Jurnal Informatika, Vol. 13, No. 2, November 2015, 56-62 DOI: 10.9744/informatika.13.2.56-62
ISSN 1411-0105

THE CPA QUALIFICATION METHOD BASED ON THE GAUSSIAN CURVE FITTING

M.T. Adithia
Informatics Department, Parahyangan Catholic University
Jl. Ciumbuleuit No. 94, Bandung, 40141
Email: mariskha@unpar.ac.id

Abstract: The Correlation Power Analysis (CPA) attack is an attack on cryptographic devices, especially smart cards. The results of the attack are correlation traces. Based on the correlation traces, an evaluation is done to observe whether significant peaks appear in the traces or not. The evaluation is done manually, by experts. If significant peaks appear then the smart card is not considered secure since it is assumed that the secret key is revealed. We develop a method that objectively detects peaks and decides which peak is significant. We conclude that using the Gaussian curve fitting method, the subjective qualification of the peak significance can be objectified. Thus, better decisions can be taken by security experts. We also conclude that the Gaussian curve fitting method is able to show the influence of peak sizes, especially the width and height, to a significance of a particular peak.

Keywords: Cryptography, side channel attack, correlation power analysis, smart cards, significant peak detection, Gaussian curve fitting

INTRODUCTION

Cryptographic devices [1] are electronic devices that implement a cryptographic algorithm and that store keys. An example of a cryptographic device is a smart card. A smart card is a device that has the same size as a credit card. It is able to store data and to process data by using an integrated chip. To process data, the chip performs a cryptographic algorithm that employs a secret key. Any attempt to extract the keys stored in the cryptographic device in an unauthorized way is called an *attack*. One class of attacks that poses serious threat to the security of cryptographic devices are the *side-channel* attacks. A side-channel attack is an attack applying information gained from the physical implementation of a cryptographic device, for example timing information, power consumption, and electromagnetic leaks.

One type of side-channel attacks is *Correlation Power analysis* (CPA) attacks. This type of attack is a refinement of another type of side channel attacks called Differential Power Analysis (DPA) attack, that was first introduced in 1999 in [2]. The CPA attack, which was introduced [3] in 2004, is a multi-bit DPA taking into account the linear relationship between the power consumption curve and the Hamming model. In general, this attack exploits the fact that the power consumption of a cryptographic device depends on the data it processes and the operation it performs [1]. By conducting this attack, an attacker may obtain the secret keys used in the cryptographic algorithm employed by the device.

In this paper, we focus on the CPA attack on smart cards. CPA is relatively easy to be carried out and has a high success rate. It is not necessary for the attacker to have detailed knowledge about the smart cards. It is sufficient to know the steps of the cryptographic algorithm that is executed by the smart cards. That is why a lot of research is done to improve the security of smart cards against this attack.

The result of the CPA attack is represented by *correlation traces* [3]. Based on the correlation traces, an evaluation is done to observe whether significant peaks appear in the traces or not. If significant peaks appear then the smart card is not considered secure since it is assumed that the secret key is revealed. If there are no significant peaks, the smart card is secure. The higher and steeper the peaks, the stronger the attack and the less secure the smart card is.

The difficulty is to objectively decide whether a peak is significant enough to be called a peak. To support the decision making process, we develop a method to detect peaks and to decide which peak is significant.

THE CPA ATTACK

The CPA attack is based on two important concepts, i.e., *leakage function* and *bit/byte trace*.

A *leakage function* [4] is an abstraction used to represent the physical output of a side-channel, monitored by some measurement setup. The input of a leakage function is a plaintext that will be processed by a cryptographic device. In the CPA attack, the output of this leakage function is the power consumption of the cryptographic device sampled with a fixed sampling frequency while processing the input plaintext. In this project, the output of the leakage function is called a *power trace*.

Practically, a power trace from a smart card is obtained by measuring the power consumption of the smart card while processing a binary input. The power trace is not the end result of the process, but the intermediate result. For example, if a smart card employs some cryptographic algorithm with several rounds where each round uses one specific secret key, the power trace is taken after one round is finished. A byte trace [1] is an approach to monitor a predictable byte during the course of the process. In the context of a power analysis, the byte trace approach is applied to check leakage of some cryptographic device. The result of the byte trace approach is a correlation coefficient between the input and the power trace at one time. All the resulted correlation coefficient is called *correlation traces*.

The investigation to check whether the smart card is leaking is done based on the correlation trace plot (see Figure 1). If there is a high peak on the plot, it means that the investigated byte has a high correlation with the power consumption at the time point at which this high peak appears. This fact already shows that there is some information leaking from the smart card.

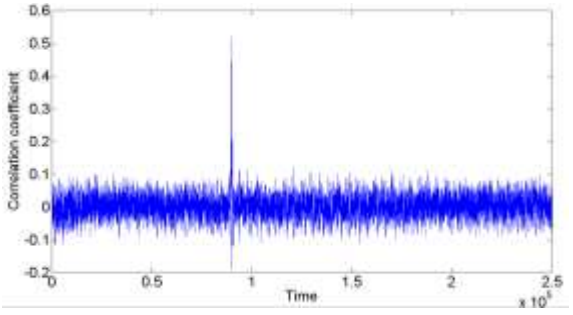

Figure 1: The plot of correlation coefficient between the input and the power trace.

More detailed explanation about the steps of the CPA attack can be found in [1].

RELATED WORKS

[5] discusses methods to evaluate and compare side-channel attacks. Some simple numerical examples of leakage function and some illustration how the functions could be evaluated and understood are given in [6]. The methods are based on two metrics: *information theoretic* and *security* metric. However, these two metrics cannot be used to solve our problem since the metrics need a lot of power traces, where each is obtained using different sets of input plaintexts. The more power traces are provided, the more accurate the results are. The fact is, power trace measurements are very expensive. Thus, carrying a lot of measurements to get results for one smart card is not practical for companies.

A method to detect peaks is also discussed in [7] by using short-time FFT. The method also includes noise removal techniques. The method is developed for MALDI data, which has different behavior from our data.

[8] and [9] introduces peak detection methods using wavelet transformation. The methods consider some characteristic shapes to identify peaks. However, the characteristic shapes introduced in this paper cannot be adapted in our problem.

A method to quantify peak is discussed in [10]. The method is developed for mass spectrum related to protein mixtures. The mass spectrum contains peaks corresponding to proteins in a sample. A statistical mixture model is developed to quantify peaks. However, the quantification mostly depends on peak height.

THE SIGNIFICANT PEAK DETECTION APPROACH

Our approach to determine whether a peak is significant or not consists of two main methods. We first develop a method to assign a score to each peak found in a correlation trace. This method is based on the Gaussian curve fitting method. Second, based on the resulted peak scores, we determine whether a peak is significant or not using the *Absolute Score Distance* computation and the clustering analysis.

The Gaussian Curve Fitting Method

We develop a method based on the Gaussian curve fitting method to give a score to each peak found in a correlation trace.

Since the correlation traces typically have too many sample points, we downsample it first. The resulted downsampled correlation trace is put in a vector called local_maxima. The main idea of this approach is to fit a curve to the local_maxima vector of each correlation trace and qualify each peak found in the new curve. We choose a sum of several Gaussian functions to fit our correlation trace local maxima. The Gaussian function is formulated as follows:

$$f(x) = a \cdot e^{-(\frac{x-b}{2c})^2} \quad (1)$$

with a is the height of the curve, b is the center of the curve, and c is the width of the curve.

Since the values of a and b can be obtained from the correlation trace, we only have to estimate c before we can apply the curve fitting function.

Suppose that we have a set of points $\{(x_1, y_1), \ldots (x_n, y_n)\}$ that we want to fit to a curve. Consider Equation 1, suppose that $a=y_t$ and $b=x_m$. The value of c is estimated by using the following steps:

1. Compute c_i for each (x_i, y_i) with given a and b using the following formula which is based on Equation 1.

$$c(i) = \frac{|x(i)-b|}{\sqrt{-\log(\frac{y(i)}{a})}} \quad (2)$$

2. Take the mean and the standard deviation of c_i, denoted by $\bar{c}$ and S, respectively.
3. Form a vector $V1$ that contains all values of c_i that are less or equal than $\bar{c}$. Take the mean of $V1$ and denote it with $\overline{v1}$
4. Form a vector $V2$ that contains all values of c_i that are less or equal than $\bar{c} + S$. Take the mean of $V2$ and denote it with $\overline{v2}$
5. Set the value c as

$$c = \frac{\overline{v_1} + \overline{v_2}}{2} \quad (3)$$

After we get c, we are ready to apply the Gaussian Curve Fitting to our correlation traces. The general algorithm we use to apply the method is given below.

The input of the algorithm are X, which is a correlation trace with $X(i)$ is the correlation coefficient for sample i, and window_size, the size of the sliding window. The output are the peak location, the peak height, the peak area, the peak inverse width, the scores of all peaks found, and the normalized scores of all peaks found. The steps of the algorithm are as follows:

1. Apply the sliding window of size window_size to the absolute value of the correlation trace X and find the global maximum of all values within the window. Slide the window without overlapping and repeat the same operation until the window reaches the last sample. Form a vector local_ maxima that contains the resulted global maximum. By performing this process, we replace all values within a window with the global maximum of values within the window. The global maximum is chosen to represent values within one window because we are interested in significant peaks.
2. Determine shorter length vectors from the vector local_maxima such that each smaller vector consists of at least one maximum and minimum values. Each smaller vector belongs to one Gaussian curve fitting function and should contain at least three members. The shorter length vectors are forms by using the following steps:
   (a) Suppose that local_maxima = $\{l_1, l_2, \ldots, l_q\}$ with l_i the ith member of the local_maxima vector and q the length of local_maxima. Suppose that we start from l_r with $l_{r-1} \leq l_r$. Form the uth shorter length vector $SV_u = \{l_{r-1}, l_r\}$.
   (b) If $l_{r+1} \leq l_r$ and $l_{r+1} < l_{r+2}$, set $SV_u = \{l_{r-1}, l_r, \ldots, l_{r+1}\}$ and stop forming SV_u. Otherwise, repeat this step for $\{l_{r+2}, l_q\}$ until the condition holds.
   (c) Form $SV_{u+1} = \{l_{w-1}, l_w,\}$ with l_{w-1} is the last member of SV_u. Go to Step (b) for checking the condition for l_{w+1}.

   See Figure 2.
3. Determine the value of a and b for each shorter length vector, see Equation 1.
4. Estimate the value of c for each shorter length vector, see Equation 3, by using the steps explained previously.
5. Determine the peak location for each shorter length vector. Since the position of the peak need not to be a sample point, we increase the resolution with a factor 10.
6. Obtain the peak properties, i.e., peak height and inverse width, from the value of a and $\frac{1}{c}$, respectively.
7. Compute the area below each Gaussian function to get the peak area. Suppose that $\{l_v, l_{v+1}, \ldots, l_{v+m}\}$ is a shorter length vector that belongs to one Gaussian function. The area A is computed as:

$$A = \int_{l_v}^{l_{v+m}} a \cdot e^{-(\frac{x-b}{2c})^2} dx \quad (4)$$

$$= \left(\frac{2ac^2}{x-b}\right) e^{-(\frac{x-b}{2c})^2} \Big|_{x=l_v}^{l_{v+m}} \quad (5)$$

8. Since the peak properties found have different scale values, rescale each peak property so that the values are between 0 and 1. This can be done using the following way. Suppose $Y_p = \{y_{1,p}, y_{2,p}, \ldots, y_{m,p}\}$ is a set of peak property values with m the number of peaks found, p is referring to a peak property, and i is referring to the ith peak found, $rescaled\_y_{i,p}$ is computed as $rescaled\_y_{i,p} = \frac{y_{i,p}}{\max(Y_p)}$

9. Suppose p_1, p_2, and p_3 are the three peak properties defined previously, namely the peak height, the peak area, and the peak inverse width. Also suppose that $S=\{s_1, s_2,\ldots,s_m\}$ is a set peak scores with m the number of peaks found and i is referring to the ith peak. Compute the peak score s_i by using the following formula:

$$s_i = rescaled\_y_{i,p1} \,.\, rescaled\_y_{i,p2} \,.\, rescaled\_y_{i,p3} \,. \quad (6)$$

The scores obtained are between 0 and 1.

10. Compute the normalized peak score $norm\_s_i$ for the ith peak found as

$$norm\_s_i = \frac{s_i}{\sum_{j=1}^{m} s_j} \quad (7)$$

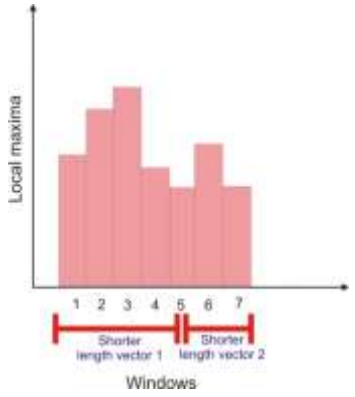

Figure 2: The plot of the local maxima and the shorter length vector division.

Figure 3 gives an example of the local maxima plot of a correlation trace along with the curve fitted to it.

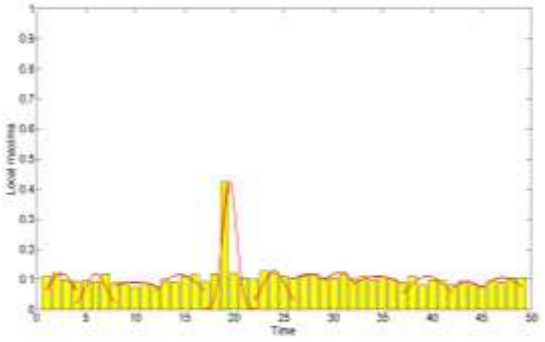

Figure 3: The plot of the local maxima and a curve fitted to it

To maintain the stability of the resulted scores that can reduce because of the data down sampling, we employ four different window sizes to get the scores. We start the score calculation from the highest window size to the lowest. Therefore, the result from this method is a matrix with each row consist of a set of four scores (each score obtained by applying one window size) for each peak found in the correlation trace.

The Peak Score Evaluation

After we obtain scores for all peaks found in a correlation trace, we would like to investigate whether the peak obtaining the highest score is a significant peak or not. We develop two methods to check properties of the highest score peak when it is compared with other peaks. The decision whether a peak is significant or not is made based on the results given by all three methods. Thus, the methods do not work independently. The methods are explained below.

Average score distance

One characteristic of a peak to be a significant peak is that the peak score should be a lot greater then those of the other peaks. Therefore, we compute the average score distance between the highest peak score with other peak scores in one correlation trace. The computation is done as follows.

Suppose that $S_i = \{s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}\}$ is a multivariate score of the ith peak after $window_1$, $window_2$, $window_3$, and $window_4$ are applied, respectively. The average score distance is computed by the following steps:

1. Compute the Euclidean distance $d_{i,j}$ of every S_i and S_j with $i \neq j$ and $i,j \in \{1,\ldots,m\}$ as

$$d_{i,j} = \left(\sum_{k=1}^{4} (s_{i,k} - s_{j,k})^2 \right)^{\frac{1}{2}} \quad (8)$$

2. Compute the average Euclidean distance $avdist_i$ of the ith peak and all other peak by using the following formula

$$avdist_i = \frac{\sum_{k=1}^{m} d_{i,k}}{m-1} \quad (9)$$

3. Compute the average score distance D as

$$D = \frac{\max(avdist)}{\max(absscore)} \quad (10)$$

Cluster analysis

We consider that cluster analysis is useful to show that a peak is significant or not. If a peak is significant, then we assume the peak score is really different with the rest of the scores. By applying a cluster analysis, we would like to show that if a peak

is significant then its score becomes a unique member of a cluster, while the other scores are clustered in one different cluster.

In practice, we use cluster analysis on the multivariate peak scores obtained, to form two clusters of peak scores within one correlation trace. The peak score clustering is done using the Statgraphics Centurion software. At the moment we use Ward's method (see [11]) in clustering the peak scores, with Euclidean distance as a method to compute the distance between two peak scores. Other clustering method may also be used without significant result differences. If the highest peak score is a unique member of a cluster, then the possibility that the highest peak score is significant becomes more likely.

We consider that a significant peak should have a score of at least 0.50. The score of 0.50 is taken based on the idea of probability theory. A probability of 0.50 means there is an equal chance that an event to happen or not. We perform more analysis to the clustering analysis results using the following steps:

1. Consider the cluster containing the highest score peak
2. If the cluster has one member and the score of the member is higher than 0.50, then the member is a significant peak.
3. If the cluster has more than one members, check the scores of all members. If all members' scores are higher than 0.50, then the highest score peak is a significant peak. If not, then the highest score peak is not significant.

EXPERIMENTAL RESULTS

We were provided with three data sets by Brightsight B.V., a security evaluator laboratory located in Delft, the Netherlands. The data sets were sampled from a smart card, with a sampling frequency of 500 MHz, while processing input plaintexts. The operation used in the process is a 16 rounds of an XOR operation defined as $c = p + k$, with c a ciphertext, p an input plaintext, and k a secret key. Each data set consists of power traces and 16 correlation traces taken from 250000 time points; each correlation trace obtained from each processing round.

The first data set, called Data_No Countermeasures, was obtained from a smart card without any countermeasures. The second and third data sets, called Data_Few Dummy Cycles and Data_More DummyCycles, respectively, are data sets obtained with some dummy cycles. Dummy cycles are processes that are more or less identical to each other. Practically, the dummy cycles do nothing and they are irrelevant to the process carried out by the smart card. The dummy cycles are inserted randomly based on hardware random function, to make the smart card more secure.

We apply the Gaussian curve fitting method to the 16 correlation traces obtained from the byte trace approach of each data set. The results from this step for each correlation trace are a list of all peaks found in the trace along with the peak properties and the score for each peak. Table 1 shows the result using one window size, i.e., 1250 samples, on the first correlation trace of the Data_NoCountermeasures.

In Table 1, it is shown that there are 15 peaks found in the correlation trace. All the peak properties are normalized so that the values are between 0 and 1. The fifth peak is the highest scored one, with a score of 0.7073. We can observe that using the Gaussian curve fitting method, we can replace the original correlation trace with scores.

Table 1: The peak properties and scores of the first correlation trace obtained from the byte trace approach of Data_NoCountermeasures with window size 1250 samples

Peak	Location	Rescaled			Normalized score
		Height	Area	Inverse width	
1	7	0.2211	0.2495	0.4795	0.0214
2	16.5	0.1922	0.1214	0.9495	0.0181
3	30.5	0.1924	0.3623	0.3251	0.0184
4	46	0.1869	0.1559	0.8307	0.0197
5	62.5	1.0000	1.0000	0.8697	0.7073
6	79.5	0.2183	0.2147	0.6266	0.0239
7	92	0.2286	0.2423	0.7204	0.0325
8	104.5	0.2055	0.2104	0.5424	0.0191
9	113.5	0.1975	0.1230	1.0000	0.0198
10	122.5	0.2184	0.2085	0.6832	0.0253
11	135	0.2175	0.2695	0.5075	0.0242
12	147.5	0.2116	0.1753	0.9528	0.0279
13	157.5	0.1797	0.1498	0.6917	0.0151
14	172	0.1742	0.3344	0.2591	0.0123
15	190.5	0.1829	0.2893	0.3517	0.0151

After that we also apply the evaluation methods to determine average score distances and to decide whether the highest score peak is significant or not. In this section, we provide the evaluation method results on peak scores computed for four window sizes, i.e., [2200 1500 800 100] samples. The results are given in Table 2, 3, and 4. Each table contains average score distance, significant peak decisions, the location of the highest score peak, and the height of the highest score peak, for 16 correlation traces taken from each Data_NoCountermeasures, Data_FewDummyCycles, and Data_MoreDummyCycles. The significant peak decisions give values 0 and 1. The value 0 indicates that the highest score peak is not significant, and the value 1 indicates the opposite.

Table 2: The results of the evaluation methods on peak scores computed based on four window sizes, i.e., [2200 1500 800 100] of Data_NoCountermeasures

Correlation trace	Average score distance	Significant peak	Peak location	Peak height
0	0.9729	1	89639	0.5190
1	0.9567	1	90565	0.4114
2	0.9556	1	91583	0.4233
3	0.8246	1	92436	0.2592
4	0.8653	1	93547	0.2800
5	0.5932	1	94586	0.1558
6	0.5081	0	95879	0.1712
7	0.9366	1	43276	0.5813
8	0.9707	1	89606	0.5189
9	0.8723	1	90585	0.2915
10	0.8287	1	91981	0.2819
11	0.9404	1	92573	0.3531
12	0.7578	1	93548	0.2274
13	0.7975	1	94507	0.2327
14	0.6413	1	95513	0.2080
15	0.9351	1	96931	0.3913

Table 3: The results of the evaluation methods on peak scores computed based on four window sizes, i.e., [2200 1500 800 100] of Data_FewDummyCycles

Correlation trace	Average score distance	Significant peak	Peak location	Peak height
0	0.6430	1	97915	0.1579
1	0.7275	1	98886	0.1858
2	0.3843	0	91428	0.1226
3	0.5103	0	101022	0.1654
4	0.4147	0	30352	0.1418
5	0.3066	0	103754	0.1206
6	0.2936	0	242664	0.1309
7	0.5546	0	56889	0.1702
8	0.5130	0	97915	0.1493
9	0.3032	0	225744	0.1344
10	0.6733	1	100026	0.1740
11	0.5690	0	101294	0.1493
12	0.6350	1	102099	0.1552
13	0.4200	0	182481	0.1337
14	0.4610	0	104363	0.1364
15	0.7368	1	105027	0.2041

Table 2 shows that among all 16 correlation traces of Data_NoCountermeasures, only one correlation trace does not have a significant peak. The other correlation traces have a signifiant peak with the average score distance generally higher than 0.80. We also observe that the peak locations in general are around the same point, which is the time point between 90000 and 97000. Based on the results, we conclude that the smart card is not secure.

From Table 3, we observe that most of the significant peaks disappear because of the dummy cycles addition. It also shows that the average score distance of the highest score peaks found in the Data_FewDummyCycles are mostly greater than 0.50 and most of them are not significant. The peak locations now are also not centralized in a certain time point range. This shows us that adding some dummy cycles improve the security of the smart card.

Table 4: The results of the evaluation methods on peak scores computed based on four window sizes, i.e., [2200 1500 800 100] of Data_MoreDummyCycles

Correlation trace	Average score distance	Significant peak	Peak location	Peak height
0	0.3267	0	165057	0.1184
1	0.3334	0	22086	0.1189
2	0.4001	0	225990	0.1358
3	0.4148	0	220618	0.1273
4	0.3594	0	109355	0.1261
5	0.3282	0	74586	0.1371
6	0.5354	1	164374	0.1425
7	0.3141	0	179148	0.1250
8	0.3305	0	59876	0.1160
9	0.3379	0	93696	0.1303
10	0.5304	0	113245	0.1495
11	0.4730	0	96061	0.1437
12	0.4179	0	142481	0.1281
13	0.3600	0	112019	0.1311
14	0.3029	0	106447	0.1282
15	0.4706	0	82370	0.1355

The data Data_MoreDummyCycles was obtainned from the smart card with dummy cycles inserted in every 4 to 20 cycles. This means that the data contains more dummy cycles than the Data_Few DummyCycles. Consistent with this fact, the results on Table 4 show that now only one correlation trace has a significant peak with a rather low average score distance. This shows that, even though this countermeasure setting does not make the smart card completely secure, it is more secure than the other settings.

CONCLUDING REMARKS

We conclude that the Gaussian curve fitting method is able to give scores to each peak found in a correlation trace. The scores represent the original correlation trace. The average score distance is able to represent the peak significance by a number, while the cluster analysis method is able to represent the peak significance by showing to which cluster the highest peak score belongs to. Using the Gaussian curve fitting method, the subjective qualification of the peak significance can be objectified. Thus, better decisions can be taken by security experts. We also conclude that the Gaussian curve fitting method is able to show the influence of peak sizes, especially the width and height, to a significance of a particular peak.

REFERENCES

[1] Mangard, S., Oswald, E., and Popp, T., *Power analysis attack: Revealing the secrets of smart cards*, Springer, 2007.

[2] P.C. Kocher, J. Jaffe, and B. Jun, *Differential power analysis*, proceedings of Crypto 1999, Lecture notes in Computer Science, vol. 1666, pp. 398-412, 1999.

[3] E. Brier, C. Clavier, F. Olivier, *Correlation power analysis with a leakage model*, proceedings of CHES 2004, Lecture notes in Computer Science, vo. 3156, pp. 16-29, 2004.

[4] F.X. Standaert, T.G. Malkin, and M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, Cryptology ePrint Archive, Report 2006/139.

[5] F.X. Standaert, T.G. Malkin, and M. Yung, *A Formal Practice-Oriented Model for the Analysis of Side-Channel Attacks*, Cryptology ePrint Archive, Report 2006/139, http://eprint.iacr.org/.

[6] F.X. Standaert, *A Didactic Classification of some Illustrative Leakage Functions*, in the proceedings of WISSEC 2006, Antwerp, Belgium, 2006.

[7] S.Q. Zhang, et al., *Peak detection with chemical noise removal using short-time FFT for kind of MALDI data*, the First International Symposium on Optimization and Systems Biology, Beijing, China, 2007.

[8] P. Du, W.A. Kibbe, and .S.M. Lin, *Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching*, Bioinformatics Advance Access, 2006.

[9] E. Lange, et al., *High-accuracy peak picking of proteomics data using wavelet techniques*, Pacific Symposium on Biocomputing 11, pp. 243-254, 2006.

[10] M. Dijkstra, et al., *Peak quantification in surface-enhanced laser desorption/ionization by using mixture models*, Proteomics, 2006.

[11] Statgraphics Centurion, *Multivariate Methods*, http://www.statgraphics.com/multivariate_methods.htm