On the Leakage between Arithmetic Components of DES Algorithm

Yijie Ge, Zheng Guo, Zhigang Mao, Junrong Liu, Jiachao Chen
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China

***Abstract*—Security of cryptographic embedded devices has become a prevalent concern, especially since the introduction of Differential Power Analysis (DPA) by Paul Kocher et al. In the past years, many efforts have been made to improve the resistance against Side Channel Attack (SCA) of cryptographic devices. Among the countermeasures, masking is a typical and efficient strategy. However, a number of effective attacks on masked cryptographic devices have been developed in recent years, and this paper continues this line of research. On theory, a DES with a masking scheme is secure under first order SCA, but we dig out a new leakage problem which makes it possible to attack a masked DES without using higher-order power analysis. Concretely, we perform a first-order correlation power analysis based on the leakage relationship between two different arithmetic components of DES. And the reason for this leakage is analyzed and verified by us through simulation and real card attacks.**

Keywords* —Data Encryption Standard (DES); Differential Power Analysis (DPA); Side Channel Attack (SCA); Leakage between components; Masking; Smartcards.

I. INTRODUCTION

The concept of Differential Power Analysis was introduced by Paul Kocher et al in 1998[1] [2], and it soon attracted great attention to the security problem of cryptographic devices. DPA belongs to the family of Side Channel Attack and DPA is undoubtedly regarded as the most popular one. Its principle is to make use of the correlation between a key-related intermediate value and the power consumption of the cryptographic device. This relies on the fact that power always reflects the clues of data being processed, so key information will be derived if correct matches can are made between power traces and key-related values.

After DPA was published, cryptographic algorithms have faced severe challenges. Consequently, many countermeasures have been proposed by researchers. Thomas Messerges developed a general countermeasure by a masking method [3]. Akkar and Giraud proposed a transformed masking scheme by modifying the S-box [4]. Goubin and al came up with a more general way by duplicating the circuit so that the power consumption of the entire device becomes resistant against power analysis [5]. This method calls for great area overhead obviously and the practical effect is hard to meet with the ideal condition, so masking becomes more popular because of its effectiveness and low cost.

In this paper, we mainly research on a DES [6] implementation with masking scheme on a smartcard [7]. By discovering a certain association between two different arithmetic parts of DES, we find a novel leakage condition to be used to launch first order DPA on masked DES which obtains satisfactory results. In normal circumstances, when we suppose a point where its power consumption releases the key information, we simulate its power value using a guessed key through a suitable power model, and this value is later used to calculate correlation with practical power traces. Intuitively, the power value deduced from a right key must lead to a distinctly higher correlation coefficient than that of others. However, once implementing a masking method, attackers cannot deduce the correct intermediate value any longer without knowledge of the masking number. On the contrary, using the leakage relationship which we will introduce in the following parts, such a tight protection of a target value can be invalid as long as it’s correlated to another module which leaks key related information.

Organization of this paper: Section 2 provides the necessary background about SCA, DES, and masking method. Section 3 describes our masking strategy and introduces our discovery of a new leakage problem. We will analyze the reason in this part and assume this leakage condition makes it possible to successfully attack a masked DES just using first order DPA based on the relationship between two different arithmetic modules. We verify our analysis and show our experiments in section 4. And we conclude in section 5.

II. BACKGROUND

A. Side Channel Attack

Side Channel Attack, first introduced in [10], generally exploits the data dependency and operation dependency of cryptographic devices leaked by physically observable phenomena. Typical instances include timing, power consumption, and electromagnetic radiation of integrated circuits. We mainly focus on power analysis in this paper. Power analysis is generally divided into two classes, Simple Power Analysis (SPA) and Differential Power Analysis (DPA). SPA attempts to deduce useful information from a single or a few power traces. SPA doesn’t always aim at recovering key immediately, but works as a necessary and efficient procedure before DPA. It’s often used to distinguish the encryption rounds and interested operations from other interference, so later analysis can be operated more specifically and efficiently.

The work is supported by National Natural Science Foundation of China (No. 61402286, 61472250, 61472249), National Science and Technology Major Project Funds(No. 2014ZX01032401-001), and Enterprise Innovation and Industry-university-research Cooperation Funds of Major Project by SHEITC (No. CXY-2013-35).

In contrast, DPA makes use of the correlation between the data under operation and the power consumption of a device. DPA is quite an effective attack method based on the existence of data dependency in the actual power consumption, and classical DPA [8] is widely adopted. Nowadays, another very powerful method using the idea of machine learning, template attack [9], becomes increasingly popular. No matter in DPA or in template attack, divide-and-conquer strategy is always implemented. And both these attacks are based on grouping power traces according to different values of a targeted intermediate variable.

B. DES Algorithm

DES is a famous symmetric-key algorithm for the encryption of electronic data, and it is quite influential and widely used. DES produces one 64-bit block of encrypted data from one 56-bit key and 64-bit input data. It consists of an initial permutation (IP), 16 rounds of function f, and a final permutation process (FP), which is also the inverse of IP. The overall structure of DES is shown in Fig. 1.

The f function works as follows. It first expands the right side (32 bits) of the block into 48 bits, then it performs exclusive-or with the roundkey. After that, it's divided into eight 6-bit groups, and these groups of data enter a nonlinear function called S-boxes respectively. Each S-box compresses its 6-bit input data into 4-bit output data. Then eight 4-bit S-box outputs make a new 32-bit data and later permutated by a P-box. The 32-bit output of P-box XORs the initial left part of the block and the result becomes the right side of the input block of the next round while the initial right side of this round works as the left side of the next round. Its structure can be explained by Fig. 2.

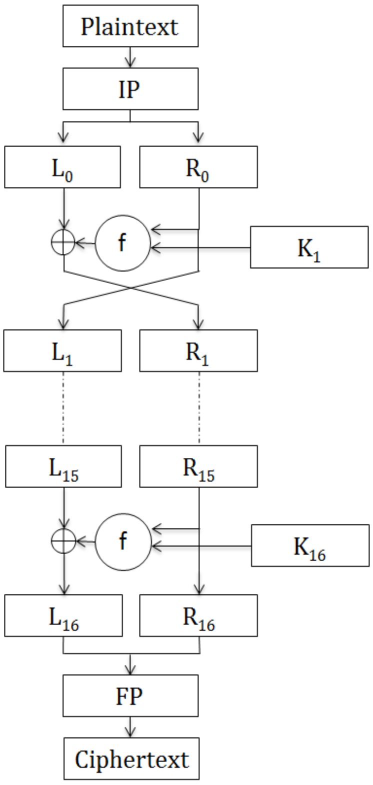


Fig. 1. Overall structure of DES

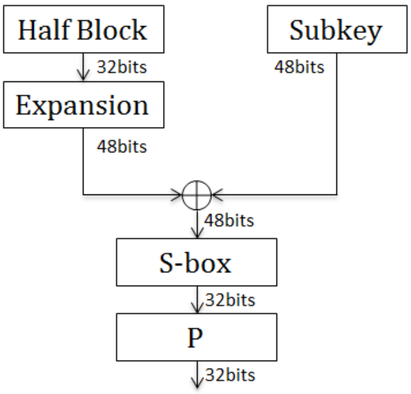


Fig. 2. The structure of the function f

C. The Masking Method

Masking method was initially suggested by Akkar et al, and during the past years, several different masking schemes have been proposed in [12] [13] [14], and soon followed by attacks aiming at these schemes such as [11].

The basic idea of masking is to break the dependency between the intermediate variables and the practical power consumption of a device. Concretely, in a masking scheme, every intermediate value v of an algorithm should be masked into another value v_m by performing a calculation with a random number m, and the choice of the calculation is the key point of a masking scheme. Ordinarily, this calculation includes exclusive-or, modular addition combined with algebraic addition, modular multiplication, and so forth. Ideally, once we mask the original intermediate variable v with m, during the process of encryption, the device is operating the masked data. As a result, no correlation with a hypothetical variable should be found by the attacker in the power trace now.

During the design and implementation of masking, designer should also pay attention to these points. 1) The randomness of m. Weak randomness of m may fail to protect the intermediate value. 2) The update of m. If a masking designer choose to use the same m for every round, this algorithm will still be vulnerable to power analysis when the attack use a hamming distance model to simulate the power. 3) The value of m should be impossible to detect by attackers.

III. Discovery and Analysis

A. Our Masking Scheme

In this paper, we design the masking scheme based on the principle introduced in section 2, and our masking strategy will be explained in the following part. First of all, some definitions should be declared.

E(x) stands for the expansion operation of DES. The input x should be 32-bit while E(x) returns 48-bit.

S(x) stands for the S-box operation of DES. The input x should be 48-bit while S(x) returns 32-bit.

P(x) stands for the P-box Permutation of DES. The input x should be 32-bit and P(x) is of the same length.

As for a 32-bit random number m and a 48-bit x, we have:

$$S_{1m}(x) = S\big(x \oplus E(m)\big) \qquad (1)$$

$$S_{2m}(x) = S(x) \oplus P^{-1}(m) \qquad (2)$$

Where P^{-1} means the inversion function of P. In original DES, $f(x) = P(S(E(x) \oplus K))$, now we have:

$$f_{1m}(x) = P\big(S_{1m}(E(x) \oplus K)\big) \qquad (3)$$

$$f_{2m}(x) = P\big(S_{2m}(E(x) \oplus K)\big) \qquad (4)$$

So, after a f_{1m} conversion, mask m would be eliminated, while a f_{2m} will mask the data with m. Now, with the definitions above, we have different situations for every round and we use different f functions for them.

A: original f;

B: input data without mask, f' = f2;

C: the right side of input is masked, f' = f1;

D: the left side of input is masked, f' = f;

E: the left side of input is masked, f' = f2;

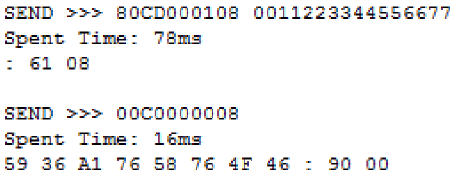
We generate two random numbers a and b for masking, then our DES implementation arranges its rounds as the following order:

$$IP \rightarrow B_aC_aD_aC_aD_aC_aE_aB_bC_bD_bC_bD_bC_bD_bC_bE_b \rightarrow FP$$

With the above definitions, we successfully design a masked DES. And it masks its first and last round with different random number, so any traditional leakage points are supposed to become invalid to this design. However, after attempting all those traditional leakage points and leakage models in our experiments, something new has been found. We will introduce and analyze this new leakage phenomenon in the next part.

B. A New Leakage Condition

We implement our masked DES on a smart card to ensure the flexibility and to simulate its application situation. After adding our algorithm into COS (Card Operating System), we test it by sending specified APDU (Application Protocol Data Unit) and the card returns us the correct result after encryption algorithm. Fig. 3 shows the command and return APDU.



```
SEND >>> 80CD000108 0011223344556677
Spent Time: 78ms
: 61 08

SEND >>> 00C0000008
Spent Time: 16ms
59 36 A1 76 58 76 4F 46 : 90 00
```

Fig. 3. The command and response APDU of our DES card.

After our functional test, we continue our further steps. In theory, a masking scheme makes sure the intermediate value in our algorithm not to reveal key information. However, a novel location of leakage or say a novel condition of leakage appears when we launch DPA on it. We successfully recover its key when we use the leakage model XOR+ABS and choose round 16 as the target round.

To make things clear, we will introduce what's the actual target intermediate value in this situation first. Fig. 4 depicts the last two rounds of DES which we focus on. Using the definitions we have declared above, this intermediate variable can be described as follows.

Target v when using model XOR + ABS and choosing round 16 as the target round:

$$v = S(E(R15) \oplus K) \oplus P^{-1}(R16) \qquad (5)$$

However, this v is no longer directly processed in our DES because it is masked with b. In another word, actual intermediate variable v' in our DES is:

$$v' = S(E(R15) \oplus K) \oplus P^{-1}(R16) \oplus P^{-1}(b) \qquad (6)$$

It's easy to detect that the expression in (5) and (6) can be simplified with L15:

$$v = P^{-1}(L15) \qquad (7)$$

$$v' = P^{-1}(L15 \oplus b) \qquad (8)$$

So the attack result seems to be strange for that there couldn't be any correlation since v' has already been masked.

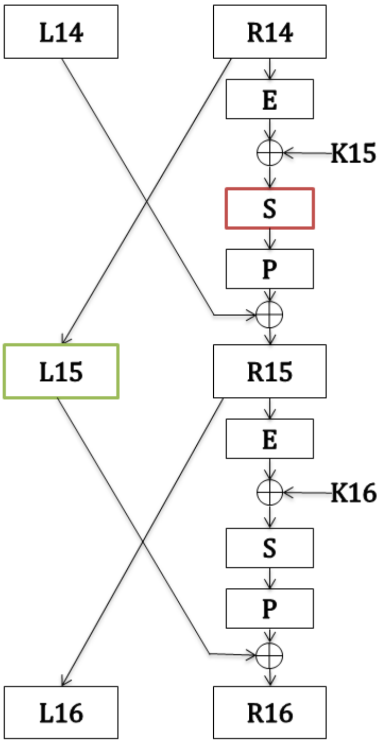


Fig. 4. The last two rounds of DES

It’s true that L15 in our DES is masked with a random number, however we assume that the leakage relationship between two arithmetic modules of the algorithm leads our attack to success.

As we all know, our target intermediate value L15 ought to be masked so that no useful power information is expected to leak here. However, from another point of view, the S-boxes of DES are not perfectly designed. Since the bit number of its output doesn’t equal that of its input, so some distribution bias inevitably exists. Then, there could be a leak situation brought about by this bias. It’s likely that the leakage problem caused by S-boxes is reflected by L15 to a certain extent. In this case, correlation is sure to be found between them, and in the following part, we will verify this deduction with both simulation and practical attack.

IV. Experimental results

A. Simulation Experiments

In this part, we will demonstrate our experiment process and corresponding results. Before performing our attack on real card, we first do some simulation work. To verify our assumption, we try to find some correlation between our masked intermediate variable and the S-box output of round 15. We perform our simulation experiments in the following order:

a) Guess a subkey of roundkey K16.

b) Deduce L16 (R15) and R16 from the ciphertext we’ve already got.

c) Make use of R15 and R16 as well as the subkey to get the corresponding L15 which is also R14. It should be noticed here that a roundkey is consist of eight 6-bit subkeys, we process them respectively. From one 6-bit subkey, we can deduce 4 nonadjacent bits in L15.

d) Use the hamming weight of these four nonadjacent bits in L15 to calculate correlation with the power consumption of the S-box output of round 15.

e) We perform a) to d) for every guessed subkey. And repeat this process for eight times to complete the correlation calculation of all the eight subkeys.

In practice, we simulate the output of the related S-boxes using expression (1) and choose hamming weight as their power model. Every roundkey consists of eight 6-bit sub parts. We use divide-and-conquer strategy to accomplish the task. Every 6-bit subkey will be diffused into 4 separate bits after P transformation. And then we can obtain four nonadjacent bits in L15, which we exactly use to calculate correlation with the power consumption of S-box in round 15. According to the structure of DES, four to six S-boxes will be influenced by these four bits, and the actual relationship between them is summarized by us in TABLE I. After performing the correlation analysis according to the above procedures, among all the theoretically related S-boxes which are shown in the second column, about 50%~60% of them have obvious correlation with the intermediate value after 10 thousand simulated traces, and the S-boxes which shows correlation are listed in the third column. For example, as to Subkey 1, it’s expected to influence the second, third, fourth, fifth, sixth, and eighth S-boxes in round 15 in theory, but our analysis result shows only the third, sixth, and eighth display obvious correlation. Although only a part of the S-boxes in the second column convincingly demonstrate expected correlation, it’s already firm enough to prove that correlation exists between L15 and S-boxes in the 15th round .

TABLE I. SUBKEY RELATED S-BOXES AND ATTACK RESULT

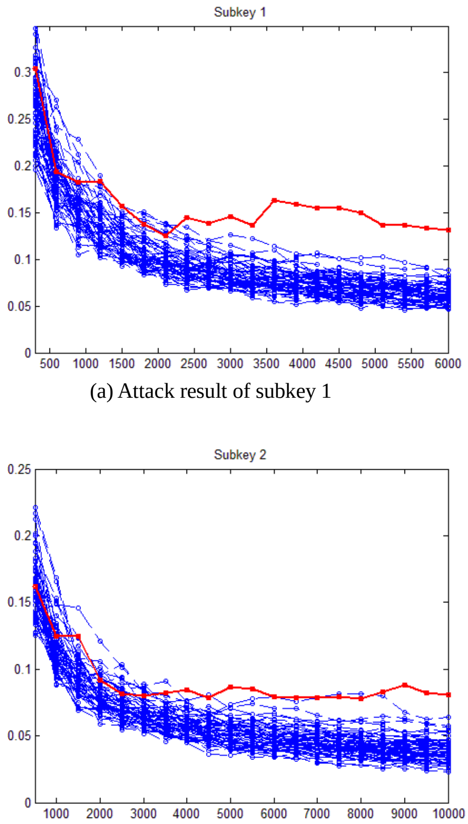
Subkey	**Related S-boxes**	**Successfully match**
Subkey 1	2, 3, 4, 5, 6, 8	3, 6, 8
Subkey 2	1, 3, 4, 5, 7, 8	1, 4, 5, 7
Subkey 3	2, 4, 6, 7, 8	2, 4, 6
Subkey 4	1, 3, 5, 6, 7, 8	1, 3, 5, 7
Subkey 5	1, 2, 4, 5, 7	1, 7
Subkey 6	1, 3, 5, 7	1, 3
Subkey 7	1, 2, 3, 4, 6, 8	2, 6, 8
Subkey 8	1, 2, 4, 5, 6, 7	2, 4, 6, 7

TABLE II. SUBKEY RELATED P-BOXES AND ATTACK RESULT

Subkey	**P-box 1**	**P-box 2**	**P-box 3**	**P-box 4**
subkey1	√	√	√	√
subkey2	√		√	√
subkey3	√	√	√	√
subkey4	√	√	√	√
subkey5	√		√	
subkey6	√	√		√
subkey7			√	√
subkey8	√	√	√	

Actually, to further consider the leakage problem, we may also suppose that our intermediate value L15 can have correlation with P-boxes in round 15. Table 2 is about the calculation result of correlation with P-box output, and most of them show correlation with intermediate value calculated with the correct subkeys.

In TABLE I and TABLE II, when successfully detecting correlation, the value is approximately 0.09~0.1. Actually, the simulated power values we calculate for different S-box outputs or P-box outputs corresponding to different plaintext-ciphertext pairs can be regarded as simulated power traces with few power consumption points. And this experiment actually launches correlation power analysis upon the traces using L15 as the intermediate value. So this simulation experiment doesn’t only verify the existence of



(a) Attack result of subkey 1

(b) Attack result of subkey 2

Fig 5. The attack result of subkeys. Each line stands for the correlation variation of a guessed subkey with the number of power traces increase, and the red line stands for the correct subkey value.

correlation between masked L15 and S-boxes or P-boxes in round 15, but also prove the effectiveness to launch a first order DPA on a masked DES using the leakage problem between different components.

B. *Real Card Analysis*

Although we have already verified our assumption in part A, we choose to strengthen the proof by launching a real card attack. We realize the masked DES on a smart card using the masking scheme introduced in Section III and the trace of the entire algorithm shows as fig. 6. We can easily figure out the outline of 16 rounds of DES. According to our experience from attacking an unmasked DES also designed by ourselves, we can efficiently locate the part of S-box of the 15th round in the trace.

In this experiment, we take the following points into account:

a) To further confirm our assumption and explain the leakage, we try to detect correlation between masked L15 and practical power value of S-boxes in round 15.

b) The problem of correlation detection can be transformed into an attack. Using the hamming weight of four nonadjacent bits of L15 deduced by a guessed subkey to match the leakage point in round 15. And the procedures are similar to those in our simulation experiments.

Adopting our strategy stated above, we succeed finding the correlation between 4 nonadjacent bits of L15 and the s-box output of round 15. Fig. 5 shows us the attack result of subkey 1 and subkey 2.From the this result, we can detect that 1) at first, the right subkey has similar correlation with other guessed candidates, but it gradually becomes stable at a value after analyzing more than 2000 thousand traces, while the other candidate has lower correlation as the number of traces increase.2) Compared with normal attack models, the value and the leading percentage of the corrected subkey are lower. For every S-box, their 1st ranked candidate subkey owns the correlation no more than 0.15 after it takes the lead. But it's already effective enough to verify our assumption and to be made use of attacking masked DES.

Now, both simulation and real card experiments have shown correlation between masked L15 and S-box output of round 15. It verifies our assumption in Section III. The basic idea lies in that even if the intermediate value is protected by a masking number, it will still be vulnerable to correlation power analysis if any correlation exists between this value and another value which leaks key information, and the unbalanced distribution of DES S-box output contributes to this correlation in this paper.

V. CONCLUSIONS

We demonstrate a new leakage problem caused by the correlation between different arithmetic components of a cryptographic algorithm through both simulation and real card attacks. Our evaluation results shows this leakage condition can be used to launch an attack on a cryptographic device even

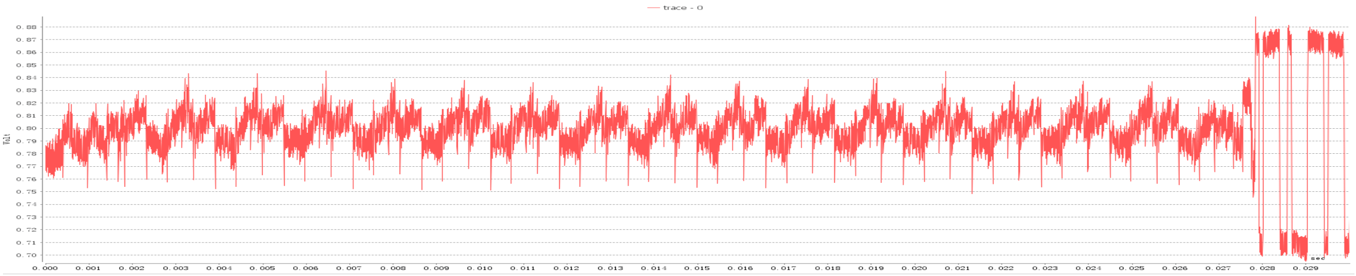


Fig. 6. The power trace of the entire DES encryption

if the original target intermediate variable is tightly protected. Just like in our attack, a first order correlation power analysis taking the leakage between different components into account is powerful enough to break a masked DES.

Our attack and evaluation is performed without any known-key methods, which is closer to the view of a real attacker, and the result shows that our discovery is quite effective in attacks. This leads us to update our view of DPA. In traditional DPA, we tend to use different power models to match the real power consumption of the point where we assume the key information leaks, but in this paper, a new point of view has shown to us. If assumed intermediate value no longer matches with practical power, just as in a masked cryptographic algorithm, power analysis may still be valid making use of the correlation between different parts of the algorithm which always exists. Varieties of masking schemes all changed the intermediate values, but little work can be done on altering the leakage correlation between different arithmetic parts. So the leakage problem revealed in this paper should be a new element to consider for algorithm and countermeasure designers in the future.

REFERENCE

[1] Paul Kocher, Joshua Jaffe and Benjamin Jun, “Introduction to Differential Power Analysis and Related Attacks”, http://www.cryptography.com/dpa/technical,1998.

[2] Paul Kocher, Joshua Jaffe and Benjamin Jun, “Differential Power Analysis”, in Proceedings of Advances in Cryptology - CRYPTO'99, Springer-Verlag, 1999, pp.388-397.

[3] Thomas S. Messerges, “Securing the AES Finalists Against Power Analysis Attacks”, in Proceedings of Fast Software Encryption Workshop 2000, Springer-Verlag, April 2000.

[4] Akkar, M.-L., Giraud, C. “An Implementation of DES and AES, Secure against Some Attacks.” In: Koç, Ç.K., Naccache, D., Paar, C. eds. (2001) Cryptographic Hardware and Embedded Systems - CHES 2001. Springer, Heidelberg, pp. 309-318

[5] Louis Goubin and Jacques Patarin, “DES and Differential Power Analysis – The Duplication Method”, in Proceedings of Workshop on Cryptographic Hardware and Embedded Systems, Springer-Verlag, August 1999, pp. 158-172.

[6] National Bureau of Standards. The data encryption standard. FIPS PUB 46, 1977.

[7] Thomas S. Messerges, Ezzy A. Dabbish and Robert H. Sloan, “Investigations of Power Analysis Attacks on Smartcards”, in Proceedings of USENIX Workshop on Smartcard Technology, May 1999, pp. 151-161.

[8] Brier, E., Clavier, C., Olivier, “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).

[9] Chari, S., Rao, J.R., 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).

[10] Kocher, Paul C. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems” — CRYPTO ’96. Springer Berlin Heidelberg, 1996:104--113.

[11] Akkar, Mehdi-Laurent, et al. "Two power analysis attacks against one-mask methods." Proceedings of Fast Software Encryption Lecture Notes in Computer Science (2004).

[12] Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-box. InFast Sof tware Encryption, 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Proceedings, volume 3557 ofLecture Notes in Computer Science, Springer, 2005.

[13] Elena Trichina and Tymur Korkishko. Small Size, Low Power, Side ChannelImmune AES Coprocessor: Design and Synthesis Results. In Proceedings of the Fourth Conference on the Advanced Encryption Standard (AES), 2004

[14] Johannes Bl ¨ omer, Jorge Guajardo, and Volker Krummel. Provably Secure Masking of AES. InSelected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers, volume 3357 ofLecture Notes in Computer Science, pages 69–83. Springer, 2005.