PC_{data}+PC_{const}$) can be viewed as the signal component and the value $PC_{el.noise}$ can be viewed as the noise component. Usually, for each point P_j of a power trace, its power consumption PC_{total} follows a normal distribution $\mathcal{N}(\mu_j, \sigma_j^2)$ and the distribution of the electronic noise $PC_{el.noise}$ is a normal distribution: $PC_{el.noise} \sim \mathcal{N}(0, \sigma_j^2)$ [18]. For fixed operation on fixed data, due to

$$Var(PC_{op}) = Var(PC_{data}) = Var(PC_{const}) = 0,$$

it has that $PC_{op} + PC_{data} + PC_{el.noise} = \mu_j$.

More accurate the signal component (the real value of μ_j) is estimated, more accurate the noise component (the value $PC_{total} - \mu_j$) will be estimated. If the signal component and the noise component are accurately estimated, accurate templates will be built and higher classification performance of Template Attacks will be achieved. In the classical way of conducting Template Attacks, for an interesting point, the attacker computes the mean value of the power consumption to estimate the real value of μ_j. Specifically speaking, for the key-dependent operation O_i, the point P_{poi} is an interesting point and the attacker obtains n power traces $(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)$ to build the template for the key-dependent operation O_i (Note that, the number of power traces is limited (i.e. the value n is small).). Therefore, the attacker obtains n values of the power consumption of the point P_{poi}, one from each power trace. The n values are $S_1[P_{poi}], S_2[P_{poi}], \ldots, S_n[P_{poi}]$. The real value of μ_j is estimated by $\tilde{\mu}_j$ as follows:

$$\tilde{\mu}_j = M_i[P_{poi}] = \sum_{k=1}^{n} S_k[P_{poi}]/n.$$

However, in our scenario, the attacker not only has n power traces (obtained from the reference device) which can be used for building the template for the key-dependent operation O_i, but also some kind of prior knowledge about the reference device (the targeted device). The prior knowledge is some kind of prior distribution of the real value of μ_j. Let's consider the most common case. Assume that the attacker knows that the real value of μ_j follows the normal distribution $\mathcal{N}(\theta_1, \theta_2^2)$[1] (Note that, from the prior knowledge, the parameters θ_1, θ_2^2 are known to the attacker.) and does not know what the real value of μ_j accurately is. The attacker can use the method of Bayes estimation to estimate the real value of μ_j with the prior knowledge as follows: The attacker computes the probability density of the real value of μ_j from prior knowledge as

$$\lambda(\mu_j) = (\sqrt{2\pi}\theta_2)^{-1} exp\Big[-\frac{1}{2\theta_2^2}(\mu_j - \theta_1)^2\Big].$$

Moreover, the power consumption of the point P_{poi} satisfies the following probability density function:

$$f(x; \mu_j, \sigma_j) = (\sqrt{2\pi}\sigma_j)^{-1} exp\Big[-\frac{1}{2\sigma_j^2}(x - \mu_j)^2\Big].$$

Although the value σ_j is not known to the attacker, it does not affect the process of computing the Bayes estimator of the real value of μ_j. What the attacker needs is an accurate estimation of the real value of μ_j. From equation (2), the attacker

[1] The attacker may possess other kind of prior distribution of μ_j, e.g., an uniform distribution over a small closed interval which contains μ_j. This case may be occur in practice because the attacker can exploit interval estimation to obtain the small closed interval.

computes the posterior probability density:

$$\lambda(\mu_j|S_1[P_{poi}],\ldots,S_n[P_{poi}])=exp\Big[-\frac{1}{2\theta_2^2}(\mu_j-\theta_1)^2-\frac{1}{2}\sum_{k=1}^{n}(S_k[P_{poi}]-\mu_j)^2\Big]/C_1,$$

the constant C_1 only has relation with $\theta_1,\theta_2,S_1[P_{poi}],\ldots,S_n[P_{poi}]$ and has no relation with μ_j. It has that

$$-\frac{1}{2\theta_2^2}(\mu_j-\theta_1)^2-\frac{1}{2}\sum_{k=1}^{n}(S_k[P_{poi}]-\mu_j)^2=-\frac{1}{2A^2}(\mu_j-B)^2+C_2,$$

where

$$A^2=1/(n+1/\theta_2^2),$$
$$B=(nM_i[P_{poi}]+\theta_1/\theta_2^2)/(n+1/\theta_2^2),$$

and C_2 has no relation with μ_j. Furthermore, the attacker can obtain

$$\lambda(\mu_j|S_1[P_{poi}],\ldots,S_n[P_{poi}])=C_3exp\Big[-\frac{1}{2A^2}(\mu_j-B)^2\Big],$$

here $C_3=C_1e^{C_2}$. Because it has that

$$\int_{-\infty}^{+\infty}\lambda(\mu_j|S_1[P_{poi}],\ldots,S_n[P_{poi}])d\mu_j=1,$$

hence $C_3=(\sqrt{2\pi}A)^{-1}$. Up to now, the attacker obtains the Bayes estimator of the real value of μ_j as

$$\tilde{\mu}_j=\frac{n}{n+1/\theta_2^2}\left(\frac{\sum_{k=1}^{n}S_k[P_{poi}]}{n}\right)+\frac{1/\theta_2^2}{n+1/\theta_2^2}\theta_1. \tag{3}$$

The equation (3) shows that if the attacker does not have the prior knowledge (i.e. the prior distribution of μ_j: $\mathcal{N}(\theta_1,\theta_2^2)$), he can only use $\sum_{k=1}^{n}S_k[P_{poi}]/n$ to estimate μ_j. If the attacker does not have power traces obtained from the reference device, he can only use the prior knowledge (i.e. the value θ_1) to estimate μ_j. If the attacker has the prior knowledge as well as power traces obtained from the reference device, by equation (3), he will use the weighted average of $\sum_{k=1}^{n}S_k[P_{poi}]/n$ and θ_1 to estimate the real value of μ_j under the ratio $n:1/\theta_2^2$. This ratio is reasonable and the relevant reasons are as follows. On one hand, when more power traces are obtained from the reference device, the proportion of $\sum_{k=1}^{n}S_k[P_{poi}]/n$ will be larger. On the other hand, when the value θ_2^2 is smaller (This means that the prior distribution of μ_j is more accurate and one (the attacker) can infer μ_j with strong confidence from the prior distribution.), the proportion of θ_1 will be larger.

In the next section, we will experimentally verify the classification performance of Template Attacks with prior knowledge.

4 Experimental Evaluations

For the implementation of a cryptographic algorithm with countermeasures, one usually first uses some methods to delete the countermeasures from power traces and then tries to recover the (sub)key using classical attack methods against unprotected implementation. For example, if one has power traces with random delays [13]. He may first use the method proposed in [14] to remove the random delays from power traces and then uses classical attack methods to recover the correct (sub)key. The methods of deleting countermeasures from power traces are beyond the scope of this paper. Therefore, we take unprotected AES-128 implementation as example. We verified both Classical Template Attacks and Reduced Template Attacks by conducting both simulated and practical experiments. In both simulated and practical experiments, we tried to attack the output of the S-boxes in the first round of AES-128[1]. Before introducing specific experiments, we first introduce how to generate the prior distribution of the signal component for every interesting point for both simulated and practical experiments.

For simplicity, let np denote the number of power traces used in the profiling stage and let ne denote the number of power traces used in the extraction stage. In this paper, we use *Guessing Entropy* [6] as a metric about the classification performance of Template Attacks (Many other papers also used Guessing Entropy as a metric. e.g. [19, 21, 22]).

Note that, in all the experiments, the different kinds of Template Attacks introduced in Section 4.1 (CTA, CTA-16, CTA-32, CTA-64, CTA-128, RTA, RTA-16, RTA-32, RTA-64, and RTA-128) used the same sets of power traces both in the profiling stage and the extraction stage.

4.1 Generating The Prior Distribution

In both simulated and practical experiments, for each key-dependent operation O_i, we considered the prior distribution of the signal component μ_j with four different levels of accuracy for every interesting point P_{poi} and assumed the prior distribution is a normal distribution $\mathcal{N}(\theta_1, \theta_2^2)$. For each key-dependent operation O_i, we generated 400 traces (simulated traces or real power traces). The 400 traces were used to estimate the parameters θ_1, θ_2^2 for every interesting point as follows. We repeated a process 300 times. Every time, we chose 16 traces (Let $m = 16$ and the 16 traces are denoted by $S_1, \ldots, S_m$.) from the 400 traces uniformly at random and computed $\sum_{k=1}^{m} S_k[P_{poi}]/m$. Therefore, there were 300 different values of $\sum_{k=1}^{m} S_k[P_{poi}]/m$. The mean value of the 300 different values was set to be θ_1 and the variance of the 300 different values was set to be θ_2^2. In this way, we obtained the estimation of θ_1 and θ_2^2. Similarly, we additionally let m = 32, 64, 128 and obtained four different groups of estimation of θ_1 and θ_2^2. Clearly, when the value m is larger, the estimation of θ_1 and θ_2^2 is more

[1] Due to the length of the output of every S-box is 8 bits long, we need to build 256 templates, one for each output.

accurate. Therefore, we obtained estimation of the parameters θ_1 and θ_2^2 with four different levels of accuracy. The corresponding four normal distributions represent the prior knowledge which the real attacker can possess in practical attack scenario.

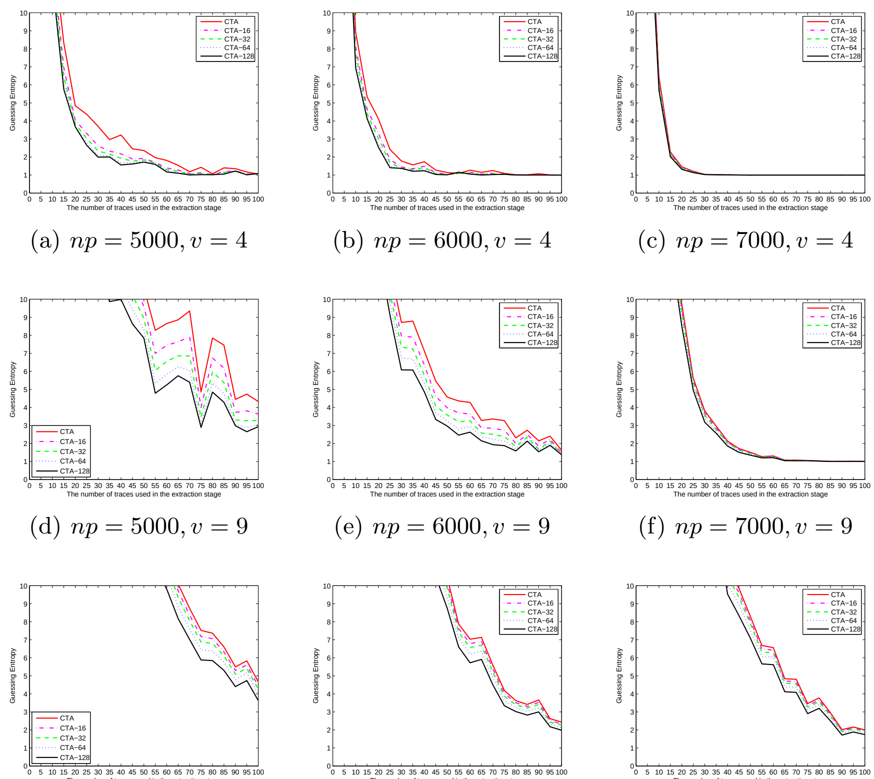
In all the experiments, we let the symbol "CTA" denotes the Classical Template Attacks without any prior knowledge. The symbol "CTA-16" denotes Classical Template Attacks based on prior knowledge (i.e. The value μ_j is estimated by equation (3)) which is obtained when the value m equals to 16. Similarly, we define the symbols "CTA-32", "CTA-64", and "CTA-128" to denote the cases that the value m equals to 32, 64, and 128 respectively. We let the symbol "RTA" denotes the Reduced Template Attacks without any prior knowledge. The symbol "RTA-16" denotes Reduced Template Attacks based on prior knowledge (i.e. The value μ_j is estimated by equation (3)) which is obtained when the value m equals to 16. Similarly, we define the symbols "RTA-32", "RTA-64", and "RTA-128" to denote the cases that the value m equals to 32, 64, and 128 respectively.

4.2 Simulated Experiments

In all simulated experiments, we chose 4 interesting points and the typical Hamming-Weight power model [20] was adopted to describe the power consumption. In all simulated experiments, the variance of simulated Gaussian noise is denoted by v. We employed three different noise levels to test the influence of noises on the classification performance of the attacks. The variances of simulated Gaussian noise for the three noise levels were 4, 9, and 16.

We used 5,000, 6,000, and 7,000 simulated traces to build the 256 templates respectively. The simulated traces were generated with a fixed subkey and random plaintext inputs. We generated additional 100,000 simulated traces with another fixed subkey and random plaintext inputs. The 100,000 simulated traces were used in the extraction stage. For fixed np and v, we tested the Guessing Entropy of the five kinds of Classical Template Attacks (CTA, CTA-16, CTA-32, CTA-64, and CTA-128) when one uses ne traces in the extraction stage as follows. We respectively repeated the five kinds of Classical Template Attacks and Reduced Template Attacks 1,000 times. For each time, we chose ne traces from the 100,000 traces uniformly at random and the five kinds of Classical Template Attacks were conducted with the same ne traces. We respectively computed the Guessing Entropy of the five kinds of Classical Template Attacks with the results of the 1,000 times of attacks. The Guessing Entropy of the five kinds of Classical Template Attacks is shown in Figure 1. From Figure 1, we find that if the prior knowledge is more accurate, the classification performance of Classical Template Attacks with prior knowledge will be higher. For moderate level noises (e.g. $v = 9$), Classical Template Attacks with prior knowledge achieve the biggest advantage over Classical Template Attacks without prior knowledge. When more simulated traces can be obtained from the reference device (e.g. np = 7,000), the advantages of Classical Template Attacks with prior knowledge will be smaller.

For Reduced Template Attacks, we computed the Guessing Entropy of the five kinds of Reduced Template Attacks (RTA, RTA-16, RTA-32, RTA-64, and RTA-128) similarly. The results of simulated experiments show that Reduced Template Attacks with prior knowledge have advantages over Reduced Template Attacks without prior knowledge, especially when the noise level is high. The space does not allow to show the results of simulated experiments of Reduced Template Attacks here.



(a) $np = 5000, v = 4$ (b) $np = 6000, v = 4$ (c) $np = 7000, v = 4$

(d) $np = 5000, v = 9$ (e) $np = 6000, v = 9$ (f) $np = 7000, v = 9$

(g) $np = 5000, v = 16$ (h) $np = 6000, v = 16$ (i) $np = 7000, v = 16$

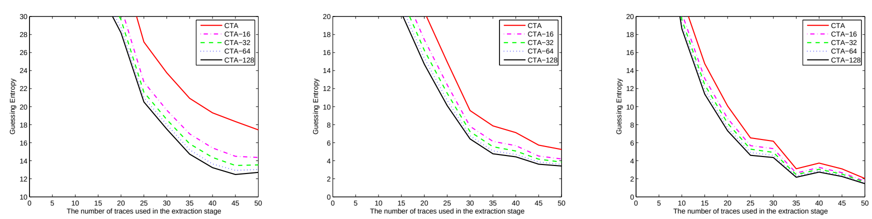
Fig. 1. Simulated Experiments Results

4.3 Practical Experiments

We tried to attack the output of the first and the second S-box in the first round of an unprotected AES-128 software implementation on an typical 8-bit microcontroller STC89C58RD+ (This 8-bit microcontroller was also exploited by other papers [15, 16].) whose operating frequency is 11MHz. The real power traces were acquired with a sampling rate of 50MS/s. The average number of real power traces during the sampling process was 10 times.

We used the same device for both the profiling stage and the extraction stage, which provides a good setting for the focuses of our research. For our device, the condition of equal covariances does not hold. This means that the differences between different covariance matrixes $\mathbf{C}_i$ are very evident (can easily be observed from visual inspection).

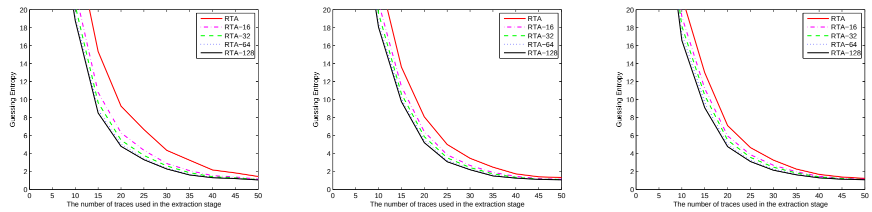
We used 5,000, 7,500, and 10,000 real power traces to build the 256 templates respectively. The real power traces were generated with a fixed main key and random plaintext inputs. We generated additional 100,000 real power traces with another fixed main key and random plaintext inputs. The 100,000 real power traces were used in the extraction stage. For fixed np, we tested the Guessing Entropy of the five kinds of Classical Template Attacks (CTA, CTA-16, CTA-32, CTA-64, and CTA-128) when one uses ne real power traces in the extraction stage as follows. We repeated the five kinds of Classical Template Attacks and Reduced Template Attacks 1,000 times. For each time, we chose ne real power traces from the 100,000 traces uniformly at random. The five kinds of Classical Template Attacks were conducted with the same ne real power traces. We respectively computed the Guessing Entropy of the five kinds of Classical Template Attacks with the results of the 1,000 times of attacks.



(a) $np = 5,000$ (b) $np = 7,500$ (c) $np = 10,000$

Fig. 2. The experiment results of Classical Template Attacks for the first S-box

The Guessing Entropy of the five kinds of Classical Template Attacks for the first S-box are shown in Figure 2. For Reduced Template Attacks, we computed the Guessing Entropy of the five kinds of Reduced Template Attacks (RTA, RTA-16, RTA-32, RTA-64, and RTA-128) similarly. The Guessing Entropy of the five kinds of Reduced Template Attacks for the first S-box are shown in Figure 3. From Figure 2 and Figure 3, we find that, for both Classical Template Attacks and Reduced Template Attacks, if the prior knowledge is more accurate, the classification performance will be higher. When more real power traces can be obtained from the reference device (e.g. $np = 10,000$), the advantages of Template Attacks with prior knowledge will be smaller. The results of the experiments about the second S-box for both Classical Template Attacks and Reduced Template Attacks are in the Appendix A. For other S-boxes in the first



(a) $np = 5,000$ (b) $np = 7,500$ (c) $np = 10,000$

Fig. 3. The experiment results of Reduced Template Attacks for the first S-box

round of the unprotected AES-128 software implementation, similar evaluation results were obtained by us.

5 Conclusion and Future Work

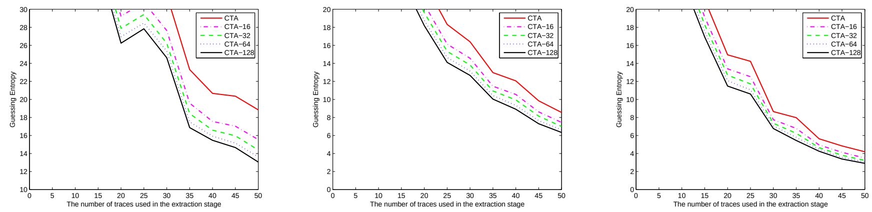
In this paper, we verify that if the attacker obtains the prior knowledge about the reference device (the targeted device), Template Attacks (both Classical Template Attacks and Reduced Template Attacks) will be practical and more powerful than we previously think. The priori knowledge really poses potential threat to the physical security of cryptographic devices. Therefore, we suggest that the designers of a cryptographic device should take the prior knowledge into consideration when he uses Template Attacks to evaluate the physical security of the cryptographic device. In the future, it is necessary to research how to exploit the prior knowledge to make other profiled side-channel attacks (such as Stochastic Model based Attacks [24]) become more powerful in a reasonable way. It is also necessary to further verify our discoveries in other devices such as ASIC and FPGA.

References

1. Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. CHES2002, LNCS 2523, pp.13-28, 2003.
2. Rechberger, C., Oswald, E.: Practical Template Attacks. WISA2004, LNCS 3325, pp.440-456, 2004.
3. Archambeau, C., Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Template Attacks in Principal Subspaces. CHES2006, LNCS 4249, pp.1-14, 2006.
4. Choudary, O., Kuhn, M.G.: Efficient Template Attacks. CARDIS2013, LNCS 8419, 2013.
6. Standaert, F.-X., Malkin, T.G., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. EUROCRYPT2009, LNCS 5479, pp.443-461, 2009.
7. Montminy, D.P., Baldwin, R.O., Temple, M.A., Laspe, E.D.: Improving cross-device attacks using zero-mean unit-variance mormalization. Journal of Cryptographic Engineering, Volume 3, Issue 2, pp.99-110, June 2013.

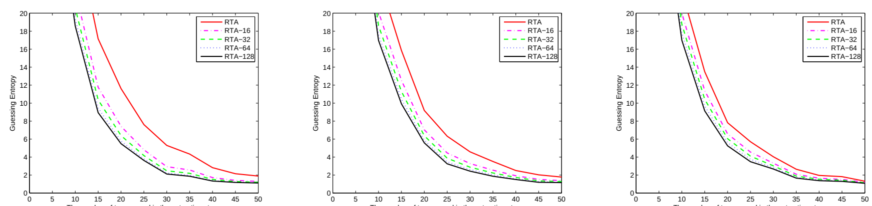
8. Oswald, E., Mangard, S.: Template Attacks on Masking—Resistance Is Futile. CT-RSA2007, LNCS 4377, pp.243-256, 2007.
9. Standaert, F.-X., Archambeau, C.: Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. CHES2008, LNCS 5154, pp.411-425, 2008.
10. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. Stochastic Methods A Performance Analysis for Side Channel Cryptanalysis. CHES2006, LNCS4249, pp.15-29, 2006.
11. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. pp.156 Springer 2007.
12. Hanley, N., Tunstall, M., Marnane, W.P.: Unknown Plaintext Template Attacks. WISA2009, LNCS 5932, pp.148-162, 2009.
13. Coron, J.-S., Kizhvatov, I.: Analysis and Improvement of the Random Delay Countermeasure of CHES 2009. CHES2010, LNCS 6225, pp.95C109, 2010.
14. Durvaux, F., Renauld, M., Standaert, F.-X. et al.: Efficient Removal of Random Delays from Embedded Software Implementations Using Hidden Markov Models. CARDIS2012, LNCS 7771, pp. 123-140, 2013.
15. Zhang, H., Zhou, Y., Feng, D.: An Efficient Leakage Characterization Method for Profiled Power Analysis Attacks. ICISC2011, LNCS 7259, pp.61-73, 2011.
16. Feng, M., Zhou, Y., Yu, Z.: EMD-Based Denoising for Side-Channel Attacks and Relationships between the Noises Extracted with Diffierent Denoising Methods. ICICS2013, LNCS 8233, pp.259-274, 2013.
17. Lehmann, E. L., Casella, G.: Theory of Point Estimation (2nd ed.). Springer. ISBN 0-387-98502-6.
18. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. pp.62-65 Springer 2007.
19. Standaert, F.-X., Gierlichs, B., Verbauwhede, I.: Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. ICISC2008, LNC-S5461, pp.253-267, 2009.
20. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. pp.40-41 Springer 2007.
21. Medwed, M., Standaert, F.-X., Joux. A.: Towards Super-Exponential Side-Channel Security with Efficient Leakage-Resilient PRFs. CHES2012, LNCS 7428, pp.193-212, 2012.
22. Yang, S., Zhou, Y., Liu, J., Chen, D.: Back Propagation Neural Network Based Leakage Characterization for Practical Security Analysis of Cryptographic Implementations. ICISC2011, LNCS 7259, pp.169-185, 2012.
23. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. pp.108 Springer 2007.
24. Schindler, W., Lemke, K., Paar, C.: A Stochastic Model for Differential Side Channel Cryptanalysis. CHES2005, LNCS 3659, pp.30-46, 2005.

Appendix A: The practical experiment results for the second S-box



(a) $np = 5,000$ (b) $np = 7,500$ (c) $np = 10,000$

Fig. 4. The experiment results of Classical Template Attacks for the second S-box



(a) $np = 5,000$ (b) $np = 7,500$ (c) $np = 10,000$

Fig. 5. The experiment results of Reduced Template Attacks for the second S-box