Security Evaluation of Cryptographic Modules against Profiling Attacks

Yongdae Kim[1], Naofumi Homma[2], Takafumi Aoki[2], and Heebong Choi[1]

[1] The Attached Institute of Electronics and Telecommunications Research Institute
{kimyd,gold}@ensec.re.kr
[2] Graduate School of Information Sciences, Tohoku University
homma@aoki.ecei.tohoku.ac.jp, aoki@ecei.tohoku.ac.jp

Abstract. Recently, profiling attacks have been attracting a great deal of attention because of their increasing efficiency. Further investigations are required to determine the potential threats of the profiling attacks. This paper focuses on these attacks. Using hardware and software implementations, we provide a security evaluation of three different types of profiling attacks: template attack, stochastic model attack, and multivariate regression attack. Our experimental results show that multivariate regression attack outperforms other attacks in terms of profiling efficiency and key extraction rates.

Keywords: profiling attack, multivariate regression analysis, template attack, stochastic model attack, power analysis attack.

1 Introduction

Cryptographic algorithms are implemented in various forms: hardware, software, firmware or sometimes in a combination of various forms. These forms are called cryptographic modules. It was believed that cryptographic modules were secure because the underlying cryptographic algorithms are theoretically unbreakable. For this reason, security evaluations were restricted to the algorithm level.

However, a new category of cryptanalysis, power analysis attack has been introduced by P. Kocher, et al. in 1999 [1]. Many cryptographic researchers have begun to investigate not only cryptographic algorithms, but also their concrete implementations.

One of the most efficient power analysis attacks is called the profiling attack, which employs reference modules that have the same characteristics as those of the target module [2], [3]. There are three methods in class of profiling attacks: the template attack [4], the stochastic model attack [5] and multivariate regression attack [6].

Testing methods for cryptographic modules have been developed in many countries under the Cryptographic Module Validation Program (CMVP). However, the methodologies for conducting security evaluation with resistance to side-channel attacks are still under discussion. Federal Information Processing Standard (FIPS) 140-2 is one security requirement for cryptographic modules in

T. Kwon, M.-K. Lee, and D. Kwon (Eds.): ICISC 2012, LNCS 7839, pp. 383–394, 2013.

USA and Canada: however, it does not contain concrete metrics for side-channel attacks. The current version of FIPS 140-2 deals with side-channel attacks as mitigations of other attacks. Therefore, the testing methods relative to side-channel attacks will be specified in the new version of the standard, FIPS 140-3. However, it is more difficult to standardize testing methods for profiling attacks than it is for other conventional side-channel attacks, such as correlation power analysis attacks [7], differential power analysis attacks [1], etc.

Because profiling attacks have several issues to be considered: (i) the selection of points that contain data-dependent variations, and (ii) the number of traces for the profiling. These two parameters have significant impacts on profiling attacks. Sometimes they lead to a decrease of performance, especially when there are a limited number of available traces. This can cause unreliable security evaluation of cryptographic modules.

To clear up these two issues, we perform security evaluation on the hardware and software implementations of the Advanced Encryption Standard (AES) to strengthen the security level of the cryptographic modules. In addition to that, we also demonstrate the effect of hiding countermeasures on software implementation. Our experimental results indicate that we can use the multivariate regression attack for an accurate and reliable security evaluation method to test the hardware and software of cryptographic modules.

2 Profiling Attacks

Compared to correlation power analysis attacks [7], profiling attacks require a far lower amount of side-channel information to retrieve the secret key, since such attacks take advantage of prior information from the profiling phase. The basic idea of this technique is to approximate the noise model rather than to reduce or eliminate noise.

Profiling attacks are known to consist of two phases: (i) profiling phase, and (ii) key extraction phase. Each phase uses different modules that have identical physical characteristics. In the profiling phase, an adversary captures physical leakage from a reference module. By analyzing that information, a property of the signal and noise can be characterized. Next, in the key extraction phase, maximum-likelihood estimation is used to determine the correct key using the information built in the profiling phase.

2.1 Template Attack

Profiling Phase. The profiling phase collects a large number of waveforms with different data d_i and key k_j, given as

$$d_i \in \{d_1, d_2, \cdots, d_D\}, \tag{1}$$

$$k_j \in \{k_1, k_2, \cdots, k_K\}, \tag{2}$$

where D and K denote the number of possible pieces of data and keys, respectively. Then, we group together the traces that correspond to the pair of (d_i, k_j), and estimate a mean vector, $\boldsymbol{m}$, and a covariance matrix, $\boldsymbol{C}$, of the multivariate normal distribution.

However, for example in case of 128-bit AES, the number of possible data and keys is $2^8 = 256$. Thus, in total, $256^2 = 65536$ templates are required. It is unrealistic to generate templates corresponding to all possible pairs of keys and data.

Therefore, in practice, templates are generated based on hypothetical power consumption for each pair of (d_i, k_j), and are written as h_{d_i,k_j}. Hence, the number of templates can be reduced. In the case of AES, we need to build only 9 templates corresponding to 9 possible Hamming distance or Hamming weight values, which are dependent on the method of implementation.

Finally, templates $(\boldsymbol{m}, \boldsymbol{C})_{h_{d_i,k_j}}$ that correspond to all possible hypothetical power consumption values are built in this phase.

Thus, the characteristic of W sampling points power consumption trace $\boldsymbol{w} = (w_1, \cdots, w_W)$ can be described as the probability density function of the multivariate normal distribution as follows :

$$\boldsymbol{q} = \boldsymbol{w} - \boldsymbol{m}, \tag{3}$$

$$p(\boldsymbol{w}; (\boldsymbol{m}, \boldsymbol{C})_{h_{d_i,k_j}}) = \frac{exp\left(-\frac{1}{2}\boldsymbol{q}^T \boldsymbol{C}^{-1} \boldsymbol{q}\right)}{\sqrt{(2\pi)^W det(\boldsymbol{C})}}, \tag{4}$$

where $det(\boldsymbol{C})$ and $\boldsymbol{q}^T$ denote the determinant of $\boldsymbol{C}$ and the transpose of vector $\boldsymbol{q}$.

Key Extraction Phase. When a power consumption trace is given, the probability $p(k_j \mid \boldsymbol{w})$ for $j = 1, \cdots, K$ is written as follows using Bayes' theorem.

$$p(k_j \mid \boldsymbol{w}) = \frac{p(\boldsymbol{w} \mid k_j) p(k_j)}{\sum_{l=1}^{K} (p(\boldsymbol{w} \mid k_l) p(k_l))} \tag{5}$$

Note that $p(k_j) = 1/K$, since all possible keys are uniformly distributed. Given a trace, $\boldsymbol{w}$, Eq. (5) indicates a probability when the correct key is equal to k_j.

The original template attack only provides a key extraction strategy based on a single available trace. However, it is difficult in practice to retrieve the correct key using only a trace. Thus, we use the following formula for given D traces: $\boldsymbol{w}_i (i = 1, \cdots, D)$.

$$p(k_j \mid \boldsymbol{w}_{1,\cdots,D}) = \frac{\left(\prod_{i=1}^{D} p(\boldsymbol{w}_i \mid k_j)\right) p(k_j)}{\sum_{l=1}^{K} \left(\left(\prod_{i=1}^{D} p(\boldsymbol{w}_i \mid k_l)\right) p(k_l)\right)} \tag{6}$$

In Eq. (6), $p(\boldsymbol{w}_i \mid k_j)$ is set to $p(\boldsymbol{w}_i; (\boldsymbol{m}, \boldsymbol{C})_{h_{d_i,k_j}})$, which is obtained in the profiling phase in Eq. (4).

Finally, we estimate the correct key k_{ck} using the maximum likelihood estimation with the probability density function, Eq. (6) as follows :

$$k_{ck} = \underset{k_j \in k^*}{\operatorname{argmax}} p(k_j \mid \boldsymbol{w}_{1,\cdots,D}), \tag{7}$$

where k^* is the set of all possible key candidates.

2.2 Stochastic Model Attack

In 2005, the stochastic model attack was introduced by W. Schindler, et al. [5]. The fundamental idea of this attack is very similar to that of the template attack. However, the stochastic model attack uses the key-independent noise model instead of the usage noise model associated to all possible key candidates in template attacks.

Profiling Phase. In stochastic model attack, a power trace at time t ($t = 1, \cdots, W$) is represented as,

$$I_t(d_i, k) = h_t(d_i, k) + r_t, \tag{8}$$

with the i-th input d_i and a correct key (which is, however, known to an adversary) k. And $h_t(d_i, k)$ denotes the deterministic part of the trace depending on d_i and k. On the other hand, r_t denotes a random part independent of d_i and k. The profiling phase is divided into two steps in order to approximate the two discrete terms.

In the first step, the deterministic part is profiled using N_1 traces from a reference module. The deterministic part is approximated by a linear combination of u-dimensional vector subspace spanned by the u known function $g_{j,t}$,

$$\hat{h}_t(d_i, k) = \sum_{j=0}^{u-1} \beta_{j,t} \cdot g_{j,t}(d_i, k), \tag{9}$$

where the coefficients $\beta_{0,t}, \cdots, \beta_{u-1,t}$ are estimated value for each instant t. In order to estimate the coefficients $\boldsymbol{\beta}_t = (\beta_{0,t}, \cdots, \beta_{u-1,t})$, the function $g_{j,t}$ is firstly determined in the u-dimensional subspace. For example, in the case of AES, a 9 dimensional subspace is usually chosen as the function $g_{j,t}$, which leads to the best approximation [5]. An adversary generates a matrix, $\boldsymbol{A}$ using N_1 traces corresponding to input d_i and key k as follows:

$$\boldsymbol{A} = \begin{bmatrix} g_{0,t}(d_1, k) & \cdots & g_{u-1,t}(d_1, k) \\ g_{0,t}(d_2, k) & \cdots & g_{u-1,t}(d_2, k) \\ \vdots & & \vdots \\ g_{0,t}(d_{N_1}, k) & \cdots & g_{u-1,t}(d_{N_1}, k) \end{bmatrix} \tag{10}$$

The estimated coefficients

$$\boldsymbol{\beta}_t = (\beta_{0,t}, \beta_{0,1}, \cdots, \beta_{u-1,t}), \tag{11}$$

are then denoted using the least square method,

$$\boldsymbol{\beta}_t = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{w}_t, \tag{12}$$

where the vector $\boldsymbol{w}_t = (w_{1,t}, w_{2,t}, \cdots, w_{N_1,t})$ represents power consumption for each instant t.

After having determined the approximators $\hat{h}_t(d_i, k)$, different set of N_2 traces are used to profile the random part. We first calculate the W-dimensional random vector $\boldsymbol{r} = (r_1, r_2, \cdots, r_W)$ as follows:

$$r_t = I_t(d_i, k) - \hat{h}_t(d_i, k), \tag{13}$$

We assume that the random vector is normally distributed with a covariance matrix $\boldsymbol{C}$. Therefore, $\boldsymbol{C} = (c_{i,j})_{1\le i,j\le W}$ is computed as follows:

$$\begin{aligned} c_{i,j} &= E(r_i r_j) - E(r_i)E(r_j) &(14)\\ &= E(r_i r_j) &(15)\end{aligned}$$

where $E(X)$ denotes the expected value of the variable X. Finally, we have approximated the deterministic part, $\hat{h}_t(d_i, k)$ and the noise model represented as the covariance, $\boldsymbol{C}$ in this phase.

Key Extraction Phase. In this phase, traces from a target module are analyzed using the model that have been obtained in the profiling phase. We assume that N_3 traces are captured from the target module corresponding to $d_i \in \{d_1, d_2, \cdots, d_{N_3}\}$ and that there is a correct key, k_{ck} (unknown to an adversary). A noise vector $\boldsymbol{z}_i$ is first computed as follows:

$$\boldsymbol{z}_i = I_t(d_i, k_{ck}) - \hat{h}_t(d_i, k_j), \tag{16}$$

where $k_j \in \{k_1, k_2, \cdots, k_K\}$. The noise vector follows a multivariate normal distribution with the profiled covariance matrix $\boldsymbol{C}$ when $j = ck$. So, we can estimate the correct key by computing the following probabilities:

$$p\left(\boldsymbol{z}_i; \hat{h}_t(d_i, k_j)\right) = \frac{exp\left(-\frac{1}{2}\boldsymbol{z}_i^T\boldsymbol{C}^{-1}\boldsymbol{z}_i\right)}{\sqrt{(2\pi)^W det(\boldsymbol{C})}}. \tag{17}$$

The maximum likelihood estimation is applied to determine a correct key k_{ck} using N_3 traces as follows:

$$k_{ck} = \underset{k_j \in k^*}{\operatorname{argmax}}\, \Pi_{i=1}^{N3} p\left(\boldsymbol{z}_i; \hat{h}_t(d_i, k_j)\right). \tag{18}$$

We can simplify the Eq. (17) by applying the logarithm.

$$\begin{aligned} &ln\left(p\left(\boldsymbol{z}_i; \hat{h}_t(d_i, k_j)\right)\right)\\ &= -\frac{1}{2}\boldsymbol{z}_i^T\boldsymbol{C}^{-1}\boldsymbol{z}_i - \frac{1}{2}ln\left((2\pi)^W det(\boldsymbol{C})\right). \end{aligned} \tag{19}$$

The second term $-\frac{1}{2}ln((2\pi)^W det(\boldsymbol{C}))$ in Eq. (19) is constant value. Therefore it can be eliminated. As a result, the estimator can be simplified as a follows:

$$\boldsymbol{z}_i^T \boldsymbol{C}^{-1} \boldsymbol{z}_i. \tag{20}$$

Note that an adversary decides the correct key that minimized the sum of Eq. (20) as follows:

$$k_{ck} = \underset{k_j \in k^*}{\operatorname{argmin}} \sum_{i=1}^{N_3} \boldsymbol{z}_i^T \boldsymbol{C}^{-1} \boldsymbol{z}_i. \tag{21}$$

2.3 Multivariate Regression Attack

This type of profiling attack can improve profiling efficiency using multivariate regression analysis. Even if an adversary can utilize several traces for profiling, the adverse effects for the key extraction can be minimized.

Profiling Phase. In the profiling phase, we need to determine explanatory and response variables to build a multivariate regression model.

Define Response Variable. We consider the response variable as the sum of all hypothetical power consumption of the components (i.e. S-Boxes). Assuming that M components are processed in parallel, we write the l-th component (hypothetical power consumption) as $h_{i,ck}^l$, given by the i-th input ($1 \leq i \leq N$, N is a number of inputs) and the correct key k_{ck}. We referred the sum of each hypothetical power consumption to s_i, which is defined

$$s_i = \sum_{l=1}^{M} h_{i,ck}^l. \tag{22}$$

We defined the value of s_i as a response variable in the regression model. Note that the value s_i is feasible only if the correct keys are known to the adversary. It is possible that the adversary will use a reference module under full control.

Define Explanatory Variables. First of all, we calculate a squared Pearson correlation coefficient vector, $\boldsymbol{\rho}'_B = (\rho_{B,1}^2, \cdots, \rho_{B,W}^2)$ between $\boldsymbol{w}_i$ and s_i considering both negative and positive correlation. If the squared coefficient is high at the t-th time instant, it is usually assumed that the time instant is highly related to the response value, s_i. Thus, the adversary select $P(< W)$ instans with the highest value of the squared correlation coefficient, and referred to as a vector $\boldsymbol{p} = (p_1, p_2, \cdots, p_P)$. Each time instant is sorted in descending order of the value of squared correlation. This means that the squared Pearson correlation have the highest value at time instant, p_1. Now, we select P time instants from $w_{i,t}$ $(1 \leq t \leq W)$ corresponding to the interesting points. We define the the explanatory variables as followings:

$$w_{i,p_1}, w_{i,p_2}, \cdots, w_{i,p_P} \ (1 \leq i \leq N_1), \tag{23}$$

where N_1 is the number of inputs for the profiling phase.

Multivariate Regression Model. Next, we can calculate the estimator of regression coefficients. Finally, the following multivariate regression model can be obtained using the regression coefficients in the profiling phase, $(1 \leq i \leq N_1)$

$$\hat{s_i} = \hat{\beta}_0 + \hat{\beta}_1 w_{i,p_1} + \hat{\beta}_2 w_{i,p_2} + \cdots + \hat{\beta}_P w_{i,p_P}, \tag{24}$$

where $\hat{s_i}$ stands for the fitted value using the model. Finally, we obtain the regression coefficients in this phase corresponding to its interesting points.

Key Extraction Phase. In this phase, an adversary measures power traces from a target cryptographic module corresponding to N_2 inputs (known) and secret key k_{ck} (unknown). We utilize the regression model in Eq. (24) to estimate the sum of hypothetical power consumption, $\hat{s_i}$ $(1 \leq i \leq N_2)$ given by the measured traces. h^l_{i,k_j} denotes the hypothetical power consumption of the l-th component (S-Box) associated with each key candidates $k_j \in \{k_1, k_2, \cdots, k_K\}$ and the i-th input. The correct key of the l-th component can be estimated using the Pearson correlation coefficient between $\hat{s_i}$ and h^l_{i,k_j} as follows:

$$k^l_{ck} = \underset{k_j \in \{k_1, k_2, \cdots, k_K\}}{\text{argmax}} \; corr(\hat{s_i}, h^l_{i,k_j}). \tag{25}$$

3 Experimental Analysis

For the hardware implementation, the side-channel attack standard evaluation board (SASEBO) incorporating cryptographic FPGA and ASIC were used for both the target and the reference module. We used an 8-bit AVR microcontroller from the Atmel Corporation for software implementation. In addition, we implemented a hiding countermeasure on the software implementation to investigate of effectiveness of the countermeasure. Ten delays before our target operation (i.e. SubBytes) are inserted. Each delay is composed of several dummy operations. The number of operations is uniformly generated between 0 and 8. Therefore the longest possible delay time is 80 clock cycles.

3.1 Evaluation on Hardware Implementations

We used 15 interesting points and 20,000 traces for profiling for both FPGA and ASIC implementations. In our analysis, the number of available power traces for key extraction is 10,000 and 20,000 for FPGA and ASIC, respectively.

The result of all subkey estimation results are illustrated in Fig. 1. The horizontal axis indicates the number of traces for key extraction: the vertical axis shows the classification rate in percentage computed as follows:

$$R_i = \frac{N^{ck}_i}{16} \times 100, \tag{26}$$

where R_i and N^{ck}_i denote the classification rate and the number of correctly estimated keys using i traces, respectively.

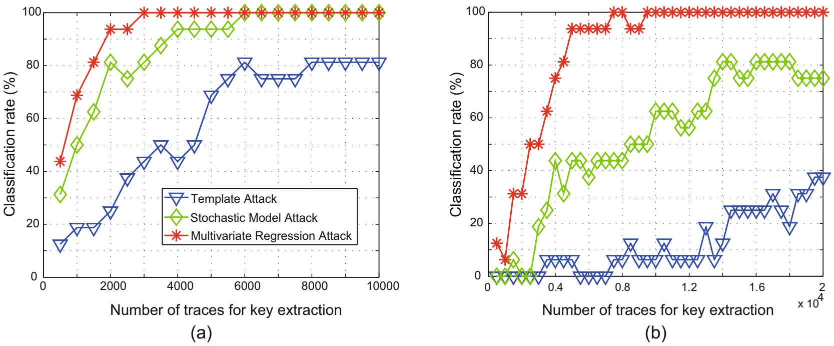


Fig. 1. Classification rate (a) FPGA, and (b) ASIC implementation

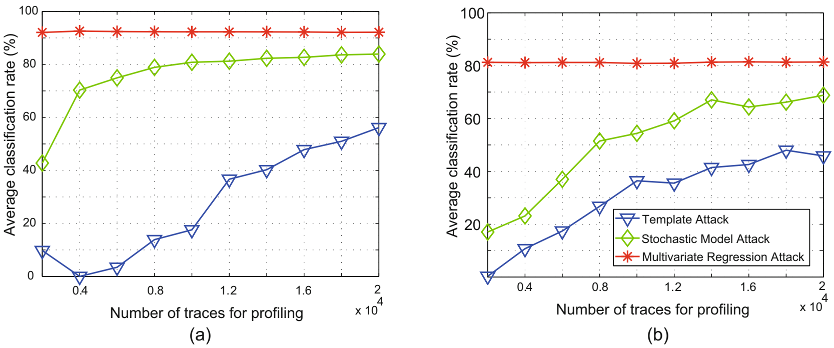


Fig. 2. Average classification rate of FPGA and ASIC implementations associated with the number of traces for profiling (15 interesting points): (a) FPGA implementation (b) ASIC implementation

As can be seen in the Fig. 1, the classification rate is increasing as the number of traces for key extraction increases. However, the multivariate regression attack has the least number of traces to estimate all the subkeys of AES in FPGA and ASIC implementations. In other words, a straightforward AES implementation on FPGA and ASIC is very vulnerable to multivariate regression attack.

In order to examine the effects of the number of traces for profiling and interesting points, we first define an average value of classification rates, $\bar{R}$ as follows:

$$\bar{R} = \frac{1}{D} \sum_{i=1}^{D} R_i, \tag{27}$$

where D represents the number of total available traces for key extraction. In our case, D is 10,000 and 20,000 for FPGA and ASIC, respectively.

Figure 2 shows the average values of classification rates of FPGA and ASIC implementations: the horizontal axis is the number of traces used in the profiling

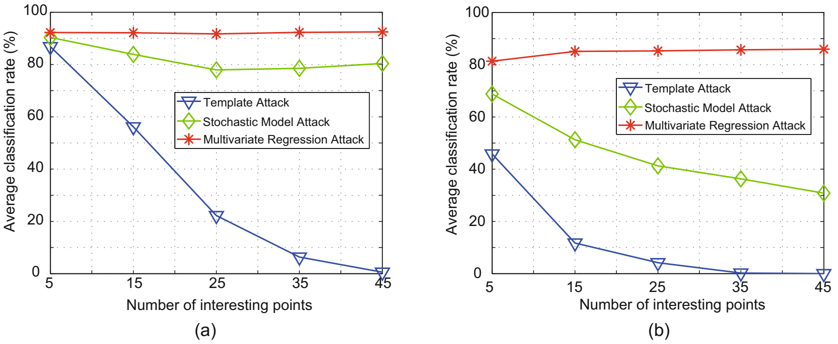


Fig. 3. Average classification rate of FPGA and ASIC implementations associated with the number of interesting points (20,000 traces for profiling): (a) FPGA implementation, and (b) ASIC implementation

phase and the number of interesting points is fixed at 15 points. We do not focus on the selection method of interesting points. However, we do only investigate selection method impact on the classification rate, and thus we assumed that an adversary had already obtained a fixed number of interesting points in the profiling phase.

Figure 3 shows the average classification rates of FPGA and ASIC implementations associated with the number of interesting points: where the number of traces for profiling is 20,000 for both implementations. The results indicate the low average values of the classification rates for the template attack and stochastic model attack as the number of interesting points is increased. The interesting points cover the data dependent time instants. Sometimes, when an adversary selects non-data dependent points as interesting points, the classification rate is negatively affected by these points.

However, it can be clearly observed that the classification rates using multivariate regression attack does not decrease even when the number of interesting points is increased comparing to other profiling attacks. This is because the less data-dependent time instants have a less significant effect on the regression model.

To observe the significance of each regression coefficient, we intentionally selected 20 irrelevant time instants adding them to the 15 selected interesting points, as shown in Fig. 4. The results show squared values of regression coefficients corresponding to the 35 selected points. This clearly confirms that the time instants corresponding to irrelevant points have relatively small values of regression coefficients. This leads to a small impact on the response variable. Thus, a profiling attack using the multivariate regression model takes less effort to determine the time instants as interesting points in order to extract all keys successfully.

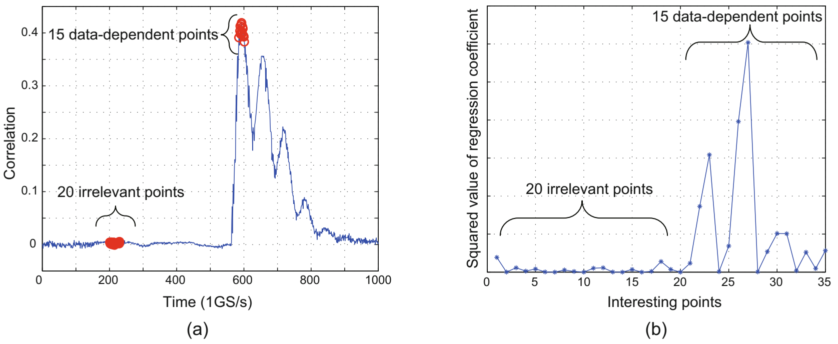


Fig. 4. (a) correlation peaks and interesting points, and (b) squared value of regression coefficient for FPGA implementations

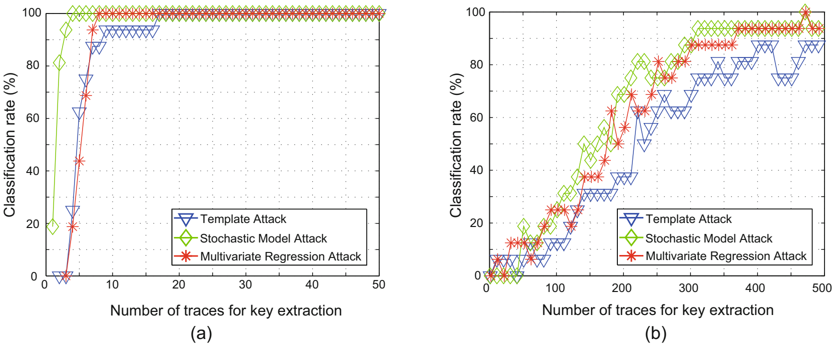
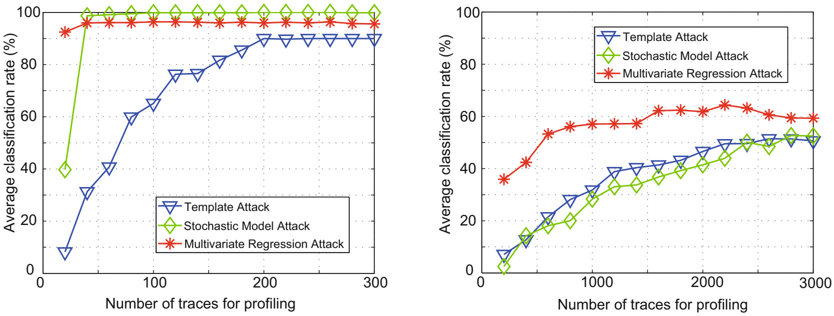


Fig. 5. Classification rate (a) unprotected smart card, and (b) hiding countermeasure implemented smart card

3.2 Evaluation on Software Implementations

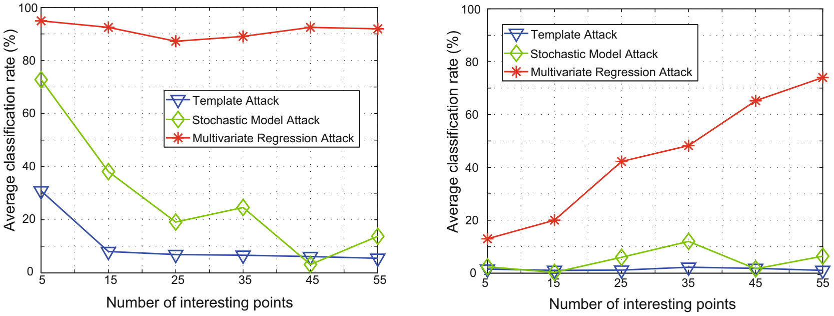
Compared to the case of hardware implementations, in software implementations all S-boxes are executed sequentially. Hence we determined different 15 different interesting points for each target S-boxes.

In Fig. 5, we show the classification rates associated with the number of traces for key extraction for an unprotected and hiding countermeasure implemented smart card. For those results in Fig. 5, we used 15 interesting points for both types of implementations: 2,000 and 5,000 traces are used for the profiling phase on the unprotected and hiding countermeasure implemented smart cards, respectively. For the unprotected smart card, it is enough to extract all subkeys using fewer than 20 traces. As can be seen in Fig. 5 (b), even when we used 10 times more power traces than those used with the unprotected smart card, it was impossible to retrieve all the secret keys successfully. This result shows that hiding countermeasure is effective but not perfect method to hinder attacks. Actually, the classification rate increases as the number of traces increases.



(a) (b)

Fig. 6. Average classification rate associated with the number of traces for profiling (15 interesting points): (a) unprotected smart card, and (b) hiding countermeasure implemented smart card



(a) (b)

Fig. 7. Average classification rate associated with the number of interesting points: (a) unprotected smart card (20 traces for profiling), and (b) hiding countermeasure implemented smart card (100 traces for profiling)

First, we examine a tendency between the number of traces for profiling and the classification rates. The experimental procedure is the same as that of the hardware implementations experiment. Figure 6 shows the results of this experiment. These results clearly indicate the visible tendencies between the classification rates and the number of traces available for profiling. In addition, the multivariate regression attack has a higher profiling efficiency than that of other profiling attacks for both smart cards. As we already demonstrated experimental results for hardware implementations, we have found that profiling strategies have almost the same performance when a certain number of traces are available in software implementations.

Next, we focus on the efficiency of the profiling phase with the number of interesting points. Figure 7 shows the average classification rate associated with the number of interesting points. Note that we used exactly the same interesting points for each number of traces for profiling. As can be seen in Fig. 7, with respect to hiding countermeasure implementation, the average classification rates show a high improvement as more interesting points are provided. This is because the number of relevant time instants are increased with hiding countermeasures.

4 Conclusion

This paper presented a security evaluation of cryptographic modules against profiling attacks. The profiling attack is one of the side-channel attacks that most effectively expose weaknesses and secret information of cryptographic modules using their physical leakages. However, profiling attacks require a large number of traces to characterize of the power consumption and relevant time instants correctly. The multivariate regression attack is able to compensate for those two issues.

Our evaluation results of hardware and software implementations have shown that multivariate regression attacks, pose a serious threat to the security level of cryptographic modules. The results indicate that we need to consider multivariate regression attacks for a proper security evaluation of profiling attacks, because the attack can perform successfully using a small number of traces in the profiling phase. In addition, such attacks are robust to selection methods of relevant time instants.

Acknowledgements. We would like to gratefully thank Dr. Takeshi Sugawara for his useful support for performing the experiments.

References

1. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
2. Le, T.H., Canovas, C., Clédière, J.: An overview of side channel analysis attacks. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security (ASIACCS), pp. 33–43 (2008)
3. 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. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 253–267. Springer, Heidelberg (2009)
4. Chari, S., Rao, J., Rohatgi, P.: Template attacks. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003)
5. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005)
6. Sugawara, T., Homma, N., Aoki, T., Satoh, A.: Profiling attack using multivariate regression analysis. IEICE Electronics Express 7, 1139–1144 (2010)
7. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)