Against Double Fault Attacks Based Countermeasures for Second Order Infection Mechanism

Jinbao Zhang, Ning WU, Xiaoqiang Zhang, Lili Shen and Fang Zhou

***Abstract*—Fault attack is a very effective way to crack the key for encryption chip, what is worse, most existing infection mechanisms are based on the signal fault assumption, and it is very difficult to resist double fault attacks, meanwhile, single map diffusion function can be broke by single byte fault attack. Aiming at the problems above, we propose a novel Second Order Infection Mechanism based on duplicating circuits to resist double fault attack. Furthermore, we use random numbers to make the fault diffusion randomization to resist single byte fault attack. The experimental results show the AES circuit that using the scheme proposed in this paper can resist fault attacks effectively including double fault attacks and single byte fault attack.**

***Index Terms*—Infection mechanism, AES, double fault attacks, side- channel attack**

I. Introduction

FAULT attack, as one kind of physical attacks, poses serious threats to embedded devices. Fault attack on a block cipher is usually mentioned as a differential fault attack(DFA) proposed by Biham and Shamir [1]. The AES (advanced encryption standard)[2], which replaces the data encryption standard (DES) for symmetric key encryption, as its popularity and status as a representative block cipher, has became the main target of DFAs [3], [4].

To effectively respond to fault attacks, previous research has proposed many countermeasures, mainly divided into two classes: the ones based on detection [4], [5] and [6]. Detection countermeasures aim to determine whether there is a fault occurred by comparing the results of the two operation through duplication or multiplex techniques for some computing, modules, or the whole algorithm[6]. If an error is detected, the algorithm does not output the true faulty ciphertext, so preventing its exploitation. But the comparison step itself is prone to fault attacks due to the detection position is related to the data being processed [7]. The other ones are based on infection [8], [9]. Infection countermeasures aim to destroy the fault invariant by diffusing the effect of a fault in such a way that it renders the faulty ciphertext unexploitable. Infection countermeasures are preferred to detection as they avoid the use of attack vulnerable operations such as comparison. But, it's not enough to resist fault attacks only using infection mechanisms without introducing the idea of randomization [9]. In 2012, Gierlichs et al [10] proposed an infection countermeasure using of dummy rounds and redundant computation with consistency checks to prevent fault attack. The paper [11]pointed out the infection countermeasure based on dummy rounds proposed by [10] was flawed and had carried on a successful attack to it, the paper [7] further pronounced the reason why the infection countermeasure in [10] insufficient was because the infection process using the same unknown mask, and made a improvement for it.

Unfortunately, most existing countermeasures which using infection mechanisms to against fault attacks are based on single fault assumption, so it is very difficult to defense double fault attacks which can bypass the existing infection mechanisms[6], [9]. What is worse, with the significantly improving accuracy of fault injections in recent years, it has become possible to carry on double fault attacks at a certain time [10], [12] and [13].

In view of these reasons above, this paper proposes a countermeasure called Second Order Infection Mechanism to resist double fault attacks based on the research of [7] and [9], we use three duplicating circuits to construct the diffusion function. Furthermore, in order to against single byte fault attack, we design a randomized diffusion function to achieve the goal of fault diffusion randomization. We will first prove the feasibility of the proposed countermeasure by theoretical analysis, and then show the efficiency of countermeasure for preventing double fault attacks and single byte error attack by experiments.

The rest of this paper is organized as follows: in Section II, we first briefly introduce the notations about AES, and then give a brief introduction of fault attacks. In section III, the fault attacks countermeasure: infection mechanism based on repetition or duplication is briefly described. And we discuss the flawed of security models in certain countermeasure designs which lead to attacks. In section IV, we propose our fault-injection countermeasure Second Order Infection Mechanism which based on three duplicating circuits to resist

Manuscript received July 19, 2016; revised August 10, 2016. This work was supported by the National Natural Science Foundation of China(No. 61376025), the Natural Science Foundation of Jiangsu Province(No. BK20160806).

J. Zhang is with the College of Electronic and Information Engineering, NUAA, Nanjing, 210016, China (e-mail: zjb4050811@126.com).

N. WU is with the College of Electronic and Information Engineering, NUAA, Nanjing, 210016, China (e-mail: wunee@nuaa.edu.cn).

X. Zhang is with the College of Electronic and Information Engineering, NUAA, Nanjing, 210016, China (e-mail: zxq198111@qq.com).

L. Shen is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China, and with Jincheng college, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China.(email:shirely_ivy@163.com).

F.Zhou is with the College of Electronic and Information Engineering, NUAA, Nanjing, 210016, China (e-mail: zfnuaa@edu.cn).

double fault attacks, and we will prove the feasibility of the proposed countermeasure by theoretical analysis. In section V, we present the experiments of our new countermeasures to defense double Fault Attacks and single byte error attacks. Finally, concluding remarks are presented in Section VI.

II. Preliminary

A. Notations

The AES [2] is a symmetric block cipher algorithm that can encrypt and decrypt a 128-bit data with three different key sizes: 128, 192 and 256 bits (respectively called AES-128, AES-192 and AES-256). In this paper, the cryptographic algorithm we focus on is AES-128 because of its popularity and simple description. AES-128 has 10 rounds, and each round is consisted of four operations: SubBytes(SB), ShiftRows(SR), MixColumns(MC), AddRoundKey(ARK), except for the first round and the last round. During the encryption or decryption process, the 16 bytes plaintext is transformed into a 4x4 byte matrix referred to as State.

B. Fault Attacks

Fault attacks [1] consist of forcing a cryptographic device to perform some erroneous operations, hoping that the result of that wrong behavior will leak information about the secret parameters involved. As the diffusion pattern of the encryption algorithm is known, the attacker could assume the form of fault diffusion by accurately controlling the location and size of the injected faults, further derive the equality relation between faulty ciphertexts, correct ciphertexts and injected faults. Finally, the attacker could break the keys according to the equality relation.

III. Infection Countermeasure

A. The principle of Infection Countermeasure

Infection countermeasures, by expanding the logical effects of injected faults to a larger range, rather than simply follow the diffusion pattern of the encryption algorithm itself, reducing the correlation between the faulty ciphertexts and the keys, so making the faulty ciphertexts could't be used to recover the keys. Fig. 1 describes the principle of Infection countermeasure.

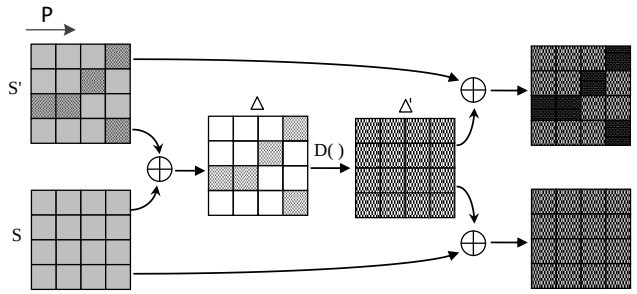

Fig. 1. The principle of Infection Countermeasure

As shown in Fig. 1, in duplication circuit, two encryptions of the same plaintext P are performed simultaneously. When a cryptographic operation has performed, the difference Δ between State matrix S and S' is computed. The diffusion of the faults is performed using a diffusion function D computing $D(\Delta)= \Delta'$, and re-injected in the current States: $S = S \oplus \Delta'$ and $S' = S' \oplus \Delta'$. In here, the diffusion function D must meet D (0) = 0.

In [14], a diffusion function D is proposed such as the expression (1), denoting 4x4 matrix of bytes as Δ, Δ_{ij} represents the element at row i and column j.

$$\Delta'_{ij} = \bigoplus_{n=0}^{3} \Delta_{in} \oplus \bigoplus_{m=0}^{3} \Delta_{mj} \quad (1)$$

This diffusion function could achieve such diffusion effect as illustrated in Fig. 2. Once there is any nonzero element among Δ, the vale of nonzero element will be diffused to the row and column of the element after the operation of diffusion function D.

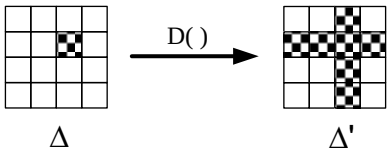

Fig.2. The diffusion effect of diffusion function D

But the infection countermeasure mentioned above is actually flawed.

B. The flawed

Although the generic infection mechanism mentioned above can resist fault attacks some extent, there are still some shortcomings.

(a) The infection mechanism mentioned above is only based on the single fault assumption and can not resist the double fault attacks.

As illustrated in Fig. 1, we assume injecting two same specified faults during one execution into the two encryption paths, respectively, i.e. double fault attacks, then S = S', $\Delta = S \oplus S'=0$, $D(\Delta)=0$, so the value of Δ' is still 0 through diffusion operation, this means the diffusion function D does not play the role of fault diffusion and the double fault attacks can bypass the existing infection countermeasures.

(b) As the diffusion function D is fixed, so it couldn't resist single byte fault attacks.

If the diffusion function D is a single map function, that is, the value of D is fixed, so the attacker can enumerate the 2^8 different diffusion results when single byte error occur, then use the idea of DFA[1]to analyze these 256 different diffusion conditions to break the keys.

In order to solve the problems above, we propose a countermeasure called as Second Order Infection Mechanism. Our Second Order Infection Mechanism based on three duplicating circuits achieves 2 times infection to resist double fault injection attack. At the same time, we also use random numbers to further against single byte error attacks, which could realize the purpose of fault diffusion randomization.

IV. Fault-injection countermeasure

A. Second Order Infection Mechanism

Double fault attacks, which requires the attacker to clear

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

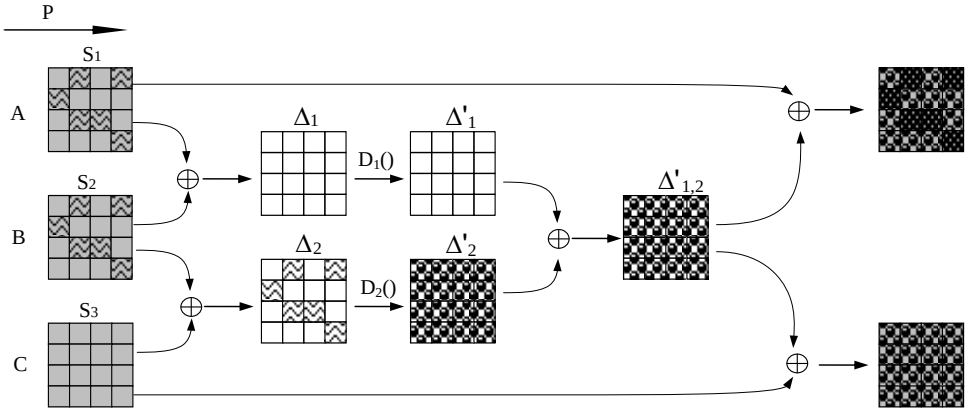

Fig. 3. The principle of Second Order Infection Mechanism

the difference of data between two encrypted paths during injecting errors, then injects the corresponding errors according to the difference to offset the difference between each others, so as to bypass the infection countermeasures. In order to resist double fault attacks, we use three duplicating circuits to achieve 2 times infection, which making the attacker could not determine the difference between the two paths in any time. We call the countermeasure proposed as Second Order Infection Mechanism. The structure of the mechanism is shown in Fig. 3.

As illustrated in Fig. 3, in three duplication circuit A, B and C, three encryptions of the same plaintext P are performed simultaneously. Δ_1 is the difference between State matrix S_1 and S_2, Δ_2 is the difference between State matrix S_2 and S_3, D_1 and D_2 are different diffusion functions, that is, when Δ is not zero, $D_1(\Delta) \neq D_2(\Delta)$, and $D_1(0) = 0$, $D_2(0) = 0$.

For simplicity, we discuss the following conditions separately.

Firstly, we assume injecting two same specified faults during one execution into the two encryption paths A and B, respectively(when the double fault occurred in the encryption paths B and C, the process of discussion is the same). So Δ_1 is zero. $\Delta'_1 = D(\Delta_1) = 0$. Δ_2 is not zero, $\Delta'_2 = D(\Delta_2) \neq 0$, so the second difference $\Delta'_{1,2} = \Delta'_1 \oplus \Delta'_2$ is not zero, this means, the diffusion function effectively play the role of fault diffusion and the double fault attacks could not bypass our infection mechanism.

Secondly, we assume the double fault attacks occurred in the two encryption paths A and C. So Δ_1 and Δ_2 are the same, but due to the diffusion functions D_1 and D_2 are different, $D_1(\Delta_1) \neq D_2(\Delta_2)$, so $\Delta'_{1,2} = \Delta'_1 \oplus \Delta'_2 = D_1(\Delta_1) \oplus D_2(\Delta_2) \neq 0$, that is to say, our infection mechanisms are also effective in this condition.

Thirdly, we assume only injecting single fault during one execution into the encryption path A or B or C, it is easy to analyze that no matter the fault occurred in any path, the second difference $\Delta'_{1,2}$ will not be zero, that is to say, our infection mechanisms can resist single fault attacks effectively.

B. The randomization for Diffusion function

For infection mechanisms, if the diffusion function D is a fixed single map function, the attacker could enumerate all the different diffusion results when single byte error occur and break the keys utilizing the idea of DFA. Based on this reason, we design a randomized diffusion function to improve the infection mechanisms. Our diffusion function Γ is as follows:

$$\begin{cases} \Delta'_{ij} = \bigoplus_{n=0}^{3} \Delta_{in} \oplus \bigoplus_{m=0}^{3} \Delta_{mj} \\ \Gamma = M \bullet \Delta' \end{cases} \quad (2)$$

Among (2), Δ and Δ' denote as 4×4 matrix of bytes, considering Δ_{ij} and Δ'_{ij} as the element at row i and column j, M represents 4×4 random number matrix and is not equal to zero, which is generated by a random number generator. For each time, the value of M is not the same. The resulting Γ is the result of diffusion. When $\Delta = 0$, $\Delta' = 0$, $M \cdot \Delta' = 0$, so meeting the characteristic of diffusion function: $\Gamma(0) = 0$. When $\Delta \neq 0$, $\Delta' \neq 0$, $M \cdot \Delta' \neq 0$, and as M is a random variable, so Γ is random.

According to the formula (1) and (2), this paper proposes a random Second Order Infection Mechanism. The infection mechanism proposed not only could resist double fault injection attacks but also could solve the problem of the single function's mapping value is fixed. The detailed description is as shown in TABLE I.

TABLE I
RANDOM SECOND ORDER INFECTION MECHANISM

Algorithm: Random second order infection mechanism

Input:: plaintext P_1, P_2 and P_3. Random number M_1,M_2 ,and M_1,$M_2 \neq 0$;
Output: ciphertext C ;

（1）S1 ← Cipher(P_1), S2 ←Cipher(P_2), S3←Cipher(P_3);

（2）$\Delta_1 \leftarrow (S_1 \oplus S_2)$, $\Delta_2 \leftarrow (S_2 \oplus S_3)$;

（3）$\Gamma_1 \leftarrow M_1 \cdot D(\Delta_1)$, $\Gamma_2 \leftarrow M_2 \cdot D(\Delta_2)$;

（4）$\Gamma \leftarrow \Gamma_1 \oplus \Gamma_2$;

（5）$S_1 \leftarrow S_1 \oplus \Gamma$， $S_2 \leftarrow S_2 \oplus \Gamma$;

（6）C← Cipher(S_1);

The whole process for the algorithm: three encryptions of the same plaintext P_1, P_2 and P_3 are performed simultaneously, through a number of encryption operations, obtains the intermediate State: S_1, S_2 and S_3, then checks whether there is an error occurred during the encryption process, i.e. whether Δ(Δ_1 or Δ_2)is zero, if $\Delta \neq 0$, indicates there is at least one encryption path occur faults in the encryption process, and then the diffusion function will infect the error to the other

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

bytes, and infects the faults to S_1 and S_2. Finally, outputs random faulty ciphertext C; If $\Delta = 0$, the infection mechanism will be out of action, at last, outputs the right ciphertext C.

V. EXPERIMENTS AND RESULTS

A. Experiment for resisting double fault attack

In order to verify the effectiveness of the infection mechanism proposed in this paper, we have carried out some attack experiments on the AES circuit added the defense measure of our infection mechanism. Without loss of generality, we assume the faults occurred in the paths of P_1 and P_3, and the exact location of the faults occurred after the ninth round of AddRoundKey, and before the tenth round of SubBytes. The schematic diagram of double faults attack is shown in Fig. 4.

Fig. 4, P_1, P_2 and P_3 are the groups of plaintext, and $P_1 = P_2 = P_3$, S^9 (as well as S'^9 and S''^9)represents the State matrix after the completion of encryption for the ninth round, SB represents SubBytes, SR represents ShiftRows, K^{10} is the key

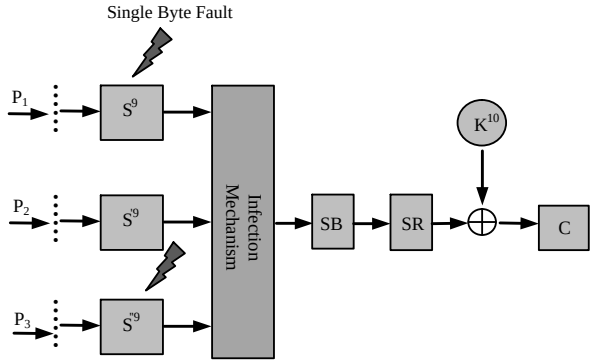

Fig.4. The schematic diagram of double faults attack

for the tenth round of encryption, C is the output ciphertext.

Taking the first byte of the intermediate State after the completion of the ninth round encryption for example, we illustrate the performance of infection mechanism proposed for resisting double fault attacks through experiments. In here, we assume just only one bit fault being injected. The experimental process is shown in TABLE II.

TABLE II
EXPERIMENT FOR RESISTING DOUBLE FAULT ATTACKS

Arbitrary plaintext P and Key K	P: d8 bf 52 38 f8 54 80 65 f4 f6 e1 ec e8 72 49 a5 K: 06 26 1c fc c8 3d 9c a3 7f 42 7e da da b2 c2 82
State matrix after ninth round	S^9: b7 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20 S'^9: b7 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20 S''^9: b7 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20
Injecting double faults for the first byte	S^9: b6 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20 S'^9: b7 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20 S''^9: b6 91 c2 9e cb 68 c5 20 0f 92 4c a7 da 97 e6 20
Random number	M_1: d0 71 6a 29 02 14 4e 9e d5 43 71 63 ec 8f af 07 M_2: d8 bf 52 38 f8 54 80 65 f4 f6 e1 ec e8 72 49 a5
Second difference $\Gamma = \Gamma_1 \oplus \Gamma_2$	Γ: a1 2b db 40 bf f6 a5 e8 ca a9 ba 3e bc 1e 2a cc

Based on the experimental results above, when the same faults are injected, $\Gamma \neq 0$, the attacker could not bypass the infection mechanism, this shows that our Second Order Infection Mechanism has the ability of resisting double fault attacks.

B. Verificating the random for diffusion function

Taking single byte fault attack for instance, we assume the difference Δ_1 between State matrix S^9 and S'^9, only the first byte is not zero, and the difference Δ_2 between State matrix S''^9 and S''^9 is also only the first byte not zero. In order to verify the random of our diffusion function, we need to perform the diffusion function several times to check whether or not when Δ_1 (as well as Δ_2) is the same, the result of diffusing Γ is random. The results of experiments are shown in TABLE III.

From TABLE III, although Δ_1 and Δ_2 are the same, the results of diffusion function are different, further, the results are uncertain and random, so the diffusion function we have proposed in this paper has the characteristic of randomness.

TABLE III
EXPERIMENT FOR RANDOM

Num	the result of diffusing
	Δ_1: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	Δ_2: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1	f2 12 18 c2 ca a9 22 a6 ca a9 22 a6 ca a9 22 a6
2	82 4b 5c 47 4e b2 28 fa 4e b2 28 fa 4e b2 28 fa
3	d7 43 a4 12 87 22 aa 11 87 22 aa 11 87 22 aa 11
4	3e 48 a4 73 c5 fe b5 57 c5 fe b5 57 c5 fe b5 57
5	f4 5b 3b 13 63 a0 c2 c0 63 a0 c2 c0 63 a0 c2 c0
6	a1 b8 2d a0 09 f3 de fd 09 f3 de fd 09 f3 de fd
7	65 c5 fb 01 d6 1a 03 07 d6 1a 03 07 d6 1a 03 07
8	78 69 a2 88 93 ca a3 5a 93 ca a3 5a 93 ca a3 5a
9	f4 d9 fb 1f 0c af bd 07 0c af bd 07 0c af bd 07
10	32 e5 5e d9 ce d6 15 60 ce d6 15 60 ce d6 15 60
11	f2 6a 38 61 d6 b3 7e ef d6 b3 7e ef d6 b3 7e ef
12	34 90 e7 64 b9 a5 a3 2e b9 a5 a3 2e b9 a5 a3 2e
13	08 88 e4 23 d4 89 3c 67 d4 89 3c 67 d4 89 3c 67
14	86 91 00 66 d0 b0 19 6a d0 b0 19 6a d0 b0 19 6a
15	d5 27 2f 22 0b f8 45 aa 0b f8 45 aa 0b f8 45 aa
16	7d f1 36 8c 95 9d 32 66 95 9d 32 66 95 9d 32 66
17	7a e5 41 39 89 c6 8b b0 89 c6 8b b0 89 c6 8b b0
18	f4 aa fa 92 e6 05 6f ac e6 05 6f ac e6 05 6f ac
19	c5 8c 02 bc b7 50 fe 3f b7 50 fe 3f b7 50 fe 3f
20	24 53 5d 7b 05 6d 79 2a 05 6d 79 2a 05 6d 79 2a

In section III, we have point out that if the value of diffusion function D is fixed, when single fault attack occurred, the attacker could enumerate the 2^8 different diffusion results to break the keys. But the map result of our diffusion function is random, so if the attacker wants to recover the key of AES, he/she must enumerate the 2^{128} different diffusion results, this is not realistic.

VI. CONCLUSION

In order to prevent the attacker from using double fault and single byte fault to attack the AES circuit, in this paper, we first analyze the flawed of the existing fault attack countermeasures, and then propose a new infection mechanism called Second Order Infection Mechanism. Furthermore, we use random numbers to make fault diffusion randomness, so as to resist single byte fault enumerate attack. The experimental results show the AES circuit which using the scheme proposed in this paper can resist fault attacks effectively including double error attacks and single byte error attacks effectively.

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The foundation of our Second Order Infection Mechanism is the injected fault no more than double. Hench, against our Second Order Infection Mechanism need to inject three same specified faults during one execution into the three encryption paths, respectively. As far as we know, it's very difficult to carry out such an attack.

REFERENCES

[1] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in CRYPTO, ser. Lecture Notes in Computer Science,B. S. K. Jr., Ed., vol. 1294. Springer, 1997, pp. 513–525.

[2] Daemen J, and Rijmen V, 'The design of Rijndael' (Springer, 2002)2 National Institute of Standards and Technology (NIST) 'Announcing the Advanced Encryption Standard (AES)'. Federal Information Processing Standards Publication, n.197, 26 November 2001.

[3] Nahid Farhady Ghalaty, Bilgiday Yuce, Mostafa Taha, et al. "Differential Fault Intensity Analysis," Proceeding of 11th Workshop on Fault Diagnosis and Tolerance in Cryptography, Busan, Korea,2014 : 49-58.

[4] Barenghi A, Breveglieri L, Koren I, et al, "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures," Proceedings of the IEEE, 2012, 100(11):3056-3076.

[5] Han Jun, Zeng Xiao-yang, Zhao Jia, "VLSI implementation of AES algorithm against differential power attack and differential fault attack," Journal on Communications, 2010,31(1): 20-29.

[6] M. Doulcier-Verdier, J-M.Dutertre, J. Fournier,et al, "A side-channel and fault-attack resistant AES circuit working on duplicated complemented values," IEEE International Solid-State Circuits Conference. IEEE, Feb. 2011:274–276.

[7] Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay, "Destroying Fault Invariant with Randomization A Countermeasure for AES Against Differential Fault Attacks," 2014 wordshop on Cryptographic Hardware and Embedded Systems, 2014:93-111.

[8] Joye M, Manet P, Rigaud J B, "Strengthening Hardware AES Implementations against Fault Attack," IET Information Security, 2007, 1(3):106-110.

[9] Lomne V, Roche T, Thillard A, "On the Need of Randomness in Fault Attack Countermeasures - Application to AES," 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE, 2012, pp. 85-94.

[10] Gierlichs B, Schmidt J M, Tunstall M, "Infective Computation and Dummy Rounds: Fault Protection for Block Ciphers without Check-before-Output," Progress in Cryptology – LATINCRYPT 2012. Springer Berlin Heidelberg, 2012:305-321.

[11] Battistello A, Giraud C, "Fault Analysis of Infective AES Computations, " In: Fischer, W., Schmidt, J.-M. (eds.) Fault Diagnosis and Tolerance in Cryptography, FDTC 2013, pp. 101–107.

[12] J. G. J. van Woudenberg, M. F. Witteman, and F. Menarini, "Practical optical fault injection on secure microcontrollers," in Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC), Sep. 2011, pp. 91–99.

[13] Bo Wang, Leibo Liu, Chenchen Deng,et al, "Against Double Fault Attacks: Injection Effort Model, Space and Time Randomization Based Countermeasures for Reconfigurable Array Architecture," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 6, JUNE 2016.

[14] Fournier J, Rigaud J B, Bouquet S, et al, "Design and characterisation of an AES chip embedding countermeasures," International Journal of Intelligent Engineering Informatics, 2011, 1(34):328-347.

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)