Security Evaluation of a DPA-Resistant S-Box Based on the Fourier Transform

Yang Li[1], Kazuo Sakiyama[1], Shinichi Kawamura[2], Yuichi Komano[2], and Kazuo Ohta[1]

[1] The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
{liyang,saki,ota}@ice.uec.ac.jp
[2] Toshiba Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
{shinichi2.kawamura,yuichi1.komano}@toshiba.co.jp

Abstract. At CHES 2006, Prouff *et al.* proposed a novel S-box calculation based on the discrete Fourier transform as a first-order DPA countermeasure. At CHES 2008, Coron *et al.* showed that the original countermeasure can be broken by first-order DPA due to a biased mask and they proposed an improved algorithm. This paper shows that there is still a flaw in the Coron's S-box algorithm with respect to a practical software implementation. We pre-process the power traces to separate them into two subgroups, each has a biased mask. For the separated power traces, we propose two post analysis methods to identify the key. One is based on CPA attack against one subgroup, and the other is utilizing the difference of means for two subgroups and a pattern matching. Finally, we compare these two attack methods and propose an algorithm level countermeasure to enhance the security of Coron's S-box.

Keywords: Side channel attacks, Masking, Fourier transform, S-box, Probability density function.

1 Introduction

Side Channel Attacks (SCAs) which expose secret information using side channel leakages gained from the physical implementations of cryptosystem was first introduced by Kocher in 1996 [7]. Since then, various SCAs based on different leakage sources or different techniques are proposed. Among them, Differential Power Analysis (DPA) has been demonstrated to be very powerful. On the other hand, many DPA countermeasures were proposed both at algorithm level [2,4,6] and at logic level [13,11]. The goal of algorithm level countermeasure is masking every sensitive variable to suppress the dependence between the side channel leakage and the secret information. While the logic level countermeasures trend to improve the hardware gates to diminish the DPA leakage at the source.

At CHES 2006, Discrete Fourier Transform (DFT) was introduced by Prouff *et al.* as a general technique achieving immunity against first order DPA [8]. The proposed AES S-box algorithm based on DFT is referred as *Prouff's S-box* in this paper. Two

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 3–16, 2009.

years later, Coron *et al.* shows that Prouff's S-box is vulnerable to first-order DPA due to a biased mask in their algorithm [3]. They also propose an improved version of S-box based on the DFT(referred as *Coron's S-box* in this paper). In the Coron's S-box, it is proved that all the intermediate variables are masked by random numbers and are uniformly distributed.

In this paper, we show that Coron's S-box still has a flaw in terms of power analysis in a practical software implementation. In Coron's S-box, according to the value of the input, the output of the S-box is calculated as a combination of pre-calculated parameters. A random mask bit is used switching this calculation into two power profiles randomly. Focusing on a practical software implementation of Coron's S-box, we show that different values of this mask bit can be distinguished from the power consumption at certain points of the algorithm. Our pre-process analysis method can separate traces into two subgroups with biased mask, and it is similar to the power analysis using *Probability Density Function* (PDF) proposed by Schaumont and Tiri [12,10]. After grouping the power traces, two post-processing analysis methods are explained and compared. One is first order DPA attack against one subgroup, and the other one is calculating the Difference of Means (DoMs) for two subgroups and matching the peaks with pre-calculated patterns.

2 Background Information

2.1 Masking Countermeasures against First Order DPA

Both [8] and [3] explain that if all the *intermediate variables* during the calculation are independent of any *sensitive variable*, the implementation is immune to first order DPA. The intermediate variables refer to the variables manipulated during the calculation, and the sensitive variables can be calculated by a key guess and public variables (*i.e.* plaintext or ciphertext). In the masking countermeasures against first order DPA, it is necessary all the intermediate variables are masked by uniformly distributed random variables to be independent of the sensitive variables.

2.2 Coron's S-Box Calculation Based on DFT

In order to explain the concept of Coron's S-box briefly, we follow the notations used in [3]. The DFT function $\widehat{F}$ for the original function F is defined with Z=$(Z_{n-1}, \cdots, Z_0)$ $\in \mathbb{F}_2^n$ by

$$\widehat{F}(Z) = \sum_{i \in \mathbb{F}_2^n} F(i)(-1)^{i \cdot Z}, \tag{1}$$

where $i = (i_{n-1}, \cdots, i_0) \in \mathbb{F}_2^n$ and "$\cdot$" denotes the scalar product calculated by

$$i \cdot Z = \bigoplus_{j=0}^{n-1} i_j Z_j. \tag{2}$$

If we perform the DFT operation against $\widehat{F}$ again, we have $\widehat{\widehat{F}} = 2^n F$ as

$$F(Z) = \frac{1}{2^n} \sum_{i \in \mathbb{F}_2^n} \widehat{F}(i)(-1)^{i \cdot Z}. \tag{3}$$

In order to mask all the intermediate variables in calculating Eq. (3), three n-bit random numbers R_1, R_3 and $R_4 \in \mathbb{F}_2^n$ and one-bit random number $R_2 \in \mathbb{F}_2$ are necessary. For the input $(\widetilde{Z} = Z \oplus R_1, R_1)$, the 3-tuple output $\left((-1)^{R_2}F(Z) + R_3, R_2, R_3\right)$ is calculated by

$$(-1)^{R_2}F(Z) + R_3 \bmod 2^n = \frac{1}{2^n}\left(R' + \sum_{i \in \mathbb{F}_2^n} \widehat{F}(i)(-1)^{R_2 \oplus (i \cdot \widetilde{Z}) \oplus (i \cdot R_1)}\right), \quad (4)$$

where $R' = 2^n R_3 + R_4$.

We denote the scalar product function $X, Y \mapsto X \cdot Y$ by SP and the function $X, T \mapsto \widehat{F}(X)(-1)^T$ by SFT. Then, the calculation used for Coron's S-box can be described as shown in Alg. 1. The operation $\boxplus$ denotes addition modulo 2^{2n}.

Algorithm 1. Calculation Steps for Coron's S-box [3]

INPUTS: A masked value $\widetilde{Z} = Z \oplus R_1$ and the mask R_1

OUTPUT: The 3-tuple output $\left(\left((-1)^{R_2}F(Z) + R_3\right) \bmod 2^n, R_3, R_2\right)$

1. Generate a random bit R_2
2. Generate two n-bit random R_3 and R_4
3. $result \leftarrow 2^n R_3 + R_4$
4. **for i from** 0 **to** $2^n - 1$ **do**
5. $\quad T_1 \leftarrow SP(i, \widetilde{Z})$ $\quad [T_1 = i \cdot \widetilde{Z}]$
6. $\quad T_1 \leftarrow T_1 \oplus R_2$ $\quad [T_1 = R_2 \oplus i \cdot \widetilde{Z}]$
7. $\quad T_2 \leftarrow SP(i, R_1)$ $\quad [T_2 = i \cdot R_1]$
8. $\quad T_1 \leftarrow T_1 \oplus T_2$ $\quad [T_1 = R_2 \oplus i \cdot Z]$
9. $\quad T_1 \leftarrow SFT(i, T_1)$ $\quad [T_1 = \widehat{F}(i)(-1)^{R_2 \oplus i \cdot Z}]$
10. $\quad result \leftarrow result \boxplus T_1$
11. **end** $\quad [result = (2^n R_3 + R_4) \boxplus \sum_{i=0}^{2^n - 1} \widehat{F}(i)(-1)^{R_2 \oplus i \cdot Z}]$
12. $result = result \gg n$ $\quad [result = \left((-1)^{R_2}F(Z) + R_3\right) \bmod 2^n]$
13. return $(result, R_3, R_2)$

3 The Flaw of Coron's S-Box and Attack Principle

3.1 Target Steps in the Coron's S-Box

Coron *et al.*'s paper proved that all the intermediate variables manipulated in Alg. 1 are well masked. However, there still exist several target steps in Alg. 1 even first-order DPA does not work directly on Coron's S-box. We review several steps of Coron's S-box considering a practical software implementation as follows.

The operation for $\widehat{F}(i), i \in \{0, 1, \ldots, N\}$, is decided by $(-1)^{R_2 \oplus i \cdot \widetilde{Z} \oplus i \cdot R_1}$ in Eq. (4), which can be further transformed as

$$\begin{aligned}(-1)^{R_2 \oplus i \cdot \widetilde{Z} \oplus i \cdot R_1} &= (-1)^{R_2 \oplus i \cdot (Z \oplus R_1) \oplus i \cdot R_1} \\ &= (-1)^{R_2 \oplus i \cdot Z \oplus i \cdot R_1 \oplus i \cdot R_1} \\ &= (-1)^{R_2}(-1)^{i \cdot Z}. \end{aligned} \quad (5)$$

We notice R_2 is a one-bit mask (*i.e.* $R_2 \in \{0, 1\}$) which affects steps 8, 9 and 10 of Alg. 1 in every loop, where steps 9 and 10 perform operations as

$$result \leftarrow result + \widehat{F}(i)(-1)^{R_2}(-1)^{i \cdot Z}. \tag{6}$$

In Eq. (6) the value of $(-1)^{R_2}(-1)^{i \cdot Z}$ at step 9 directly decides the operation (addition or subtraction) at step 10. Here, we assume that addition with $-\widehat{F}(0)$ is equivalent to subtraction with $\widehat{F}(0)$ in terms of power consumption. And we denote the addition and subtraction operations by $+operation$ and $-operation$, respectively. As the $-$ operation involves an additional complement transform compared with the $+$ operation in a practical micro-processor, there may exists some difference between the $+$ and $-$ operations in power consumption. Here we denote steps 9 and 10 as the *target steps*.

By way of experiment, we measured the power consumption when executing an experimental C code that operates only steps 8, 9 and 10 using fixed values of $R_2 \oplus i \cdot Z$ and $\widehat{F}(0)$. As shown in Fig. 1, an obvious power difference was observed between the cases of the $+$ and $-$ operations that correspond to $R_2 \oplus i \cdot Z = 0$ and 1, respectively.

3.2 Experiment Setup and Overview of Our Attacks

We implemented Coron's S-box over a composite field $\mathbb{F}_{2^4}^2$ based on Alg. 1 on the SASEBO-G (Side-channel Attack Standard Evaluation Board, type-G) [9,5]. The SASEBO-G is designed to develop evaluation schemes against physical attacks with FPGAs. We use a 32-bit reconfigurable CPU called Microblaze on a Xilinx FPGA (Vertex-II pro, xc2vp30) for the experiment. Coron's S-box is written in C code and 1-MHz clock is used for executing the compiled code. The power consumption of the CPU is obtained by measuring the voltage drop of a resistor inserted between FPGA's GND pins and the ground of the board. As shown in Fig. 2, where the 8-bit plaintext, the 8-bit secret key and the unmasked input of the S-box are denoted as P, K and a, respectively. And Z is unmasked input of Alg. 1.

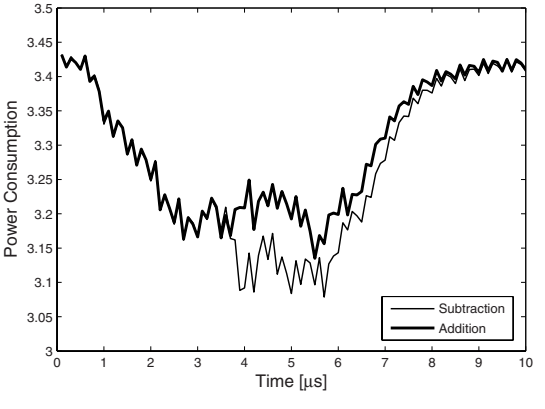

Fig. 1. Averaged power traces for steps 8, 9 and 10 using fixed values of $R_2 \oplus i \cdot Z$ and $\widehat{F}(0)$

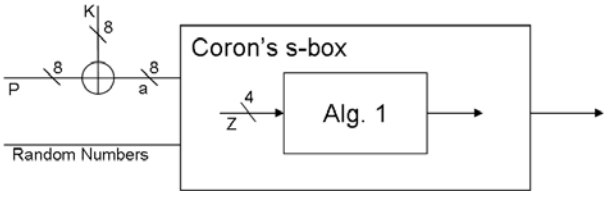

Fig. 2. The attacked S-box model

P is randomly chosen and all random variables are set to be uniformly distributed. The S-box calculation is repeated for 4000 times and the averaged power traces are shown in Fig. 3.

In Fig. 3, sixteen patterns can be clearly recognized corresponding to the 16-time loop of Alg. 1 and furthermore the beginning and the ending of every loop can be roughly guessed. Although the difference of means corresponding to $+$ and $-$ operations can be distinguished, but it only relates to the masked intermediate values, so first order DPA doesn't work. However the mask that masking target steps are used sixteen times in Coron's S-box, attackers can uses multiple points of power trace to reveal the secret information.

In our attacks, the goal of our pre-processing for power traces is separating the power traces into two subgroups with regards to the biased values of R_2. This separation can be achieved by looking into the probability density function of the certain segment of the power traces and details will be explained later. After separation, this paper proposes two post-processing methods to identify the secret key. One is perform traditional DPA to one subgroup. The other method is calculation of the Difference of Means (DoM) for two subgroups and match the peaks pattern and operation pattern to reveal the secret information.

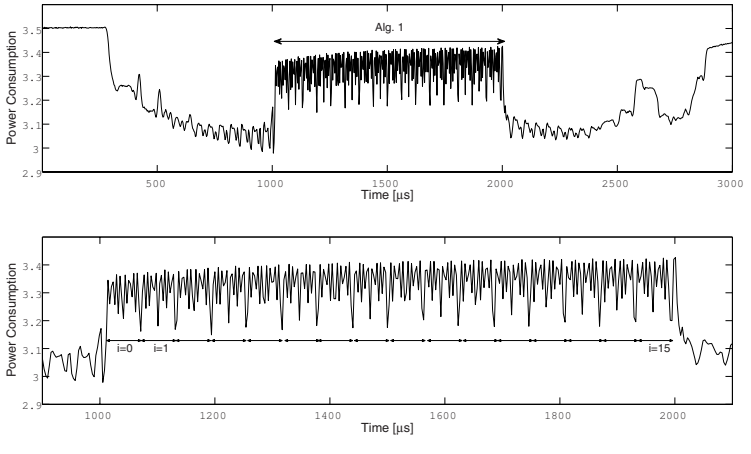

Fig. 3. The upper figure plots the averaged trace for a software implementation of Coron's S-box. The lower figure shows the magnified average traces focusing on the calculation of Alg. 1.

4 Pre-processing

The PDF scheme introduced by [12,10] involves using the histogram to specify the unbalanced influences caused by different values of a random mask bit. Then after a simple filtering and grouping of the power traces, every single subgroup should be vulnerable to traditional power analysis again. However it is normally difficult to specify the correct time when the target steps is performed. In the case of Coron's S-box, we would like to see the PDF histogram at the moment of the $+$ or $-$ operation is performed. For the purpose of identifying the correct attacking point, we divide the end of the first loop into several parts along the time axis and plot the histogram for every part as shown in Fig. 4. That is, we propose a new power analysis called *PDF scanning method*, in which an attacker scans the power traces along the time axis and makes several plots for the PDF. In this way, we can seek out the exact position where $+$ or $-$ operation is performed by checking the shape of each plot for the PDF. In fact, as shown in Fig. 4, we could find a special PDF plot at the time part 8. The magnified figure is shown in Fig. 5.

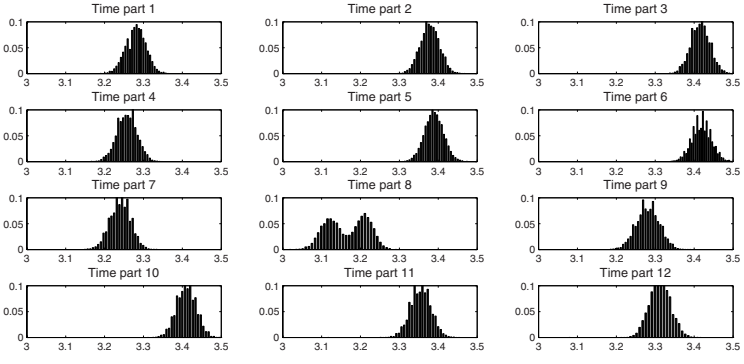

Fig. 4. Results of the PDF scanning method applying to around the end of the first loop of Alg. 1

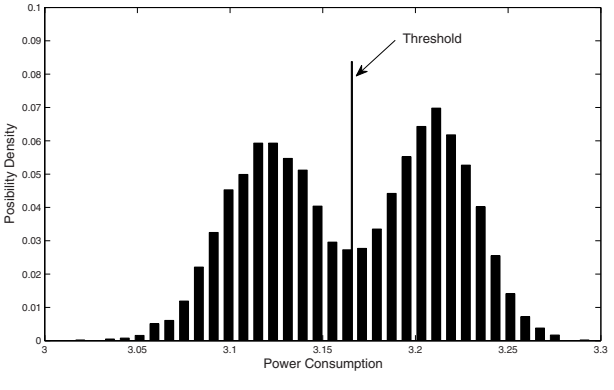

Fig. 5. PDF with two peaks

In Fig. 5, there are two peaks around the average power of 3.12 and 3.21. It is considered that two different distributions overlap each other and form those two peaks in the PDF figures as also discussed in [10]. By setting a threshold around the average power of 3.17, we separate the power trace into two subgroups. Notice that each subgroup still contains the power traces influenced by both the $+$ and $-$ operations because the threshold cannot separate two distributions perfectly. However, one subgroup contains the power traces where more $+$ operations are executed than that the $-$ operations, and vice versa for the other subgroup. Some discussion about the biased rate of R_2 in each subgroup is shown in Appendix A.

5 Two Post Analysis Methods

Our attack model used in this paper is similar to the one mentioned in [3] as

$$Z = \phi(P, K), \tag{7}$$

where ϕ is a Boolean function. The unmasked sensitive input of Alg. 1 Z is determined by value of plaintext P and the secret key K. Denote a key guess by $K^\star$ and the sensitive variable according to this $K^\star$ is denoted by Z_k.

5.1 CPA Attack against One Subgroup

Notice that when one of the inputs for the scale product operation is zero, the result becomes zero. Accordingly, when the first for-loop in Alg.1 is executed (*i.e.* $i = 0$ and $i \cdot Z = 0$), the value of $(-1)^{R_2 \oplus i \cdot Z} = (-1)^{R_2}$ is decided only by R_2. When it is the first for-loop in Alg. 1 where the PDF separation is performed, even if P is randomly selected, the biased $+$ operation or $-$ operation in every subgroup directly corresponds to a biased R_2. Therefore as a kind of post analysis, we can apply an improved version of traditional DPA called (CPA [1]), where the correlation coefficient is applied to determine the linear relationship between data. In the case of an implementation without masking countermeasures, the correct key guess produces a recognizable higher correlation coefficient between the vector of power traces and Z_k at certain moments. When the masking countermeasure is used but a mask is biased, we can still obtain similar results although the correlation coefficient will become lower.

For the case of our software implementation of Coron's S-box, we need two attacking points; one is a point for separating the power traces (denoted by *separation point*) and the other is for CPA attacks (denoted by *CPA point*). As mentioned previously we can find the separation point by using the PDF scanning method at the first loop ($i = 0$). Then, for the biased power traces, one of the remaining fifteen loops can be chosen as a candidate for the CPA point. Here, we choose the second loop ($i = 1$) as the CPA point.

When we used 10 000 power traces in separating groups by the PDF scanning method, the correct key can be distinguished from others with only about 200 traces. In other words, a good separation by the PDF scanning method leads to a small number of traces for CPA. However, the smaller the number of measurements become, the worse separation would be obtained. Therefore, we would anticipate that an optimal number of measurements exist between 200 and 10 000 in our experiment.

5.2 Calculation of DoMs for Two Subgroups and Pattern Matching

Before introducing another post analysis method, we review the Fourier transform used in Alg. 1 again. The basic formula of the S-box calculation based on the Fourier transform is described in Eq. (3). Denoting $2^n - 1$ by N, Eq. (3) can be expressed using the Hadamard matrix as

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N) \end{bmatrix} = \frac{1}{2^n} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots \\ 1 & -1 & 1 & -1 & \dots \\ 1 & 1 & -1 & -1 & \dots \\ 1 & -1 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ (-1)^{0 \cdot N} & (-1)^{1 \cdot N} & (-1)^{2 \cdot N} & (-1)^{3 \cdot N} & \dots \end{bmatrix} \begin{bmatrix} \widehat{F}(0) \\ \widehat{F}(1) \\ \widehat{F}(2) \\ \widehat{F}(3) \\ \vdots \\ \widehat{F}(N) \end{bmatrix}. \tag{8}$$

As the Alg. 1 is a 16-time loop, the 1-bit R_2 actually switches the practical calculation into two power profiles with regards to the $+$ operation or $-$ operation at sixteen positions. As long as the PDF separation at one position generates two subgroups with biased R_2, the bias of $+$ operation or $-$ operation preserves for the rest fifteen target positions. So after calculating the Difference of Means for two subgroups, 16 peaks should appear. The important thing is that according to Eq. (2) the pattern of polarities of these peaks should directly correspond to the value of Z.

Recover the Unmasked Input of Coron's S-box: As a new post analysis method, the input is fixed and 2000 power traces are used for the PDF separation. After that, we calculate the average traces from each subgroup, and the difference of them is calculated and plotted as shown in Fig. 6.

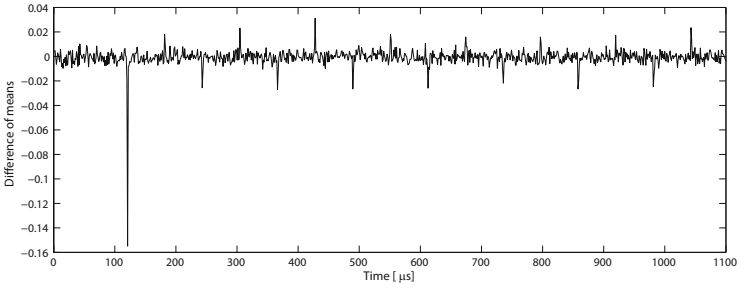

Fig. 6. Difference of mean traces for two groups ($Z = 1$)

As expected, Fig. 6 shows a trace with sixteen peaks with positive or negative directions. According to the Eq. (2), the pattern of 16 peaks starts from time 110 in Fig. 6 matches the operation pattern for $Z = 1$. Therefore the sensitive unmasked input of Alg. 1 Z is revealed by calculation of DoMs and the pattern marching.

Key Recovery from Unmasked Input: Real attackers can choose the first round of the AES as the target to identify the secret key. As shown in Fig. 2, we have $a = P \oplus K$.

As P is understandable and changeable by attackers, the recovery of a is equivalent to the recovery of K.

The unmasked input of S-box a is an 8-bit unknown variable, while the unmasked input of the Alg. 1 Z is a 4-bit variable. So given the value of Z, obviously the value of a cannot be identified. The relation between a and Z is determined according to the irreducible polynomial used in the combinational logic. By checking 256 possible values of a in our implementation, we obtain the relationship between Z and a as shown in Table 1.

From Table 1, we find that normally when Z is known, there are seventeen possible values of a. However there is a special one-to-one corresponding pair that in the case of $Z = 0$, $a = 0$. As a result, the 8-bit key can be successfully identified when the attacker obtain the DoM figure as shown in Fig. 7. Figure 7 with sixteen peaks at the

Table 1. The relationship between Z and possible unmasked input of S-box a in our implementation

Z	Possible unmasked input of S-box./a (number of possible values).
0	0 (1)
1	1, 8, 29, 47, 53, 54, 57, 64, 74, 99, 102, 171, 179, 194, 211, 232, 239 (17)
2	9, 17, 44, 72, 76, 86, 92, 118, 123, 132, 134, 136, 157, 214, 233, 234, 245 (17)
3	3, 24, 35, 39, 42, 75, 90, 93, 95, 110, 113, 165, 170, 192, 206, 222, 230 (17)
4	25, 26, 28, 60, 62, 65, 87, 138, 142, 153, 164, 200, 208, 218, 224, 235, 251 (17)
5	5, 40, 49, 73, 91, 101, 105, 121, 126, 147, 178, 221, 225, 229, 231, 238, 244 (17)
6	4, 27, 32, 37, 51, 97, 116, 131, 141, 145, 151, 154, 188, 212, 216, 228, 250 (17)
7	2, 16, 58, 77, 94, 106, 108, 114, 125, 128, 148, 159, 189, 197, 198, 203, 204 (17)
8	10, 61, 80, 98, 127, 146, 161, 182, 199, 202, 209, 210, 213, 217, 242, 243, 252 (17)
9	6, 48, 70, 78, 79, 81, 84, 135, 150, 155, 167, 180, 186, 190, 215, 220, 226 (17)
10	12, 21, 45, 55, 85, 96, 103, 111, 115, 140, 156, 158, 162, 163, 168, 181, 223 (17)
11	13, 14, 30, 31, 69, 71, 82, 100, 104, 109, 112, 129, 166, 173, 193, 240, 248 (17)
12	22, 36, 38, 43, 46, 59, 66, 67, 68, 107, 117, 133, 137, 176, 195, 247, 249 (17)
13	20, 63, 89, 119, 122, 143, 149, 160, 169, 177, 185, 191, 196, 227, 253, 254, 255 (17)
14	11, 18, 19, 23, 33, 34, 88, 144, 152, 172, 183, 184, 201, 207, 236, 241, 246 (17)
15	7, 15, 41, 50, 52, 56, 83, 120, 124, 130, 139, 174, 175, 187, 205, 219, 237 (17)

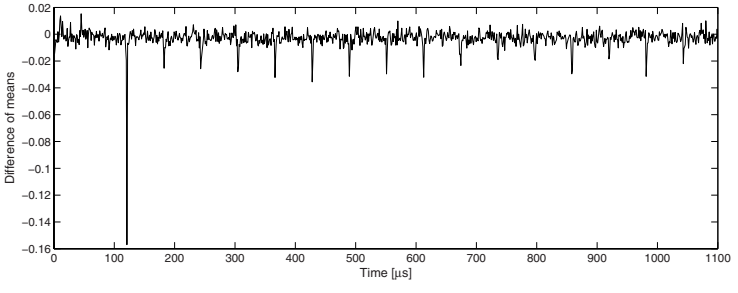

Fig. 7. Difference of mean traces for two groups ($Z = 0$)

same direction corresponds to $Z = 0$. Then according to the one-to-one correspondence in Table 1, $a = 0$. The corresponding 8-bit key piece is the bitwise inversion of the corresponding P. On the other hand, notice that when attackers get a pair of P_1 and P_2 corresponding two different groups of possible values of a, the attacker can restraint the key space by using the relationship as:

$$a_1 \oplus a_2 = (P_1 \oplus K) \oplus (P_1 \oplus K) = P_1 \oplus P_2. \tag{9}$$

We checked our implementation with Table 1, with only a pair of P_1 and P_2 that lead us to two groups of a, at most two key candidates are left. In other words, at most 3 different values of Z are needed to identify the key.

6 Comparison of Two Attack Methods

The common parts of those two attack methods (referred as PDF+CPA attack and PDF+DoMs attack) are the same target steps and the use of PDF separation. The major differences between them are listed as follows:

- In the PDF+CPA attack, after the PDF separation traditional CPA is operated (a comparison between PDF+CPA and Second-Order DPA is given in Appendix B), while in the PDF+DoMs attack, only DoMs is calculated and pattern matching is performed to reveal the secret information.
- In the PDF+CPA attack, for the subsequent CPA attack, P is randomly chosen, while in the PDF+DoMs attack, a fixed P is used 2000 times to identify the value of corresponding Z. PDF+DoMs attack needs fewer power traces, however attackers need more details of the implementation.
- In the PDF+CPA attack, the power consumption for the first two loops of Alg. 1 is enough for a successful attack, while in the PDF+DoMs attack, the power traces containing the entire 16 loops in Alg. 1 are necessary. Since we can have a better resolution for the power traces in PDF+CPA attack, we have a better PDF separation rate compared with that in the PDF+DoMs attack.
- Only the PDF at the first loop ($i = 0$ in Alg. 1) is meaningful in the PDF+CPA attack, while the PDF separation can be applied to any loop ($i = 0 \sim 15$ in Alg. 1) in the case of the PDF+DoMs attack.

7 Possible Countermeasures for the PDF+CPA Attack

We propose a possible countermeasure to enhance the security of Coron's S-box. Considering that the loop index i is also a sensitive variable, we try to mask i with a random value as a countermeasure. For instance, we introduce a look-up table $q[l]$ $(0 \leq l \leq 15)$ where each entry has a different integer randomly chosen from $0, 1, \cdots 15$ to randomize the loop index in the Coron's S-box algorithm (see steps 2a and 4a in Alg. 2 in Appendix C). This countermeasure can prevent the PDF+CPA attack completely, however, in the case of $Z = 0$ the randomization of i cannot disorder the $+$ and $-$ operations. As a result, the peaks of DoMs still exists when $Z = 0$.

8 Conclusions and Future Work

This paper reviews the algorithm of the S-box calculation based on the DFT that is introduced as a first order DPA countermeasure [3]. Based on a practical software implementation, this paper presents a flaw of this s-box algorithm due to the difference of the power consumptions between different operations. By analyzing the power consumption related to this flaw, we use PDF of the power traces to separate them into two subgroups with biased operations. We propose and compare two post analysis methods in which a first-order DPA, and pattern matching after calculation of DoMs are used, respectively. Finally, we propose a possible algorithm-level countermeasure against the attack based on DPA attack. However, our countermeasure cannot prevent the attack based on DoMs and pattern matching completely, and the corresponding countermeasures should be considered in the future.

References

1. Brier, E., Clavier, C., Oliver, 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)
2. Coron, J.-S.: Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 292–302. Springer, Heidelberg (1999)
3. Coron, J.-S., Giraud, C., Prouff, E.: Attack and improvement of a secure S-box calculation based on the Fourier transform. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 1–14. Springer, Heidelberg (2008)
4. Coron, J.-S., Goubin, L.: On boolean and arithmetic masking against differential power analysis. In: Paar, C., Koç, Ç.K. (eds.) CHES 2000. LNCS, vol. 1965, pp. 231–237. Springer, Heidelberg (2000)
5. Research Center for Information Security (RCIS). Side-channel attack standard evaluation board (SASEBO), `http://www.rcis.aist.go.jp/special/SASEBO/index-en.html`
6. Itoh, K., Takenaka, M., Torii, N.: DPA countermeasure based on the masking method. In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 440–456. Springer, Heidelberg (2002)
7. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
8. Prouff, E., Giraud, C., Aumônier, S.: Provably secure S-box implementation based on Fourier transform. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 216–230. Springer, Heidelberg (2006)
9. Rijmen, V.: Efficient implementation of the Rijndael S-box, `citeseer.ist.psu.edu/293912.html`
10. Schaumont, P., Tiri, K.: Masking and dual-rail logic don't add up. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 95–106. Springer, Heidelberg (2007)
11. Suzuki, D., Saeki, M., Ichikawa, T.: Random switching logic: A new countermeasure against DPA and second-order DPA at the logic level. IEICE Transaction on Fundamentals E90-A(1), 160–169 (2007)
12. Tiri, K., Schaumont, P.: Changing the odds against masked logic. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 134–146. Springer, Heidelberg (2007)

13. Tiri, K., Verbauwhede, I.: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: Proceedings of Design, Automation and Test in Europe Conference (DATE), pp. 246–251 (2004)

Appendix

A The Biased Rate of R_2 after PDF Separation

Suppose that the power consumption of $+$ and $-$ operations, denoted as P_{add} and P_{sub} respectively, both follow a normal distribution with different means of μ_{add} and μ_{sub} and the same deviation σ.

As long as there exists a comparably big difference between μ_{add} and μ_{sub}, We expect that the histogram of points at the appropriate time has the shape as shown in Fig. 8. The two peaks in Fig. 8 correspond to μ_{add} and μ_{sub}, respectively. The shape of

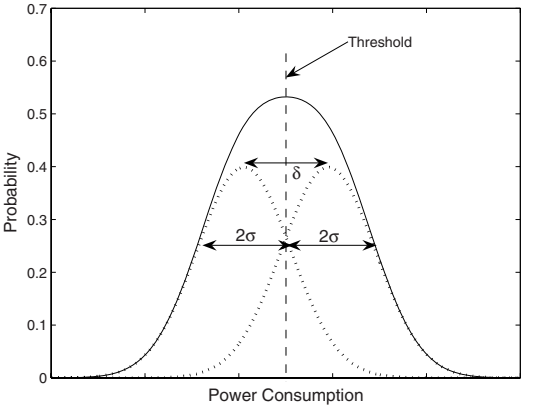

Fig. 8. PDF of two normal distributions

every mountain relates to the value of the standard deviation σ. Because $\mu_{add} \neq \mu_{sub}$, two distributions will not overlap perfectly. If we use the middle of the two distributions (*i.e.* the mean of all power traces) as a threshold to separate power traces into two subgroups, we can expect in this loop one subgroup contains the addition cases more than the subtraction cases, and vice versa for the other group. The biased operation is equivalent to the biased value of R_2. Denoting the R_2 which accounts for more than a half in one group as *the major* R_2, and *the separation rate* is defined as the ratio of the number of traces with the major R_2 to the total number of traces in one group. Denoting $\mu_{add} - \mu_{sub}$ and the separation rate by δ and α, respectively. The separation rate α can be computed by the ratio of δ to the standard deviation σ as shown in Fig. 9.

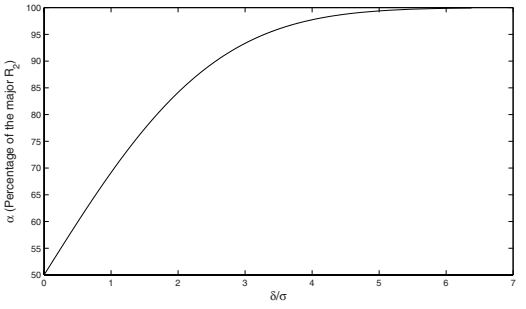

Fig. 9. The relationship between δ/σ and α

Suppose that the value of δ ($\delta > 0$) is specific in a certain implementation. The result of our attack scenario only relates to the value of σ. The smaller the σ is, the higher the separation rate α becomes and the bigger peak we can get[1].

B Comparison between PDF+CPA and Second Order DPA

If we stretch the interpretation of second-order DPA, PDF+CPA attack can be regarded as a kind of second-order DPA because two different attack points are used for power analysis. When permitting this extended interpretation, it is natural for us to have been successful in attacking Coron's S-box. However, the efficiency of PDF+CPA attack is better than or equivalent to first-order DPA in terms of the number of power traces. Therefore, the separation step in our two-step approach can be considered as a sort of the filtering processes that make it possible or easy to implement the succeeding first-order DPA.

It is worth mentioning that our attack can reduce the number of traces compared to conventional first-order DPA attacks. To explain this, we consider attacking Prouff's S-box (refer to [8] for the detailed algorithm). Coron *et al.* pointed out a flaw

$$T = i \cdot \tilde{Z} \oplus R_1 \cdot (\tilde{Z} \oplus i \oplus R_2) \quad = i \cdot Z \oplus R_1 \cdot (\tilde{Z} \oplus R_2), \tag{10}$$

where R_1 and R_2 are n-bit random variables, i is the loop index and $\tilde{Z} = Z \oplus R_1$. Coron *et al.* utilized the fact that $R_1 \cdot (\tilde{Z} \oplus R_2)$ equals to 0 or 1, respectively with the probabilities of $17/32$ or $15/32$ (see the lemma in Sect. 4.1 in [3]). They applied first-order DPA by using a set of the power traces where the bias is not very prominent as the absolute difference between the probabilities λ is only $1/16$. And it turned out that λ in our experiment is much more than $1/16$ after the PDF separation for both cases of Coron's S-box and Prouff's S-box. This way, our attack can reduce the number of power traces compared to a straightforward first-order DPA.

[1] A rough approximation for the amplitude of the peak in difference of means is $(2\alpha - 1)\delta$.

C Algorithm of a Possible Countermeasure against PDF+CPA Attack

Algorithm 2. Coron's S-box with Randomized For Loop

INPUTS: A masked value $\tilde{Z} = Z \oplus R_1$ and the mask R_1

OUTPUT: The 3-tuple output $\Big(\big((-1)^{R_2} F(Z) + R_3\big) \bmod 2^n, R_3, R_2\Big)$

1. Generate a random bit R_2
2. Generate two n-bit random R_3 and R_4
2a. Generate a look-up table $q[l]$ $(0 \leq l \leq 15)$ (each entry has a different integer randomly chosen from $0, 1, \cdots 15$
3. $result \leftarrow 2^n R_3 + R_4$
4. **for** l **from** 0 **to** $2^n - 1$ **do**
4a. $i \leftarrow q[l]$
5-12. The same as Alg. 1.
13. return $(result, R_3, R_2)$