Uncertain? No, It's Very Certain!

Changhai Ou, Zhu Wang, Degang Sun, Xinping Zhou, Juan Ai

Uncertain? No, it's very certain!

Recovering the Key from Guessing Entropy enhanced CPA

Changhai Ou[1,2], Zhu Wang[1,*], Degang Sun[1], Xinping Zhou[1,2], and Juan Ai[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
{ouchanghai, wangzhu, sundegang, zhouxinping, aijuan}@iie.ac.cn

Abstract. It has always been the concern of side channel analysis that how to recover the key with a probability of about 1.00 under the condition that the number of power traces is very small and the success rates is very low. In order to recover the key, the attacker has to try to reduce the guessing entropy to decrease the uncertainty of the key. Unfortunately, guessing entropy is only a evaluation of attack ability in most cases. In this paper, we introduce the statistical characteristics of guessing entropy and propose guessing entropy enhanced CPA (GE-CPA). Its feasibility is verified in theory and experiment. Experiments on both AES algorithm implemented on an AT89S52 single chip and power trace set *secmatv*1 of DES encryption on the side channel attack standard evaluation board(SASEBO) from the website *DPA contest v*1. The experimental results show that, by only repeating the experiments less than 30 times, our GE-CPA can effectively recover the key even under the bad condition that success rate only ranges from 5% to 8%. Thus, the problem is well solved.

Keywords: guessing entropy, CPA, guessing entropy enhanced CPA, GE-CPA, side channel, *DPA contest v*1

1 Introduction

Standaert et al. proposed two evaluation methods success rate and guessing entropy to evaluate the efficiency of side channel attacks [14]. Souissi et al. detailed that, on one hand, the first-order success rate denoted the probability that, given a pool of traces, the attack's best guess was the correct key; On the other hand, the guessing entropy measured the position of the correct key in a list of key hypotheses ranked by a kind of side channel attack[13]. A. Venelli expressed guessing entropy as the average position of the correct hypothesis in the sorted hypothesis vector of an attack [16]. The positions of wrong keys are not taken into consideration.

In short, as well as the concept of entropy firstly proposed by Shannon [12], which denotes the uncertainty of information, guessing entropy is also used to denote the uncertainty of information. The higher the guessing entropy, the greater uncertainty of the key.

1.1 Related Works

There are three main types of applications of the guessing entropy:

Firstly, as well as success rate in [15, 11, 17, 6], guessing entropy is used to evaluate the efficiency of side channel attacks. In other words, guessing entropy is used to evaluate the uncertainty of key [10, 4, 7].

Secondly, as well as entropy, guessing entropy is also used to evaluate the uncertainty of information. The information here is not limited to the key. For example, Michael Backes and Boris Köpf proposed a novel approach for quantifying a system's resistance to unknown message side channel attacks using guessing entropy [2].

Thirdly, guessing entropy is used to improve the efficiency of side channel attacks. Nassar et al. proposed an empirical approach named Rank Corrector (RC) aiming at enhancing most side channel attacks [9]. The main principle of *RC* is to detect and discard the false keys hypotheses when analyzing the ranking evolution. With the increase number of power traces, the correct key will reach the first position. Martin et al. constructed an extremely efficient algorithm that accurately computing the rank of a (known) key in the list of all keys [8]. This approach is tweaked and can be also utilised to enumerate the most likely keys in a parallel fashion.

However, the above improvements only consider that guessing entropy changes with the number of power traces used in each repetition. They don't consider the relationship between the repetitions and guessing entropy. So, very different from [8], we make use of the information leaking from the guessing entropy to recover the key rather than only a kind of evaluation in this paper. Our scheme significantly improves the efficiency of CPA.

1.2 Our Contributions

It is difficult to recover the key using in cryptographic device with a probability of about 1.00 under the condition that the number of power traces is very small and the success rate is very low. To solve the problem, we propose guessing entropy enhanced CPA (GE-CPA) in this paper. Its feasibility is proved in theory and experiment. Experiments on an AT89S52 single chip and the Side Channel Attack Standard Evaluation Board (SASEBO) show that, our scheme can effectively recover the key with a probability of about 1.00 even under the condition that the success rate of traditional CPA is only about 5% to 8%. Thus, the problem is well solved.

1.3 Organization

This paper is organized as follows. The concept of guessing entropy in side channel attack is given in Section 2. In Section 3, we introduce our guessing entropy enhanced CPA (GE-CPA). Then, in Section 4, experiments on AES algorithm implemented on an AT89S52 single chip and power trace set $secmatv1$ on DES algorithm implemented on the side channel attack standard evaluation board

(SASEBO) from the website *DPA contest V*1 [1] are performed to compare our GE-CPA with traditional CPA. Finally, we conclude this paper in Section 5.

2 Guessing entropy

Similar to Shannon entropy, guessing entropy (GE) [14] in side channel attacks also denotes the uncertainty of information. Guessing entropy is defined as the key position in side channel attack. For example, the attacker performs CPA on AES and gets 256 correlation coefficients corresponding to guessing key from 0 to 255. Then, the correlation coefficients are sorted in descending order. The bit length of key corresponding to the same intermediate value is 8. For each 2^8 possible key in each experiment, if the guessing entropy of guessing key $k^{'}\left(0 \leq k^{'} \leq 2^8 - 1\right)$ is in the position ν, the the guessing entropy is equal to ν.

The guessing entropy, which represents the uncertainly of the key used in cryptographic device, is widely used to evaluate the efficiency of side channel attacks.

3 Guessing entropies enhanced CPA

In this section ,we will introduce the statistical characteristics of guessing entropy. Then, we introduce the way to determine the correct key from guessing entropy enhanced CPA (GE-CPA). The attack flow and the special success rate of our GE-CPA are also given in this section.

3.1 The Statistical Characteristic of Guessing Entropy

We have shown the guessing entropy defined by Standaert et al [14] in Section 2. The location i of the correct key s is returned for each experiment. In this paper, We assume that all keys are likely to be the correct one. However, the correct key is the most superior one of all possible keys. Thus, different to [14], in order to select the optimal key, we calculate guessing entropies for all possible keys. Each guessing key returns a guessing entropy for each repetition. We sort the guessing entropies in descending order and assign different scores according to their positions in the guessing vector. For example, small guessing entropies are given higher scores.

Suppose that the bit length of key corresponding to the same intermediate value is ξ. For each of 2^{ξ} possible keys in each repetition, if the guessing entropy of guessing key $k^{'}\left(0 \leq k^{'} \leq 2^{\xi} - 1\right)$ is ν, then we define the score of guessing entropy of $k^{'}$ as

$$W_{k^{'}} = 2^{\xi} + 1 - \nu. \tag{1}$$

3.2 Determine The Correct Key from Guessing Entropies

Standaert et al. also defined the first order success rate in [14]. A guessing key is returned in each experiment. If this key is the correct one, then we say that the attack experiment satisfies the first order success. Similar to their definition, we define a successful CPA [3] if the correct key s satisfies

$$s = \underset{k}{argmax} \{W_k\}. \tag{2}$$

Actually, the equation (2) is usually satisfied when the guessing entropy of the correct key is close to 1. That's to say, if the number of power traces used in our experiment is n, then the probability

$$\lim_{n\to\infty} Pr[s = \underset{k}{argmax} \{W_k\}] = 1. \tag{3}$$

For each successful repetition, the score of guessing entropy of the correct key is greater than these of wrong guessing keys. So, guessing entropy can be used to distinguish the correct key from the wrong ones. This is why guessing entropy can also contribute to the key recovery.

Let n denote the number of power traces used in each repetition. We define a function of scores of guessing entropies corresponding to guessing key $k^{'}$ on η repetitions $(Exp_1, Exp_2, \cdots, Exp_\gamma)$ as

$$\psi\left(\gamma, n, k^{'}\right) = f\left(W_{k'}^{Exp_1}, W_{k'}^{Exp_2}, \cdots, W_{k'}^{Exp_\gamma}\right). \tag{4}$$

Actually, ψ can be any function of the scores of guessing entropies, such as multiplication, addition, etc. In this paper, we only use a very simple function

$$\psi\left(\gamma, n, k^{'}\right) = \frac{1}{\eta}\sum_{i=1}^{\eta} W_{k'}^{Exp_i}. \tag{5}$$

The limit of function ψ is equal to 2^ξ if the guessing key $k^{'}$ is correct. However, if the guessing key is incorrect, the limit of ψ is equal to $\frac{2^\xi+1}{2}$. That's to say, for each repetition i $(1 \le i \le \eta)$, the following formula 6 is satisfied.

$$\lim_{\gamma\to\infty} \psi(\gamma, n, s) > \lim_{\gamma\to\infty} \psi(\gamma, n, \delta) \quad \delta \in \left\{0, 1, \cdots, 2^\xi - 1\right\} \setminus \{s\}. \tag{6}$$

Formula (6) indicates that when the number of power traces we use in each experiment is large enough, the average score of guessing entropies $\lim_{\gamma\to\infty} \psi(\gamma, n, s)$ of correct key s will be the maximum with a limitation of 2^ξ. The limits of scores of guessing entropies of other wrong guessing keys are $\frac{2^\xi+1}{2}$. That is why the correct key falls in the location of largest score. So as to meet the definition of the first order success rate in [14]. Thus, the feasibility of GE-CPA is well explained in theory.

3.3 Attack Flow

In the above subsection 3.2, we show how to recover the key from GE-CPA. Suppose that we randomly encrypt m plaintexts and acquire m power traces. The number of repetitions C is set to 0, the array $W\left[0\cdots 2^{\xi}-1\right]$ saves the scores of guessing entropies for all possible keys from 0 to $2^{\xi}-1$, and k indicates the number of power traces added in each repetition. The steps to recover the key using GE-CPA are as follows:

Step 1: The attacker randomly selects $n(n << m)$ power traces and the corresponding n plaintexts. He then sets the number of repetitions $C = C + 1$.

Step 2: For each guessing key, he calculates the assumed power consumption using power model(i.e. Hamming distance model). Then, he calculates the correlation coefficients between the assumed power consumption and the power traces. All correlation coefficients are returned, which is different from traditional CPA.

Step 3: The attacker sorts the correlation coefficients, determines the score of guessing entropy for each possible key key_i. Then, he adds them to the array of scores of guessing entropies W. Actually, Other functions like multiplication can also be used.

Step 4: The attacker sorts the array W, determines the score of guessing entropy for each key. If the score of guessing entropy of the correct key is still not the maximum in W, then, $n = n + k$. Otherwise, he goes to step 1.

The greater the score of guessing entropy of the correct key, the higher certainty the key is. The correct key can be recovered with a probability of 1.00 when the score of guessing entropy corresponding to the correct key is equal to 256.

3.4 Success Rate in Guessing Entropy Enhanced CPA

The relationship between different repetitions and guessing entropies has not been taken into consideration in traditional side channel attacks. They just judge whether an experiment is successful. Therefore, in the statistics, the success rate will be close to a fixed value (i.e. 0.50), which not grows with the increase number of repetitions. In our GE-CPA, we consider the relationship between repetitions and guessing entropies. When the number of power traces used in each repetition is sufficient to make the score of guessing entropy corresponding to the correct key larger than those of wrong keys, the success rate grows with the increase number of repetitions. Finally, the success rate of GE-CPA is close to 1. Thus, the success rate of GE-CPA is very different from that of traditional side channel attacks.

Actually, the number of power traces in each repetition is far from infinity. The score of guessing entropy of the correct key is just close to a fixed value larger than $\frac{2^{\xi}+1}{2}$ rather than 2^{ξ} after many repetitions if the number of power traces we use in each repetition is relatively small. That is to say, the score of guessing entropy corresponding to the correct key will be larger than other ones

corresponding to wrong guessing keys. In this case, the correct key can also be distinguished from the wrong ones.

Suppose that we randomly select τ power traces from a total number of N and repeat this operation γ times. The average score of guessing entropy of the correct key is μ $\left(\mu > \frac{2^{\xi}+1}{2}\right)$. The limits of scores of guessing entropies corresponding to the wrong guessing keys are $\frac{2^{\xi}+1}{2}$ when $\gamma \to \infty$. Let $T_{suc}^{\gamma,C_N^{\tau}}$ and $T_{unsuc}^{\gamma,C_N^{n}}$ denote the the number of successful and unsuccessful repetitions respectively, then

$$\lim_{\gamma\to\infty} T_{suc}^{\gamma,C_N^{\tau}} = \gamma. \tag{7}$$

Then, the success rate ϕ of our GE-CPA is

$$\phi = \lim_{\gamma\to\infty} \frac{T_{suc}^{\gamma,C_N^{\tau}}}{T_{suc}^{\gamma,C_N^{\tau}} + T_{unsuc}^{\gamma,C_N^{\tau}}} = 1 \tag{8}$$

Actually, the variable γ doesn't need to be infinity, the greater the variable γ is, the closer to μ the average score of guessing entropy of the correct key will be. The score of guessing entropy of the correct key is obviously larger than those of wrong keys after many repetitions. γ is less than 30 when only a small number of power traces with success rate ranging from 5% to 8% are used. Failure will not happen if more repetitions are done. This also demonstrates the high efficiency of our GE-CPA.

4 Experimental Results

4.1 Experiments on AT89S52 single chip

Our first experiment is performed on an AT89S52 single chip. The clock frequency of this chip is 12 MHz. The minimum instructions takes 12 clock cycles for execution. We utilize a $Tektronix\ DPO$ 7254 oscilloscope, and the sampling rate is set to $1GS/s$. The output of S-box in the first round of AES encryption is chosen as the attack point. We test the instruction '$MOVCA, @A + DPTR$', which treats the value of register A as the offset and treats the address $DPTR$ of S-box as the base address, then looks up table S-box and writes the result back to register A (as shown in Fig. 1).

We randomly encrypt 4000 plaintexts and acquire 4000 power traces. Each power trace contains 5000 time samples. We calculate the correlation coefficients between each time sample and Hamming weights of the outputs of S-box. We randomly choose 150 power traces and the corresponding success rate is 93%. We randomly choose power traces from a total number of 150 and repeat this operation for 200 times to calculate the success rate and guessing entropy (as shown in Fig. 2). The success rate ranges from 1% to 61% when $10 \sim 110$ are used. The corresponding guessing entropies range from 115.9 to 1.35 (as shown in Fig. 2(b)). When 40, 60, 80 and 100 power traces are used, the success rates

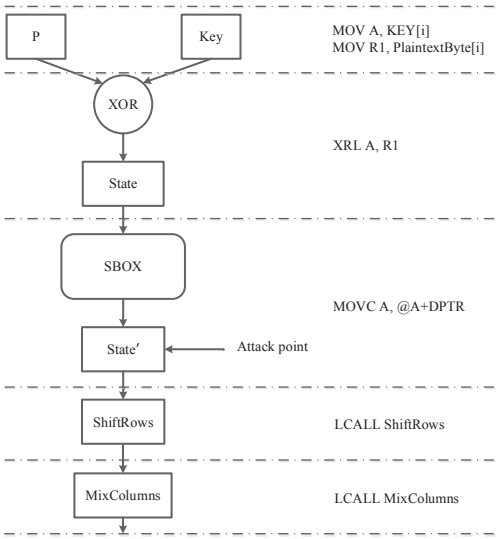

Fig. 1. The output of S-box in the first round of AES algorithm.

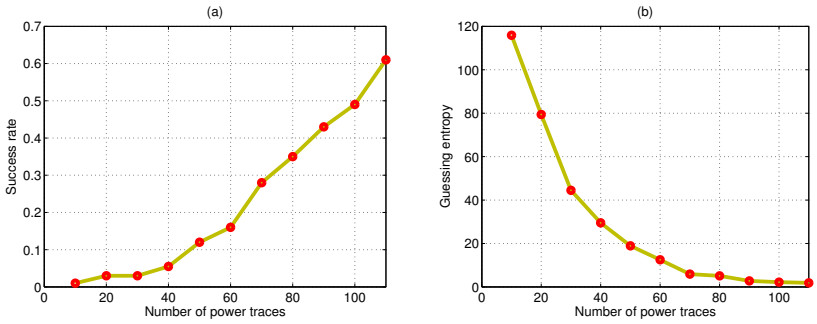

Fig. 2. Success rate (a) and guessing entropy (b) of CPA on AT89S52.

reach 3%, 7.6%, 11% and 16.5%. The corresponding guessing entropies reach 29.5, 12.5, 5.1 and 2.2 respectively.

The experimental results of our GE-CPA are shown in Fig. 3. The key even can be recovered after only 10 repetitions when 40 power traces with a success rate of 5.5% are used. This also indicates that our GE-CPA can significantly improve the efficiency of CPA. The average score of guessing entropy of the correct key is close to 226.5, which is stably larger than these corresponding to wrong keys. When more power traces are used in each repetition, the guessing entropy corresponding to correct key decreases, leading to the increase of the corresponding score of guessing entropy.

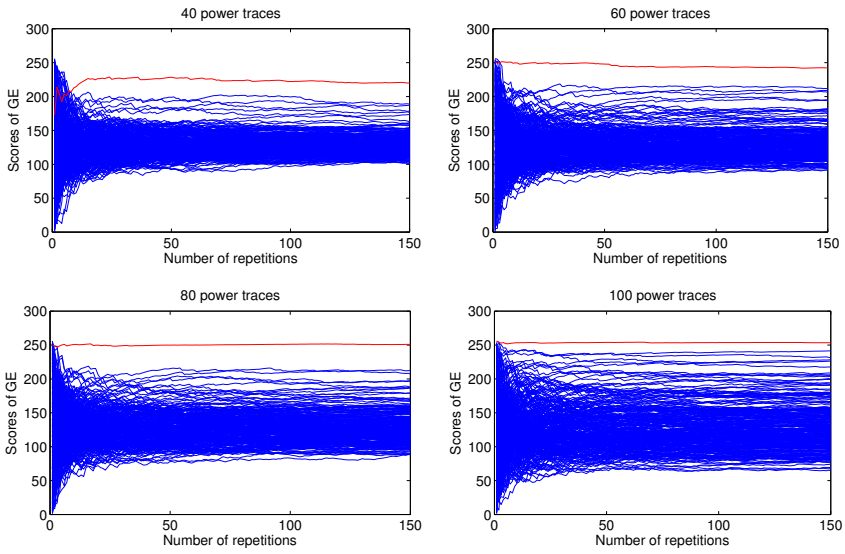

Fig. 3. Score of guessing entropy for each key in different repetitions when 60, 80, 100 and 120 power traces are used in each repetition.

The guessing entropies are 29.5, 12.5, 5.1 and 2.2 when 40, 60, 80 and 100 power traces are used respectively. With more repetitions, the average score of guessing entropy corresponding to the correct key are close to 226.5, 243.5, 250.9 and 253.8 respectively. Other scores of guessing entropies corresponding to wrong keys are relatively smaller.

The experimental results will be better if a small number of power traces are randomly selected from a large power trace set in each repetition. However, this paper tries to solve the problem that how to recover the key with a probability of about 1.00 under the condition that the number of power traces is small, and success rates is low. So, we only consider to use a small number of power

traces in our experiment. For example, we use 150 power traces in total in this experiment. If we randomly select 150 power traces from a total number of 4000 to perform CPA, the success rate is equal to 0.93, which is less than 1.00.

4.2 Experiments on SASEBO

Our second experiment is on DES algorithm implemented on the side channel attack standard evaluation board (SASEBO). We use the power trace set of *DPA contest* $v1$ provided on the website of *DPA contest* [1]. 5000 power traces of power trace set $secmatv1$ are downloaded. We attack the first S-box with 6 bits input and 4 bits output in the last round of the DES algorithm (as shown in Fig. 4). We use the $14000^{th} \sim 16000^{th}$ time samples in our experiments.

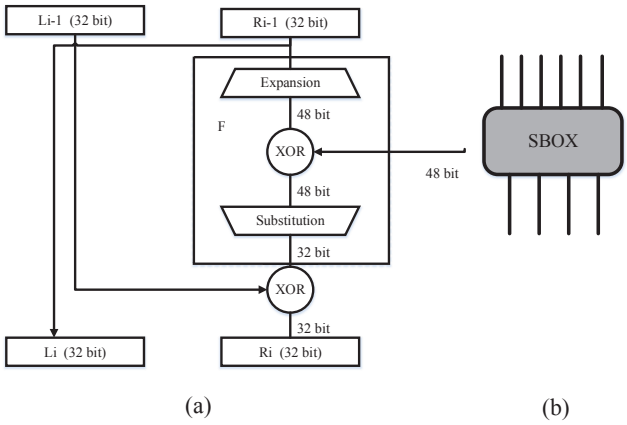

Fig. 4. DES algorithm. (a) shows a round of DES, and (b) shows the S-box of DES.

We randomly select 200 power traces from a total number of 5000 and repeat the operation 200 times. The first order success rate is about 80%. Then, we randomly select power traces from a total number of 200 to perform CPA. The success rates range from 1% to 39% when $10 \sim 140$ power traces are used (as shown in Fig.5(a)). The corresponding guessing entropies range from 33.6 to 6.13 (as shown in Fig. 5(b)). When 40, 60, 80 and 100 power traces are used, the success rates reach 3%, 7.6%, 11% and 16.5% (as shown in Fig.5(a)). The corresponding guessing entropies are 25.56, 19.4, 16.1 and 10.8 respectively (as shown in Fig. 5(b)).

When more power traces are used, the guessing entropy corresponding to the correct key decreases, leading to the increase of the corresponding score (as shown in Fig. 6). The score of guessing entropy corresponding to the correct key

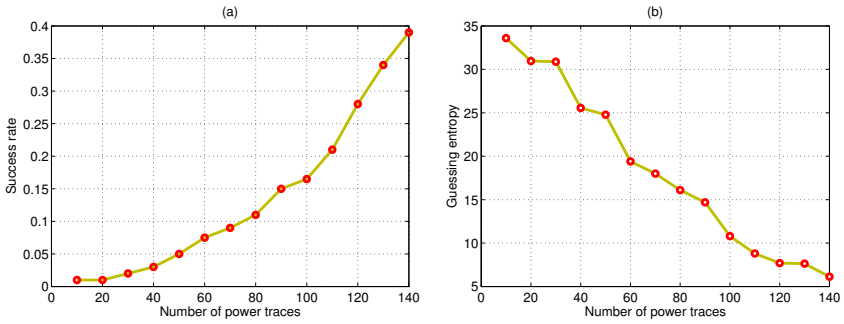

Fig. 5. Success rate (a) and Guessing entropy (b) of CPA on SASEBO.

becomes the largest one after 5 repetitions when average 60,100 and 120 power traces are used.

However, in order to obtain a stable success rate, the experiments with average 60 power traces have better results than that with 80 power traces. This also indicates, the guessing entropies of the initial several experiments are very important, which may affect the starting position of success. The score of guessing entropy of the correct key will be greater if more power traces are used after a few repetitions. However, this doesn't mean the difference of the scores of guessing entropies between the correct key and wrong keys is better. Because of the limited number of repetitions, less power traces used in each repetition may get greater difference.

When 80 power traces are randomly selected, the score of guessing entropy of the correct key becomes stable in different repetitions (as shown in Fig. 7). Sometimes, the experiment has a good beginning (as shown in Fig. 7(a)), the score of guessing entropy of the correct key becomes the maximum after $1 \sim 6$ repetitions. However, it gradually reduce after 7 repetitions. The change is stable after 26 repetitions. The score of guessing entropy of the correct key is sometimes small at the beginning of our experiments. However, it becomes the maximum after many repetitions (as shown in Fig. 7(b)). Regardless of the beginning, the correct key can finally be distinguished from the wrong ones.

It is worth noting that we use a total number of 200 power traces to compare the difference of guessing entropies between the correct key and wrong keys. Actually, when 60 power traces are used in each repetition, our GE-CPA can recover the key (As shown in Fig. 6). Some power traces may be repeatedly selected if the total number of power traces is small and the number of repetitions is large. The effectiveness may be a little worse than randomly selecting power traces from a large power trace set. In addition, the quality of this power trace set may affect the experimental results. However, the experimental results are almost the same if we randomly select 60 power traces from a total number of 120 or 200.

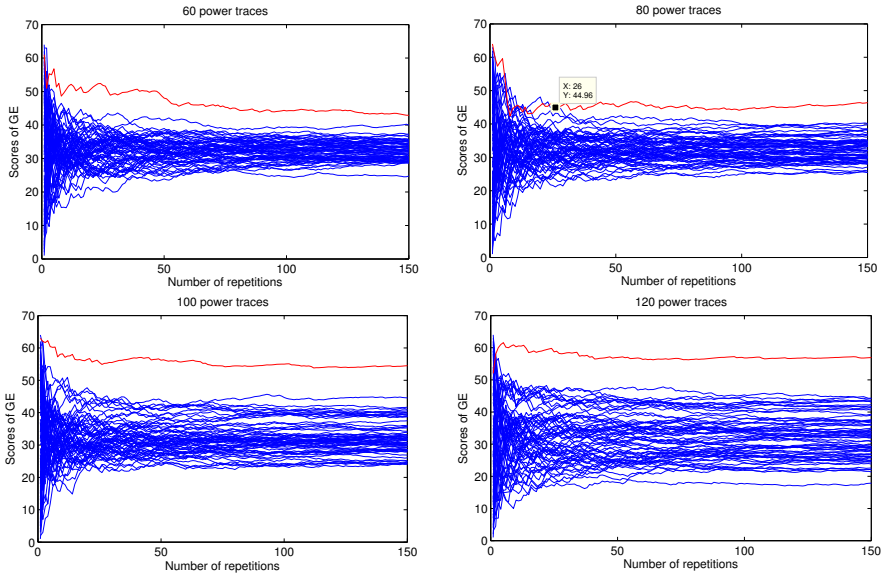

Fig. 6. Scores of guessing entropies for each key in different repetitions when 60, 80, 100 and 120 power traces are used in each repetition.

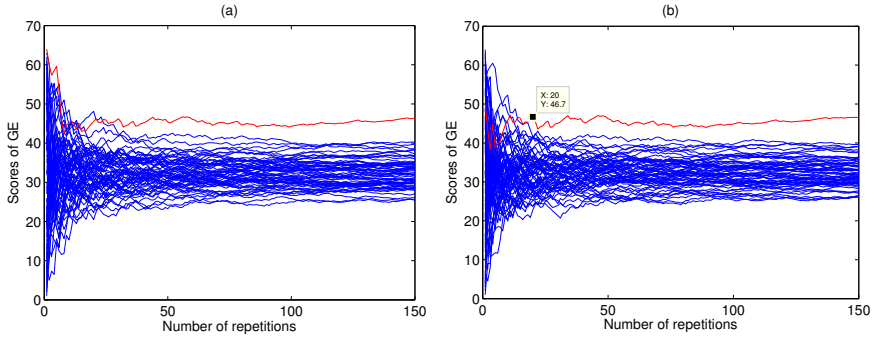

Fig. 7. Scores of guessing entropies for each key in different repetitions when 80 power traces are randomly selected in each repetition.

Komano et al. proposed built-in determined sub-key correlation power analysis, 65 power traces are used to recover the key. Which is more efficiency than our GE-CPA [5]. We also simply compare the efficiency of our guessing entropy ehanced CPA (GE-CPA) with other enhanced side CPA attacks shown on the web site of DPA contest v1 [1]. Hideo used 107 power traces by using his advanced BS-CPA. Yongdae used 119 power traces by using his "CPA with chosen order". Daisuke used 120 power traces by using his "Dual round attack by BS-CPA using improved power model". The efficiency of those attacks are similar to our GE-CPA. Benedikt used 329 power traces by using the difference of means on the last round of DES. Victor used 322 power traces by using his "CPA on the 16th round of the DES selecting the good temporal window". Other attacks using more power traces are not deteiledly introduced here. Our GE-CPA is more efficiency than these attacks.

5 Conclusion

As a common evaluation to evaluate the effectiveness of side channel attacks, guessing entropy is used to measure the uncertainty of the key. In this paper, we analyze the statistical characteristics of guessing entropy and propose GE-CPA. Experiments on AES algorithm implemented on an AT89S52 single chip and power trace set $secmatv1$ of DES algorithm implemented on the side channel attack standard evaluation board (SASEBO) from the website $DPA\ contest\ v1$ show that our scheme can efficiently recover key. Our scheme can significantly improve the effectiveness of CPA.

Actually, CPA is just a common type of side channel analysis. Guessing entropy can enhance many other types of side channel analysis like DPA and template attack, etc. This also indicates the practicability of our scheme.

Acknowledgment. This research is supported by the Nation Natural Science Foundation of China (No.61372062). The authors would like to thank Prof Kerry-Lynn Thomson from Nelson Mandela Metropolitan University and the anonymous referees of IFIP SEC 2016 for the suggestions to improve this paper.

References

1. Dpa contest. http://www.dpacontest.org/home/.
2. M. Backes and B. Köpf. Formally bounding the side-channel leakage in unknown-message attacks. In *Computer Security-ESORICS 2008*, pages 517–532. Springer, 2008.
3. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 16–29, 2004.
4. Y. Fei, Q. Luo, and A. A. Ding. A statistical model for dpa with novel algorithmic confusion analysis. In *Cryptographic Hardware and Embedded Systems–CHES 2012*, pages 233–250. Springer, 2012.

5. Y. Komano, H. Shimizu, and S. Kawamura. Bs-cpa: Built-in determined subkey correlation power analysis. *Ieice Transactions on Fundamentals of Electronics Communications & Computer Sciences*, 93(9):1632–1638, 2010.
6. V. Lomné, E. Prouff, M. Rivain, T. Roche, and A. Thillard. How to estimate the success rate of higher-order side-channel attacks. In *Cryptographic Hardware and Embedded Systems–CHES 2014*, pages 35–54. Springer, 2014.
7. Q. Luo and Y. Fei. Algorithmic collision analysis for evaluating cryptographic systems and side-channel attacks. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 75–80. IEEE, 2011.
8. L. Mather, E. Oswald, and C. Whitnall. Multi-target dpa attacks: Pushing dpa beyond the limits of a desktop computer. In *Advances in Cryptology–ASIACRYPT 2014*, pages 243–261. Springer, 2014.
9. M. Nassar, Y. Souissi, S. Guilley, and J.-L. Danger. rank correction: A new side-channel approach for secret key recovery. In *Security Aspects in Information Technology*, pages 128–143. Springer, 2011.
10. M. Nassar, Y. Souissi, S. Guilley, and J.-L. Danger. Rsm: a small and fast countermeasure for aes, secure against 1st and 2nd-order zero-offset scas. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pages 1173–1178. IEEE, 2012.
11. M. Rivain. On the exact success rate of side channel analysis in the gaussian model. In *Selected Areas in Cryptography*, pages 165–183. Springer, 2009.
12. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
13. Y. Souissi, M. Nassar, S. Guilley, J.-L. Danger, and F. Flament. First principal components analysis: a new side channel distinguisher. In *Information Security and Cryptology-ICISC 2010*, pages 407–419. Springer, 2011.
14. F. Standaert, T. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques,Cologne, Germany, April 26-30, 2009. Proceedings*, pages 443–461, 2009.
15. F.-X. Standaert, B. Gierlichs, and I. Verbauwhede. Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected cmos devices. In *Information Security and Cryptology–ICISC 2008*, pages 253–267. Springer, 2009.
16. A. Venelli. Efficient entropy estimation for mutual information analysis using b-splines. In *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, pages 17–30. Springer, 2010.
17. N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F.-X. Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In *Selected Areas in Cryptography*, pages 390–406. Springer, 2013.