An Efficient Countermeasure against Fault Sensitivity Analysis Using Hybrid Parallel S-boxes

Qipeng Li, Fang Zhou, Ning Wu and Yasir

***Abstract*—Fault Sensitivity Analysis (FSA) is one of the fault attacks which can threaten the security of cryptographic module equipped with conventional countermeasure. In this paper, we present an efficient countermeasure against FSA based on mask strategy and hybrid parallel S-boxes structure. The masked AES circuit with the hybrid parallel S-boxes structure was proposed. The hybrid parallel S-boxes structure is composed of random selectors and four kinds of mask S-boxes. The proposed countermeasure can destroy the relationship between the fault sensitivity and the input Hamming weight, but also destroy collisions among the fault sensitivity characteristics of S-boxes. We conduct two kinds of FSA attacks against the AES circuit implemented on Xilinx Spartan FPGA, and the results show that FSA cannot threaten the security of the AES circuit with proposed countermeasure.**

***Index Terms*—Fault Sensitivity Analysis, Countermeasures, Hybrid Parallel S-boxes, Mask, AES**

I. Introduction

For the past few years, fault attacks (FAs) [1] have become a real threat to the security of cryptosystems. For symmetric key cryptography, Differential Fault Analysis (DFA) [2] is one of the most important FAs. DFA requires that the faulty ciphertexts must be valuable and have some relevance to the encryption key. It is not difficult to defend DFA. For example, the concurrent error detection techniques and Wave Dynamic Differential Logic (WDDL) [3] are both known to be effective countermeasures against DFA. However, Fault Sensitivity Analysis (FSA) [4] has broken the encryption module that is equipped with countermeasures against conventional FAs. Therefore, the study of countermeasure against FSA has a great significance for the security of cryptosystems.

FSA is a new kind of FAs which exploits the dependency between secret information and fault sensitivity. Li et al. [4] used the Hamming weight as fault sensitivity model and attacked AES circuits with Positive Polarity Reed-Muller Form (PPRM) [5] S-boxes. After that, collision FSA [6], an extension of the FSA, was proposed in 2011. Collision FSA combines the concept of fault sensitivity analysis and related collision attacks [7], and does not require any fault sensitivity model or analysis phase for key recovery. The literature [6] shows that this attack can successfully attack a variety of S-boxes. In addition, a variety of conventional countermeasures were also defeated by Collision FSA such as Masked AND-OR (MAO), WDDL and concurrent error detection.

At present, there are not many studies on FSA countermeasure. A countermeasure against FSA was proposed in [8], which can use the enable signal to eliminate the correlation between the secret data and the fault sensitivity. The enable signal is the key to this method, but it did not mentioned that how to generate the enable signal. Endo et al. [9] presented a countermeasure against FSA based on a configurable delay blocks (CDBs) and gave the method of generating enable signal. This countermeasure is actually a combination of CBD technology and Li's concept proposed in [8]. This strategy needs a configuration process in practical application. And it is not easy to generate and control the enable signal.

In order to resist FSA, an easy to implement countermeasure based on mask strategy and hybrid parallel S-boxes structure was proposed in this paper. The rest of this paper is organized as follows:

1) Two kinds of FSA, FSA based on the Hamming weight model and Collision FSA, were introduced briefly.
2) The overall structure of the masked AES circuit with hybrid parallel S-boxes was proposed. Then the design of hybrid parallel S-boxes based on four kinds of masked S-boxes was described in detail.
3) Two FSA attacks against the AES circuit implemented on FPGA were conducted to verify the ability of proposed countermeasure.
4) Give the conclusions of this paper.

II. The Principle of FSA

A. FSA based on the Hamming weight model

Fault sensitivity analysis (FSA) is a new fault attack using Fault Sensitivity (FS). FS, a new kind of side channel information, means the critical condition of clock frequency or supply voltage. The FS data change with secret keys, so the key can be obtained by measuring FS

Manuscript received June 30, 2017. This work was supported in part by the National Natural Science Foundation of China (61376025), the Fundamental Research Funds for the Central Universities (NS2017023) and the Natural Science Foundation of Jiangsu Province (BK20160806).

Q. Li is with College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China (e-mail: liqipeng0811@163.com).

F. Zhou is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China

N. Wu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China (e-mail: wunee@nuaa.edu.cn).

Yasir is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China.

data. The FS data can be measured by injecting a glitch clock into the circuit. The frequency of the glitch clock is a FS data where the first error occurred of the output ciphertext.

FSA based on the Hamming weight model is the primary fault sensitivity analysis. It requires that the FS data of each S-box depends on input values. In [4], the AES circuit with PPRM S-box was attacked and the secret key was successful recovered from 50 plaintexts.

B. Collision FSA

Collision FSA, a more powerful attack, is the combination of FSA and Correlation Collision Attack. This kind of attack uses the collision among the FS characteristics of S-boxes to recovery the key. Therefore, Collision FSA does not need any FS model, such as Hamming weight model. Let k_1 and k_2 be the input of S-box1 and S-box2, respectively. According to the concept of Correlation Collision Attack, when a collision between these two S-boxes occurs, $k_1 = k_2$.

We use the Collision FSA attack against the last round in the 128-bit AES to explain the principle of Collision FSA. Let *i* be a byte index ($0 \le i \le 15$). Let K_i^{10} and C_i be a tenth round sub-key and a ciphertext at the *i*th byte, respectively. Let *SR(i)* be a byte index after ShiftRows (SR). The input value of tenth round I_i can be calculated as

$$I_i^{10} = InvSbox(C_{SR(i)} \oplus K_{SR(i)}^{10}). \quad (1)$$

The FS data can be obtained from the distribution of faulty ciphertexts. Let FS be a function of the S-box input $FS(I_i^{10})$. The FS data of first two byte are

$$FS(I_1^{10}) = FS(InvSbox(C \oplus K_{SR(1)}^{10})), \quad (2)$$

$$\begin{aligned} FS(I_2^{10}) &= FS(InvSbox(C \oplus K_{SR(2)}^{10})) \\ &= FS(InvSbox(C \oplus K_{SR(1)}^{10} \oplus \Delta_K^{10})). \end{aligned} \quad (3)$$

The C is the correct ciphertext, and the Δ_K^{10} is the difference of two sub-keys at *i*th byte. We can use the hypothetical sub-key difference to rearrange $FS(I_2^{10})$ as

$$FS'(x) = FS(x \oplus \Delta_K'^{10}). \quad (4)$$

Then we can make sure that the hypothetical sub-key is correct if the distributions of the FS data from the two S-boxes match.

III. The Proposed Masked AES Circuit with Hybrid Parallel S-boxes Structure

A. Masked AES circuit design

AES encryption algorithm is a block cipher algorithm. The 128-bit input plaintext is divided into 4x4 state matrices. The elements of the state matrix are 8-bits of data, that is. According to the different size of the key: 128, 192, 256 bit, the state matrix is operated by 10, 12 or 14 rounds transformation respectively. Each round consists of SubBytes (SB), ShiftRows (SR), MixColumns (MC), and AddRoundKey (ARK). And there are only three transformations of SB, SR and ARK at last round. In this design, 128-bit key is used and there are 10 rounds transformation. The design of overall structure of mask AES circuit is shown in Fig. 1.

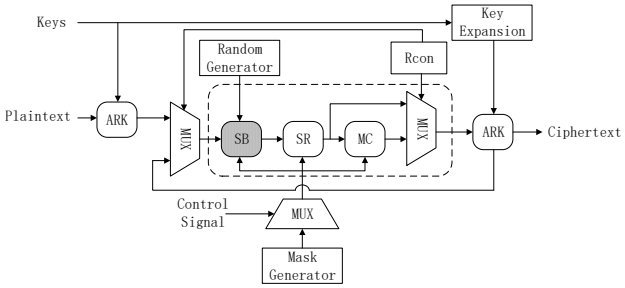

Fig. 1. The overall structure of masked AES circuit

The Key Expansion module generates round key that used in ARK. The Rcon records the round count and eliminates the symmetry. The random mask is generated from Mask Generator, and controlled by Control Signal. The special design of this circuit is SB module which uses a 32bit random from the Random Generator and has the hybrid parallel S-boxes structure inside.

Four kinds of masked S-boxes based on composite field were used in the design of SB module. For $GF((2^4)^2)$ and $GF(2^4)$ field, there are four basis methods, denoted as PP, PN, NP and NN, where the first one is for $GF((2^4)^2)$ field, and the second one for $GF(2^4)$ field. For each basis, we choose the optimal coefficient and the corresponding optimal root to design masked S-boxes among 256 combinations.

B. Hybrid parallel S-boxes structure design

In general, the AES circuit is designed to use only one kind of fixed S-box. For example, when designing an AES circuit with a 128 bit data, SB unit uses 16 parallel identical S-boxes to perform 16-byte SB operations for the state matrix, respectively. The combined logic circuit delay of this AES circuit is fixed when the input is fixed. The attacker can use the same plaintext to repeat the test, by injecting the fault clock to measure the fault sensitivity, and then crack the key. Mask strategy is a way to increase the complexity of fault sensitivity measurement. Add a different random mask to the circuit at each encryption, and the fault sensitivity of the circuit changes due to the input of the random mask, even though the input is fixed. But the mask strategy is not enough to resist the collision FSA. A more efficient countermeasure called hybrid parallel S-boxes structure is proposed base on four kinds of mask S-boxes in this paper.

The hybrid parallel S-box structure is designed for SB unit. The SB unit includes 4 PP mask S-boxes, 4 NN mask S-boxes, 4 PN mask S-boxes and 4 NP mask S-boxes. These 16 S-boxes are divided into 4 groups, and each group includes 4 kinds of S-boxes. In the traditional design, each byte of the state matrix corresponds to an S-box. In this design, each column of the state matrix corresponds to a set of S-boxes, and the correspondence between the state matrix and the S-box is shown in the Fig. 2.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

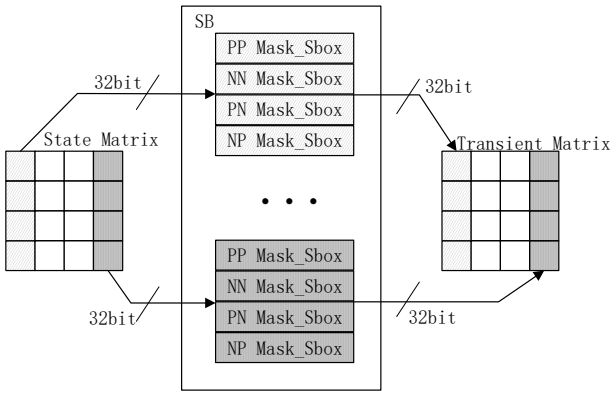

Fig. 2. The structure of SB module

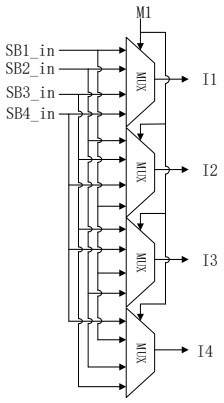

Fig. 4. The design of input random selector

Each column element of the state matrix enters a set of S-boxes and the result of the operation is sent to the corresponding position of the transient matrix. A random selection unit is designed at the input position of each group S-box. The input random selection unit can randomly select the input of 4 bytes so that each byte data can be entered into different S-boxes. In order to correct the function of the circuit, it is necessary to design a corresponding output selector in the output position of each group S-box. The output selector can assign the S-box output to the corresponding byte of the transient matrix. Taking the 4 bytes of the first column of the state matrix as an example, the SB unit is shown in Fig. 3.

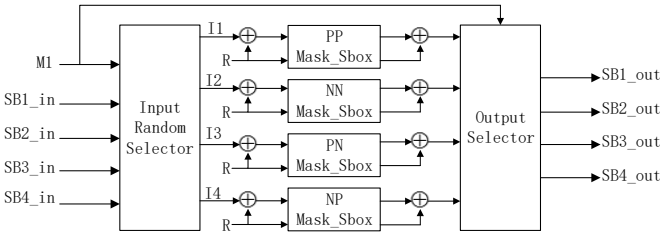

Fig. 3. The structure of hybrid parallel S-boxes

SB1_in, SB2_in, SB3_in, SB4_in is the 4-byte data of the first column of the state matrix, SB1_out, SB2_out, SB3_out, SB4_out is the 4-byte data of the first column of the transient matrix, M1 is a 2-bit random selection factor, R is an 8-bit random mask. The random number generator has been integrated in many devices, so we do not consider the problem of random number generation in this paper. The SB units requires an 8-bit random number and a 32-bit random number at runtime. The 8-bit random number is used as the random mask R. The 32-bit random number is decomposed into 4 bytes of data, and the lower two bits of each byte are taken as the random selection factor. Taking the 4 bytes of the first column of the state matrix as an example, the input random selector is shown in Fig. 4.

M1 is a 2-bit random selection factor generated by the high byte of the 32-bit random number. I1, I2, I3, I4 are the first column data of the state matrix after the scrambling sequence. The Output Selector is similar to the Input Random Selector and will not be described here. When the selection factor is changed, the two selectors make the state matrix data to enter the different S-boxes for SB operations and send them to the correct position of the transient matrix for subsequent operations. For example, M1=00, I1=SB1_in, first byte data enters the PP mask S-box for SB operation. M1=01, I4=SB1_in, first byte data enters the NP mask S-box for SB operation. But the calculation results are output from SB1_out in both cases.

IV. Experiments and Analysis

In this section, we conduct FSA attacks against the AES circuit with proposed countermeasure. The AES circuit was implemented on Xilinx Spartan6 FPGA. FSA attacks were conducted by using Post-Route simulation model which can calculate all the delay information of the implemented AES circuit.

A. The primary FSA attack against the AES circuit

The primary FSA refers to the FSA that requires Hamming weight model for key recovery in this section. We attack the last round of the AES circuit by injecting a glitch clock in the experiment. By gradually increasing the frequency of glitch clock until the ciphertext begins to go wrong, the clock frequency at this time is the fault sensitivity information. In order to ensure the authenticity of the experiment, the step size of the fault clock frequency change needs to be as small as possible and achievable in hardware. We set the fault clock in steps of 20ps which has been implemented in hardware [10].

Fig. 5 shows the primary FSA attack results of 4 sub-keys against proposed AES circuit using 1500 plaintexts. Each sub-figure corresponds to a sub-key. The key guess is represented on the horizontal axis. The correlation coefficient between the critical fault injection intensities and Hamming weight of the input is represented on the vertical axis. The actual correct key is marked by an ×, and the guess key result is marked by an +. As can be seen from the figure, the first four bytes of key recovery all failed. Therefore, the AES circuit, equipped with hybrid parallel S-box structure, has the ability to resist the primary FSA attack.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

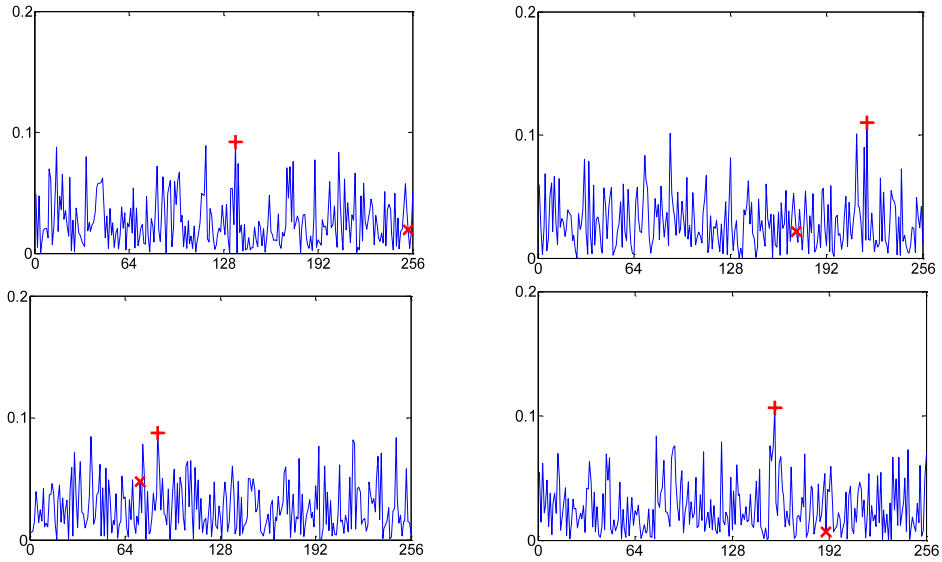

Fig. 5. The results of primary FSA attack

B. Collision FSA against the AES circuit

To verify the resistance of the proposed AES circuit to collision FSA, we try to detect a collision between first two byte of last round input. 256 groups of plaintext was been selected according to 256 differences in the first two sub-keys. The distribution of first two faulty ciphertexts was collected by executing the circuit 400×256 times. Then the distribution of the second byte was been rearranged according to the key differences. For 256 kinds of key differences, the result of detecting collision is shown in Fig. 6.

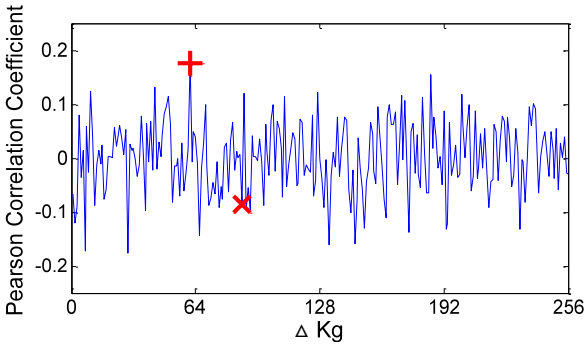

Fig. 6. Correlation vs. Key byte difference

The figure corresponds to first two sub-keys. The key difference guess is represented on the horizontal axis. The correlation coefficient between the first byte error ciphertext distribution and the second byte error ciphertext distribution is represented on the vertical axis. The actual correct key difference is marked by an ×, and the guess key difference result is marked by an +. As can be seen from the figure, the difference of two key byte cannot be measured at 400×256 execution. The failing attack experiments have been also shown that it cannot recover the difference between other key bytes. Therefore, the hybrid parallel S-boxes structure is an efficient countermeasure against collision FSA.

V. Conclusions

In this paper, we have presented an efficient countermeasure against Fault Sensitivity Analysis (FSA) based on mask strategy and hybrid parallel S-box structure. The results of FSA against the AES circuit implemented on FPGA show that the primary FSA attack with 1500 plaintexts and collision FSA attack with 400×256 executions cannot threaten the security of the AES circuit with proposed countermeasure.

References

[1] D. Boneh, R. DeMillo, and R. Lipton, "On the importance of checking cryptographic protocols for faults," in *EUROCRYPT '97*, LNCS 1233, pp. 37–51, 1997.

[2] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *CRYPTO '97*, LNCS 1294, pp. 513–525, 1997.

[3] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *DATE 2004*, vol. 1, pp. 246– 251, 2004.

[4] Y. Li, K. Sakiyama, and S. Gomisawa, et al, "Fault sensitivity analysis," in *CHES 2010*, LNCS 6225, pp. 320–334, 2010.

[5] S. Morioka and A. Satoh, "An optimized S-box circuit architecture for low power AES design," in *CHES 2002*, LNCS 2523, pp. 172-186, 2002

[6] A. Moradi, O. Mischke and C. Paar, et al, "On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting," *International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 6917, pp. 292-311, 2011.

[7] A. Moradi, O. Mischke and T. Eisenbarth, "Correlation-enhanced power analysis collision attack," in *CHES 2010*, LNCS 6225, pp. 125-139, 2010.

[8] Y. Li, K. Ohta and K. Sakiyama, "Toward effective countermeasures against an improved fault sensitivity analysis," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E95–A, no. 1, pp. 234–241, 2012.

[9] S. Endo, Y. Li and N. Homma, et al, "An Efficient Countermeasure against Fault Sensitivity Analysis Using Configurable Delay Blocks," *The Workshop on Fault Diagnosis and Tolerance in Cryptography IEEE Computer Society*, vol. 7024, pp. 95-102, 2012.

[10] S. Endo, T. Sugawara, and N. Homma, et al, "An on-chip glitch-clock generator for testing fault injection attacks," *Journal of Cryptographic Engineering*, *1*(4), 265-270, 2011.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)