applied sciences

MDPI

Article

Research on a Random Mask Infection Countermeasure against Double Fault Attacks

Weixiang Wu [1,2], Xusen Wan [2], Jinbao Zhang [2,*] and Shi Cheng [2]

[1] Xinglin College, Nantong University, Nantong 226236, China; wwx@ntu.edu.cn
[2] School of Information Science and Technology, Nantong University, Nantong 226019, China; 2110310021@stmail.ntu.edu.cn (X.W.); chenshi@ntu.edu.cn (S.C.)
* Correspondence: kingbao@ntu.edu.cn; Tel.: +86-1361-5201-360

Abstract: The infection countermeasure, in which the main idea is to prevent adversaries from exploiting faulty ciphertexts to break the key by spreading the induced fault, is a very effective countermeasure against fault attacks. However, most existing infection countermeasures struggle to defend against double-fault attacks effectively due to the single-fault assumption. By analyzing the principle of infection mechanism and adding different random Boolean masks in the two encryption paths, this paper proposes a measure called a random mask infection countermeasure to defend against double-fault attacks. In addition, the multiplication mask is used to randomize the fault diffusion to further resist single-byte fault attacks. The experimental results indicate that the random mask infection countermeasure proposed can perform fault diffusion effectively when the cryptographic circuit suffers double-fault attacks, and the fault diffusion shows randomness, and can effectively defend against these fault attacks.

Keywords: AES; infection countermeasure; mask; fault attacks

Citation: Wu, W.; Wan, X.; Zhang, J.; Cheng, S. Research on a Random Mask Infection Countermeasure against Double Fault Attacks. *Appl. Sci.* **2023**, *13*, 12530. https://doi.org/10.3390/app132212530

Academic Editor: Nuno Silva

Received: 4 October 2023
Revised: 3 November 2023
Accepted: 15 November 2023
Published: 20 November 2023

1. Introduction

As an international common encryption algorithm, block cipher has been widely used in security systems such as IoT, aerospace, medical, transportation, finance, and identity verification [1–5]. Since the proposal of the advanced encryption standard (AES), as an important part of block cipher, many attack methods have emerged against its hardware and software [1,5–13], and, among these attack methods, fault attacks have the advantages of high attack capability and low time complexity, which pose a great threat to the security of the AES [5,11–13].

Until now, lots of countermeasures have been proposed to resist fault attacks, which mainly fall into two categories: one type of approach is the use of fault detection [14–20], which detects the occurrence of faults by repeating the computation of some operation, module, or the entire algorithm utilizing a multiplexing technique [15,18,19], with the disadvantage that the comparison phase itself is vulnerable to new attacks. This is because the detection locations are associated with the data being processed [21]. Another type of approach is the use of fault infection [22–24]. Since the infection strategy does not need to utilize the detection locations, it effectively avoids the potential threat of being attacked due to the presence of detection locations. It spreads the fault by destroying the invariance of the fault propagation, and, thus, making the faulty ciphertext unavailable [23]. However, Lomne et al. has proved that the infection mechanism alone is not sufficient to resist fault injection attacks, and that the idea of randomization should be introduced on the basis of the infection mechanism [24]. On the basis of infection mechanism, the literature [25] advocates the method of randomly adding virtual rounds transformation and redundant calculation to resist fault attacks. However, the paper [26] pointed out that the virtual round transformation infection mechanism proposed in the literature [25] was not completely secure, and successfully attacked it. And the paper [21] further pointed out that the infection

mechanism in [25] was deficient due to the same unknown mask being used in the infection process and improved on [25]. In 2019, by utilizing the encryption/decryption circuit, [22] proposed a countermeasure against fault attack called random infection mechanism.

Since most of these existing countermeasures against fault attacks assume a single-fault condition, i.e., the assumption that only one of the two concurrent circuits employed (e.g., two identical cryptographic circuits) will be injected with a fault [16,23], thus, it is difficult to defend against double-fault attacks. Although it is difficult to inject the same fault for both encryption or decryption paths (in other words, it is difficult to perform a double-fault attack), with the development of fault injection accuracy, it has become possible to inject double faults [27]. Fortunately, researchers have recognized the threat of double-fault attacks and have discussed them to some extent [22,24,28,29]. In [24], when analyzing existing methods against fault attacks, the possibility of double-fault attacks on cryptographic circuits is clearly stated, but no specific countermeasure is given. In 2016, Wang et al. first systematically analyzed the problem of double-fault attacks in the fault injection phase, constructed a fault injection model, and analyzed how to reduce the difficulty of double-fault injection in the fault injection phase and discussed how to effectively defend against double-fault attacks [28]. Immediately afterward, Zhang et al. proposed a fault propagation method using encrypted/decrypted circuits to resist double-fault attacks and analyzed the fault depth with good results [22]. In [29], the authors proposed the concept of a second-order infection mechanism to counter the threat of double-fault attacks by constructing a three-strip cryptographic circuit, which increases the circuit area by approximately half compared to the traditional infection mechanism due to the use of a three-strip cryptographic circuit and the risk of higher fault (e.g., triple-fault) attacks.

In view of the serious threat of fault attacks to cryptographic circuits, and the shortcomings of existing infection mechanisms against double-fault attacks, and the fact that the infection mechanism has a single diffusion mode and cannot resist single-byte faults, this paper proposes an infection countermeasure based on random masks. Furthermore, to defend against possible single-byte fault attack, a random multiplication mask also is proposed to be used in the fault diffusion to achieve the purpose of randomizing the fault diffusion. By randomly selecting a set of plaintext and key, and conducting multiple experiments, the experimental results show that when a fault is induced, even if the induced fault is the same each time, the output ciphertext of the AES circuit designed is different, and the ciphertext exhibits a certain degree of randomness; thus, it shows that our proposed countermeasure has good performance in defending against fault attacks.

Compared to existing related works, our research has the following contributions:

(1) A called random mask infection countermeasure is proposed. By adding a random Boolean mask in the encryption process, the AES circuit which applied the random mask infection countermeasure can resist double-fault attacks;
(2) The countermeasure proposed can also resist single-byte exhaustive fault attacks. By using a random multiplication mask to achieve randomization of fault diffusion, the adversaries are, thus, unable to effectively retrieve the keys through exhaustive methods.

The rest of this paper is organized as follows. In Section 2, a brief introduction to the fault attack countermeasure as well as the principle and deficiency of existing infection countermeasures are presented. Section 3 provides the specific principles and design steps of the proposed countermeasure. Experimental analysis is given in Section 4. Finally, Section 5 offers the conclusion of this paper.

2. Preliminaries

2.1. Previous Fault Attack Countermeasures on the AES

Here, we briefly summarize the existing fault attack countermeasures against the AES.

(1) *Countermeasures against differential-fault attack(DFA)*. These use redundant calculations to detect whether a fault has occurred, such as [15,18,19]. Due to the comparison

phase, there is the disadvantage of vulnerability to new attacks, and other countermeasures by infecting faults were presented, for example [23–26]. Although these countermeasures can effectively defend against DFA, due to them assuming that there is only one single encryption path, they are unable to defend against double-fault attacks which are based on two encryption paths;

(2) *Countermeasures against double-fault attack*. In 2016, Wang et al. first analyzed how to reduce the difficulty of double-fault injection in the fault injection phase and discussed how to effectively defend against double-fault attacks [28]. Soon afterwards, Zhang et al. proposed the concept of a second-order infection mechanism, which utilized three encryption paths [29]. Due to the use of three encryption paths, it will undoubtedly significantly increase the area consumption of the circuit, and there is also a risk of being attacked by three faults. In 2019, Zhang et al. further designed encryption/decryption circuits to defend against double-fault attacks [22], for which the designed circuit adopts an entire encryption path and a partial decryption path.

2.2. Principle and Deficiency of Existing Infection Countermeasures

To perform fault attack on AES cryptographic circuits, the attacker first makes the cryptographic device perform some faulty operations by inducing faults (the precision and size of the induced faults are controllable) and, then, retrieves the key according to the faulty ciphertexts. Due to the diffusion mode of the AES being known, the adversary can accurately infer the fault spread according to the characteristics of the cryptographic algorithm, and, on this basis, derive the equation relationships between the faulty ciphertexts, correct ciphertexts, and the induced fault, and, finally, obtain the key by exhaustive search under these equation relationships.

The infection countermeasure defends against fault attacks by amplifying the impact of the injection fault so that the fault information inside the cryptographic circuit spreads over a wider area rather than simply following the flow of the AES algorithm itself, thus making it impossible for an attacker to crack the key through the faulty ciphertexts. Figure 1 depicts the principle of the existing infection countermeasure.

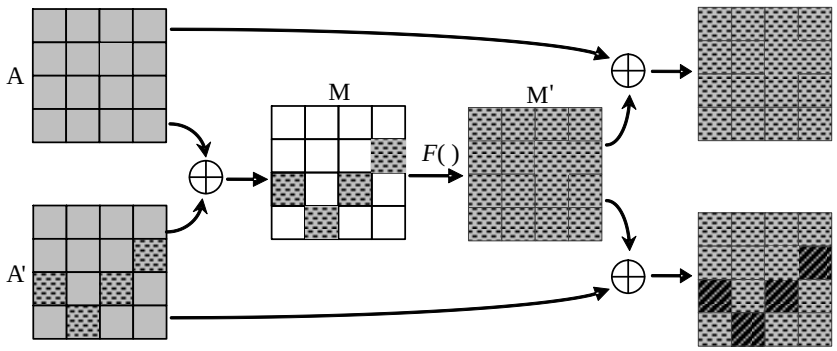

Figure 1. Principle of the existing infection countermeasure.

In Figure 1, the state matrices A and A′ represent the intermediate states of the normal encrypted data and the encrypted data after inducing fault at the same moment in the AES circuit, respectively.

The principle of resisting a single-fault injection attack based on the countermeasure in Figure 1 is as follows: two identical plaintexts P are encrypted simultaneously via two paths, and, when a certain encryption operation is executed, the difference M is calculated between matrices A and A′. And, then, utilizing the diffusion function F (F satisfies $F(0) = 0$) to diffuse the difference M to obtain M′, and, then, XOR A and A′ with M′, respectively, to achieve the purpose of diffusion fault, thus preventing the attacker from retrieving the key based on the output faulty ciphertext and correct ciphertext.

In [30], a general diffusion function F as shown in Expression (1) was presented, where M_{ij} is denoted as the elements of the i-th row and j-th column of a $4 \times 4-$byte matrix.

$$F : M'_{ij} = \sum_{n=0}^{3} M_{in} \oplus \sum_{m=0}^{3} M_{mj} \quad (1)$$

A fault will spread to its row and column by the diffusion function F (as shown in Figure 1). However, there are two shortcomings in the above infection mechanisms.

The infection countermeasure above cannot effectively defend against double-fault attacks.

Double-fault attack refers to inducing a fault at the same location in each of the two paths of the circuit in Figure 1 to avoid the detection of mismatched intermediate values in the two paths.

For example, after simultaneous injection of the same fault on both encryption paths of the cryptographic device, the state matrices A = A′, so that the difference between A and A′ is M =A⊕A′ = 0. And, since the diffusion function satisfies $F(0) = 0$, the value M′ after diffusion by $F()$ is still 0. Then, A and A′ are XOR with M′, respectively. In the final output results, neither S nor S′ are changed. That is, the faults induced could not be diffused by $F()$, and the defence against double-fault attacks by the infection mechanism in Figure 1 cannot be rendered by the infection countermeasure described in Figure 1.

The diffusion mode of $F()$ is fixed and is not resistant to fault attacks which are based on the byte fault model.

When there is a single-byte fault attack, if $F()$ is fixed, there will be only 256 (that is 2^8) different diffusion cases after diffusion by $F()$. And, then, the attacker can retrieve the key according to the idea of DFA by enumerating and analyzing these 256 possible diffusion results.

To address the problems above, what is called a random mask infection countermeasure is presented to defend against fault attacks. By using random Boolean masks and a random multiplication mask in the two encryption paths to achieve randomization of fault propagation, fault attacks such as double-fault attacks as well as single-byte fault attacks are, finally, prevented.

3. Design of a Random Mask Infection Countermeasure

The specific ideas and steps of designing the proposed random mask infection countermeasure are shown in Figure 2.

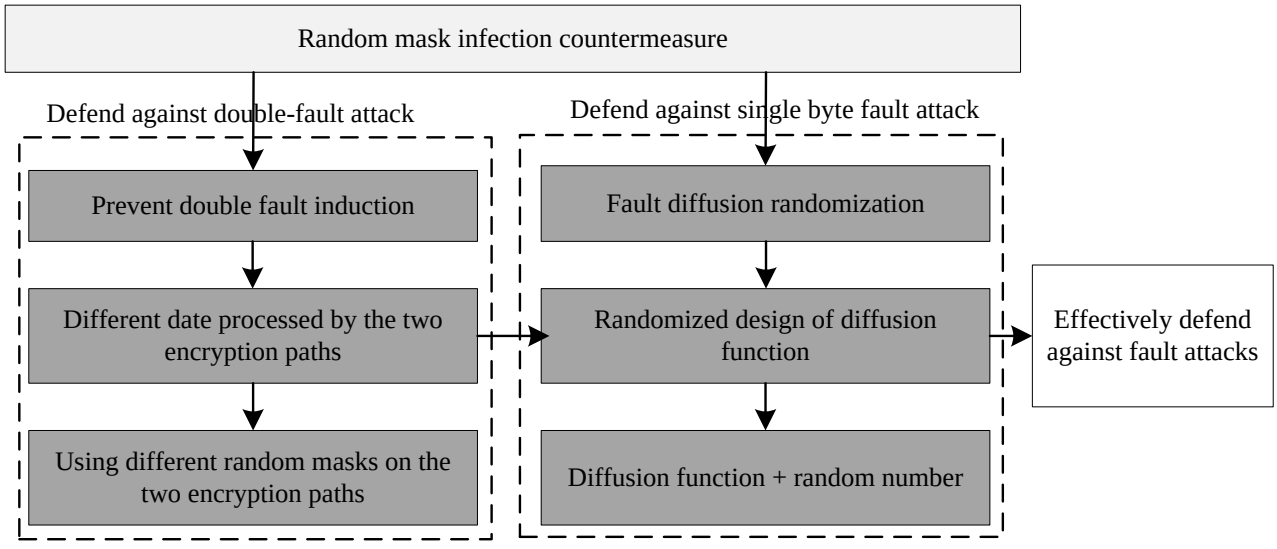

Figure 2. The design ideas and steps of the proposed random mask infection countermeasure.

As described in Figure 2, the designed countermeasure must meet two conditions, that is, it can defend against both double-fault attacks and single-byte attacks. For defending against double-fault attacks, the main idea is to prevent the attacker from inducing double faults, making it impossible to obtain effective fault information. This requires that, in the event of a fault occurring, the two encryption paths handle different data. Thus, different random masks are used on the two encryption paths. For defending against single-byte fault attacks, the main idea is to randomize the spread of faults, preventing the attacker from easily cracking the key through exhaustive methods. This is mainly achieved by designing randomized diffusion functions. Finally, by combining the two ideas above, the random mask infection countermeasure is proposed.

3.1. Prevent Double-Fault Induction

To conduct a double-fault attack, the adversary must first know the difference between the data in the two encryption paths when injecting faults, and, then, inject the corresponding faults based on the difference, so as to offset the difference between the two and bypass the infection mechanism.

As pointed out in [31], the mask cannot protect against fault attacks. However, the purpose of using the random mask here is to render attackers unable to induce double faults. Based on the idea of information hiding, this paper proposes the introduction of different random masks (N_1 and N_2) in each of the two paths of data encryption, so that the adversary cannot ascertain the difference between the data in the two paths at any moment. Thus, the attacker cannot inject the same fault into the two paths simultaneously and avoid the influence of the infection countermeasure; the specific structure is shown in Figure 2.

As shown in Figure 3, the principle of resisting double-fault attacks is: the plaintext P is encrypted by the encryption paths ① and ②, respectively. The random number generator generates two different random numbers N_1 and N_2 (N_1 and $N_2 \neq 0$), which are XOR with P, respectively, to obtain $P_1 = P \oplus N_1$ and $P_2 = P \oplus N_2$. The register unit stores the data for unmasking in advance, and outputs the random numbers N_1' and N_1' corresponding to N_1 and N_2 for unmasking. The infection countermeasure unit implements a countermeasure of the diffusion of fault data and infects the diffused fault data back to the original encryption path (seeing Figure 1).

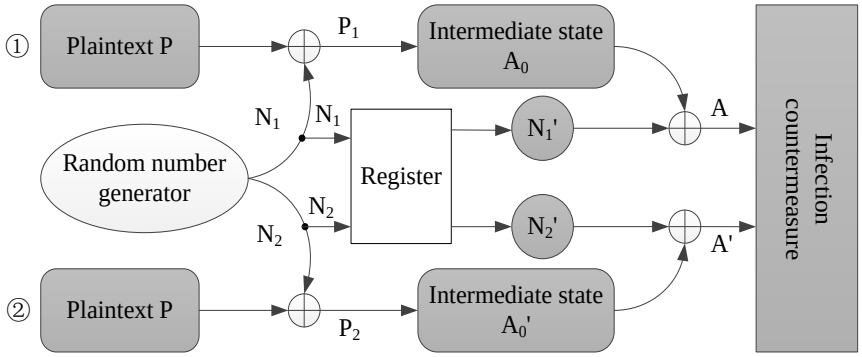

Figure 3. Structure of the random mask infection.

The data are encrypted in several rounds to obtain the intermediate states A_0 and A_0'. Because of the using of random mask N_1 and N_2, it can guarantee that the values of A_0 and A_0' are two completely different values and the difference between them is randomly varying. The process of demasking is conducted by the XOR operation $A_0 \oplus N_1'$ and $A_0' \oplus N_2'$. Through the above processing, it is impossible to synchronously inject faults in A_0 and A_0' so that $(A_0 \oplus N_1') \oplus (A_0' \oplus N_2') = 0$. Because A_0 and A_0' are two completely different values and the difference between them varies randomly, the adversary can not obtain the exact difference between them.

3.2. Randomized Design of the Diffusion Function

In the infection countermeasure unit, a randomized diffusion function is designed to close the loopholes in the infection mechanism mentioned in Section 2.2 (that is, it cannot be resistant to fault attacks which are based on the byte fault model). The specific expression of the designed diffusion function F is shown in (2).

$$F:\begin{cases} \mathrm{M}'_{il} = \sum_{n=0}^{3} \mathrm{M}_{in} \oplus \sum_{m=0}^{3} \mathrm{M}_{mj} \\ \theta = \mathrm{N}_3 \,.\, \mathrm{M}'_{il} \end{cases} \tag{2}$$

where N_3 is the 4×4 random number matrix generated by the random number generator and is non-zero. F is the result after diffusion, also a 4×4 matrix. When M = 0, the value of M'_{ij} and $\theta = \mathrm{N}_3$. M'_{ij} equals zero and satisfies the property that the diffusion function $F(0) = 0$. Once $\mathrm{M} \neq 0$, then the value of M'_{ij} and F will not be equal to zero, and θ has randomness due to N_3 being randomly varying.

Based on Figure 3 and Equation (2), what is called a random mask infection countermeasure is designed, which can both defend against double-fault attacks and avoid the disadvantage of the simplification of diffusion function mapping. In Algorithm 1, the algorithm description of this strategy is shown.

Algorithm 1: Random Mask Infection Countermeasure

Input: plaintexts P, random masks N_1, M_2, and M_3 (M_1, M_2, and $\mathrm{M}_3 \neq 0$)
Output: ciphertext C.

(1) $\mathrm{P}_1 \leftarrow \mathrm{P} \oplus \mathrm{N}_1$, $\mathrm{P}_2 \leftarrow \mathrm{P} \oplus \mathrm{N}_2$;
(2) $\mathrm{A}_0 \leftarrow \mathrm{Cipher}(\mathrm{P}_1)$, $\mathrm{A}_0' \leftarrow \mathrm{Cipher}(\mathrm{P}_2)$;
(3) $\mathrm{A} \leftarrow (\mathrm{A}_0 \oplus \mathrm{N}_1')$, $\mathrm{A}' \leftarrow (\mathrm{A}_0' \oplus \mathrm{N}_2')$;
(4) $\mathrm{M} \leftarrow \mathrm{A} \oplus \mathrm{A}'$;
(5) $\theta \leftarrow \mathrm{N}_3 \cdot F(\mathrm{M})$;
(6) $\mathrm{A} \leftarrow \mathrm{A} \oplus \theta$, $\mathrm{A}' \leftarrow \mathrm{A}' \oplus \theta$;
(7) $\mathrm{C} \leftarrow \mathrm{Cipher}(\mathrm{A})$.

The flow of the entire algorithm is: the initial plaintexts of the two encryption paths are all P. We perform masking operations on P, i.e., $\mathrm{P}_1 \leftarrow \mathrm{P} \oplus \mathrm{N}_1$, $\mathrm{P}_2 \leftarrow \mathrm{P} \oplus \mathrm{N}_2$. Then, we encrypt the mask-transformed P_1 and P_2, and, after several encryption operations, the intermediate states obtained are A_0 and A_0'. Subsequently, we calculate $\mathrm{A} \leftarrow (\mathrm{A}_0 \oplus \mathrm{N}_1')$, $\mathrm{A}' \leftarrow (\mathrm{A}_0' \oplus \mathrm{N}_2')$, and $\mathrm{M} \leftarrow \mathrm{A} \oplus \mathrm{A}'$, where N_1' and N_2' are the random numbers used for demasking corresponding to the random numbers N_1 and N_2, respectively. Then, we detect whether M is zero and if $\mathrm{M} \neq 0$, it means that at least one of the two paths possess a fault in the encryption process, followed by the diffusion function $\theta \leftarrow \mathrm{N}_3 \cdot F(\Delta)$ which spreads the fault to other bytes. Finally, after mixing the diffused faults into A and A', i.e., $\mathrm{A} \leftarrow \mathrm{A} \oplus \theta$ and $\mathrm{A}' \leftarrow \mathrm{A}' \oplus \theta$, we continue to perform subsequent encryption operations and output the random faulty ciphertext C. If M = 0, then the random mask infection countermeasure will not be effective, and will continue to complete subsequent encryption operations, and output the correct ciphertext C. Finally, the correct ciphertext C will be output.

4. Experimental Analysis

4.1. Verification of Against Double-Fault Attacks

The double-fault attacks assumed in this paper refer to a class of double-fault attacks that force single-bit data at the same location in two encryption paths to change to 0 or 1. To prove the effectiveness of our proposed random mask infection countermeasure to defend against double-fault attacks, the AES cryptographic circuit incorporating this countermeasure is simulated and attacked. In this paper, the AES is utilized with count mode. Assuming the location of the fault injection occurs after the $R-1$ (R represents the number of rounds of the AES, and for AES-128, R = 10) round of key-add operation and

before the last round (that is, the $R-$th round) of subbytes (SB) operation, the schematic diagram of the double-fault attack is described in Figure 4.

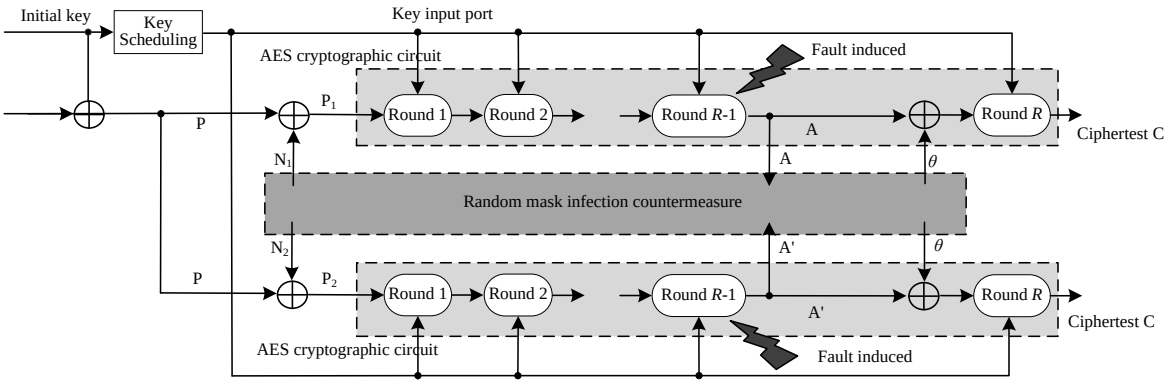

Figure 4. Diagram of double-fault attacks.

In Figure 4, the random mask infection countermeasure unit implements the mask, unmask, and random diffusion of fault data, and infects the diffused fault data back to the original encryption path (see Algorithm 1 for specific description). Here, we take AES$-$128 as an example to discuss, then Round $R-1$ is the 9th round of the AES$-$128 cryptographic circuit. A and A′ represent the state matrix after the completion of the 9th round of encryption, N_1 and N_2 are random numbers used for the mask, and $N_1 \neq N_2$. Ciphertext C is the final output ciphertext of the circuit.

Select an arbitrary pair of plaintext P and key K as follows:

P: f3 01 a6 8a 9e 9f fa 50 84 45 81 d9 e2 90 d8 18

K: 09 36 3c fc c8 3d 9c a3 7f 42 7e da da b2 c2 82

The random masks N_1 and N_2 are as follows:

N_1: d0 71 6a 29 02 14 4e 9e d5 43 71 63 ec 8f af 07

N_2: d8 bf 52 38 f8 54 80 65 f4 f6 e1 ec e8 72 49 a5

When the 9th round of encryption operation ends, the intermediate states A and A′ at the end of the 9th round of encryption are as follows:

A: 79 71 3f 8b 57 e3 4f 9d 93 c8 66 6b 06 4e 2e 76

A′: 7b 1a 81 47 5f 45 51 d5 f9 e0 e8 23 a4 cc 50 01

The random numbers N_1' and N_2' (N_1' and N_2' correspond to N_1 and N_2, respectively) corresponding to A and A′ demasking are as follows:

N_1': ba bc 39 3f 9e 37 96 ac d2 60 d5 f4 b7 57 1b a5

N_2': b8 d7 87 f3 96 91 88 e4 b8 48 5b bc 15 d5 65 d2

When there is a fault induced in the first byte of A and A′ at the same time (this means that there is a double-fault attack), then, the values of A and A′ become as follow:

A: 70 71 3f 8b 57 e3 4f 9d 93 c8 66 6b 06 4e 2e 76

A′: 70 1a 81 47 5f 45 51 d5 f9 e0 e8 23 a4 cc 50 01

The difference $M = (A_0 \oplus N_1') \oplus (A_0' \oplus N_2')$ at this time is:

M: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

From the above experiments, it can be seen that when the same fault is injected, $M \neq 0$, the adversary cannot bypass our countermeasure, which indicates that the random mask infection mechanism designed in this paper can defend against double-fault attacks.

4.2. Verification and Analysis of the Randomness of the Random Mask Infection Countermeasure

When there is a single-byte fault attack, taking only the first byte of the difference M between A and A′ is not zero, for example. To verify the randomness of our random mask infection countermeasure, the same plaintext needs to be encrypted multiple times (here, the corresponding key is kept constant) to check whether the final output ciphertext C exhibits randomness when the input M of the diffusion function is the same. The specific results of the experiments are as follow in Table 1.

Table 1. Diffusion simulation results.

Number	Diffusion Results
	P: f3 01 a6 8a 9e 9f fa 50 84 45 81 d9 e2 90 d8 18 K: 09 36 3c fc c8 3d 9c a3 7f 42 7e da da b2 c2 82 M: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1	f4 5b 3b 13 63 a1 c2 c0 63 b5 d8 f6 63 a0 c2 e3
2	22 ab 41 b0 18 e5 af 9d 30 4e 32 67 02 cd fe 43
3	ae 3b 0e 64 c3 67 30 d5 81 66 74 9a c6 36 42 6b
4	08 88 e4 23 d4 89 3c 67 d4 89 3c 67 d4 89 3c 67
5	86 1e 3c 78 f0 91 53 a5 3d 7a 46 8c 69 d2 d5 db
6	c7 ff 8e 6f de cd eb a7 3f 7e fc 89 63 c6 fd 8b
7	7d e2 36 8c 95 9d 32 67 95 9d 32 66 95 9d 32 ce
8	ed ab 27 4e 9c 49 92 55 aa 25 4a 94 59 b2 15 2a
9	54 a8 21 42 84 79 f2 95 5b b6 1d 3a 74 e8 a1 33
10	66 cc e9 a3 37 6e dc c9 e3 b7 1f 3e 7c f8 80 74
11	e6 bd 0b 16 2c 58 b0 11 22 34 61 c2 f5 9b 47 8e
12	6d da c5 fb 87 7f fe 8d 6b d6 dd cb e7 bf 0f 1e
13	3c 78 f0 91 53 a6 3d 7a f4 99 43 86 7d fa 85 7b
14	f6 9d 4b 96 5d ba 05 0a 14 28 50 a0 31 62 c4 f9
15	83 77 ee ad 2b 56 ac 29 52 a4 39 72 e4 b9 03 06
16	18 30 60 f1 93 57 ae 2d 5a b4 19 32 64 c8 e1 b3
17	17 2e 5c b8 01 02 71 e2 b5 1b 34 68 d0 cf 5e 09
18	12 24 90 51 a2 35 6a d4 d9 c3 f7 9f 4f 9e 4d 9a
19	45 8a 65 ca e5 bb 07 0e 1c 38 70 e0 b1 13 26 4c
20	98 41 82 75 ea a5 3b 76 ec a9 23 46 8c 69 d2 c7

In Table 1, P and K are the input plaintext and the encryption key, respectively. M is the difference between the intermediate states A and A', and the serial numbers 1 to 20 are the output ciphertext C encrypted 20 times for plaintext P.

From Table 1, although P, K, and M are the same, the output ciphertext C for each encryption is different. In addition, the ciphertext shows uncertainty and randomness, which indicates that the diffusion effect of our countermeasure is random.

In Section 2.2 we have explained that the complexity of the diffusion function determined by the mapping relationship is 2^8 when only a single-byte fault occurs, and an attacker can exhaust the 2^8 cases to break the key. Since the countermeasure proposed is a mapping relationship constructed by a random diffusion function with randomness, its complexity is 2^{128} when a single-byte fault occurs, and it is unrealistic to crack the key by exhaustively enumerating 2^{128} cases.

4.3. Comparison of Existing Countermeasures

In Table 2, we briefly compare some countermeasures against fault attacks mentioned in this paper and indicate the possible threats that they may suffer from power attacks, as well as fault attacks, such as double-fault attacks, single-byte fault attacks, and higher fault attacks.

Papers [11,14,23,25] only assume single-fault attacks and, thus, neither can defend against double-fault attacks. Paper [11] improves the performance of AES circuits in defending against byte fault attacks by redesigning matrix for mix-column module, and successfully increases the exhaustive complexity of byte fault attacks to 2^{116}. In [14], four

countermeasures were proposed, and they were all based on Hamming code and parity bits. These carry the risk of being exploited by power analysis (PA) attacks. In [25], the authors utilized randomly added virtual round transformation and redundant computation based on the infection countermeasure to resist fault attacks, but there is a risk of suffering from single-byte fault attacks because the same unknown mask is used in the infection process. The second-order infection mechanism proposed in [29], although it can defend against double-fault attacks, is subject to the risk of triple-fault attacks (three cryptographic circuits injecting the same fault at the same time) because of the use of three cryptographic circuits.

Table 2. Comparison of existing countermeasures.

Countermeasures	Whether They Can Defend against Double-Fault Attacks (DFA)	Power Attack Threats	Fault Attack Threats
[11]	Y/N	N	double-fault attack
[14]	Y/N	Y	double-fault attack
[23]	Y/N	N	double-fault attack, single-byte fault attack
[25]	Y/N	N	double-fault attack, single-byte fault attack
[29]	Y/Y	N	triple-fault attack
This article	Y/Y	N	

5. Conclusions

This paper addresses the shortcomings of the traditional infection countermeasure, i.e., it cannot resist double-fault attacks and the diffusion function of single mapping can be broken by exhaustive enumeration. Based on the idea of randomization, we propose a random mask infection countermeasure, which resists double-fault attacks by adding a random Boolean mask to the encryption process and using a random multiplication mask to achieve randomization of fault diffusion to further resist single-byte exhaustive fault attacks. Through theoretical analysis and experimental verification, it is shown that the AES cryptographic circuit designed using the countermeasure proposed can effectively resist fault attacks, including double-fault attacks.

When considering some other block ciphers, such as 3DES, Blowfish, IDEA, etc., due to their similar structure to the AES, the countermeasure designed in this paper can still be applied in them. The main issue to consider is how to efficiently remove the random mask after encryption. In addition to this, this paper mainly considers the defense performance of the designed countermeasure against fault attacks, without considering the area consumption. How to further improve defense performance while reducing area consumption is also the research direction of our future.

Author Contributions: Conceptualization, W.W. and J.Z.; methodology, X.W. and J.Z.; software, W.W. and X.W.; validation, W.W., J.Z. and S.C.; formal analysis, X.W., J.Z. and W.W.; resources, W.W. and J.Z.; data curation, W.W.; writing—original draft preparation, W.W. and J.Z.; writing—review and editing, X.W. and J.Z.; supervision, S.C. and J.Z.; project administration, W.W. and J.Z.; funding acquisition, J.Z. and W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Research in Colleges of Jiangsu Province (grant numbers 21KJB520040 and 22KJD520006) and the Basic Science Research Project of Nantong (grant number JC2020144).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The author would like to thank the anonymous reviewers for their valuable suggestions and comments that improved the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Subramanian, S.; Mozaffari-Kermani, M.; Azarderakhsh, R.; Nojoumian, M. Reliable hardware architectures for cryptographic block ciphers LED and HIGHT. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2017**, *36*, 1750–1758. [CrossRef]
2. Zhang, J.; Wang, J.; Bin, G.; Li, J. An Efficient Differential Fault Attack against SIMON Key Schedule. *J. Inf. Secur. Appl.* **2022**, *66*, 103155. [CrossRef]
3. Li, L.; Fang, J.; Jiang, J.; Gan, L.; Zheng, W.; Fu, H.; Yang, G. Efficient AES implementation on Sunway Taihu Light supercomputer: A systematic approach. *J. Parallel Distrib. Comput.* **2020**, *138*, 178–189. [CrossRef]
4. Kumar, T.M.; Balmuri, K.R.; Marchewka, A.; Bidare Divakarachari, P.; Konda, S. Implementation of Speed-Efficient Key-Scheduling Process of AES for Secure Storage and Transmission of Data. *Sensors* **2021**, *21*, 8347. [CrossRef] [PubMed]
5. Sheikhpour, S.; Ko, S.-B.; Mahani, A. A low cost fault attack resilient AES for IoT applications. *Microelectron. Reliab.* **2021**, *123*, 114202. [CrossRef]
6. Sugawara, T.; Shoji, N.; Sakiyama, K.; Matsuda, K.; Miura, N.; Nagata, M. Side-channel leakage from sensor-based countermeasures against fault injection attack. *Microelectron. J.* **2019**, *90*, 63–71. [CrossRef]
7. Gérault, D.; Lafourcade, P.; Minier, M.; Solnon, C. Revisiting AES related-key differential attacks with constraint programming. *Inf. Process. Lett.* **2018**, *139*, 24–29. [CrossRef]
8. Kang, J.; Jeong, K.; Sung, J.; Hong, S.; Lee, K. Collision Attacks on AES-192/256, Crypton-192/256, mCrypton-96/128, and Anubis. *J. Appl. Math.* **2013**, *2013*, 713673. [CrossRef]
9. Chen, J.; Hu, Y.; Zhang, Y. Impossible differential cryptanalysis of advanced encryption standard. *Sci. China Ser. F—Inf. Sci.* **2007**, *50*, 342–350. [CrossRef]
10. Huang, H.; Liu, L.; Huang, Q.; Chen, Y.; Yin, S.; Wei, S. Low Area-Overhead Low-Entropy Masking Scheme (LEMS) against Correlation Power Analysis Attack. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2019**, *38*, 208–219. [CrossRef]
11. Ghosal, A.K.; Sardar, A.; Chowdhury, D.R. Differential fault analysis attack-tolerant hardware implementation of AES. *J. Supercomput.* 2023; *early access*. [CrossRef]
12. Dunkelman, O.; Keller, N.; Shamir, A. Improved Single-Key Attacks on 8-Round AES-192 and AES-256. *J. Cryptol.* **2015**, *28*, 397–422. [CrossRef]
13. Kim, C.H. Improved Differential Fault Analysis on AES Key Schedule. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 41–50. [CrossRef]
14. Potestad-Ordonez, F.E.; Tena-Sanchez, E.; Acosta-Jimenez, A.J.; Jimenez-Fernandez, C.J.; Chaves, R. Design and Evaluation of Countermeasures against Fault Injection Attacks and Power Side-Channel Leakage Exploration for AES Block Cipher. *IEEE Access* **2022**, *10*, 65548–65561. [CrossRef]
15. Mestiri, H.; Barraj, I.; Mohamed, A.A.; Machhout, M. An Efficient AES 32-Bit Architecture Resistant to Fault Attacks. *CMC—Comput. Mater. Contin.* **2022**, *70*, 3667–3683. [CrossRef]
16. Barenghi, A.; Breveglieri, L.; Koren, I.; Naccache, D. Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures. *Proc. IEEE* **2012**, *100*, 3056–3076. [CrossRef]
17. Bertoni, G.; Breveglieri, L.; Koren, I.; Maistri, P. An efficient hardware-based fault diagnosis scheme for AES: Performances and cost. In Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Cannes, France, 10–13 October 2004; pp. 130–138.
18. Guo, X.; Karri, R. Recomputing with permuted operands: A concurrent error detection approach. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2013**, *32*, 1595–1608. [CrossRef]
19. Doulcier-Verdier, M.; Dutertre, J.M.; Fournier, J.; Rigaud, J.B.; Robisson, B.; Tria, A. A side-channel and fault-attack resistant AES circuit working on duplicated complemented values. In Proceedings of the IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, 20–24 February 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 274–276.
20. Yifei, Q. *Clock Fault Injection Attack on AES and Countermeasures*; Huazhong University of Science and Technology: Wuhan, China, 2017.
21. Tupsamudre, H.; Bisht, S.; Mukhopadhyay, D. Destroying Fault Invariant with Randomization a Countermeasure for AES against Differential Fault Attacks. In Proceedings of the 2014 Workshop on Cryptographic Hardware and Embedded Systems, Busan, Republic of Korea, 23–26 September 2014; pp. 93–111.
22. Zhang, J.; Wu, N.; Zhang, X.; Zhou, F. Against fault attacks based on random infection mechanism. *IEICE Electron. Express* **2016**, *13*, 20160872. [CrossRef]
23. Joye, M.; Manet, P.; Rigaud, J.-B. Strengthening Hardware AES Implementations against Fault Attack. *IET Inf. Secur.* **2007**, *1*, 106–110. [CrossRef]

24. Lomne, V.; Roche, T.; Thillard, A. On the Need of Randomness in Fault Attack Countermeasures—Application to AES. In Proceedings of the 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium, 9 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 85–94.
25. Gierlichs, B.; Schmidt, J.-M.; Tunstall, M. Infective Computation and Dummy Rounds: Fault Protection for Block Ciphers without Check-before-Output. In *Progress in Cryptology—LATINCRYPT 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 305–321.
26. Battistello, A.; Giraud, C. Fault Analysis of Infective AES Computations. In Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Los Alamitos, CA, USA, 20 August 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 101–107.
27. van Woudenberg, J.G.; Witteman, M.F.; Menarini, F. Practical optical fault injection on secure microcontrollers. In Proceedings of the Workshop on Fault Diagnosis Tolerance in Cryptography (FDTC), Nara, Japan, 28 September 2011; pp. 91–99.
28. Wang, B.; Liu, L.; Deng, C.; Zhu, M.; Yin, S.; Wei, S. Against Double Fault Attacks: Injection Effort Model, Space and Time Randomization Based Countermeasures for Reconfigurable Array Architecture. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1151–1164. [CrossRef]
29. Zhang, J.; Wu, N.; Zhang, X.; Shen, L.; Zhou, F. Against Double Fault Attacks Based on Countermeasures for Second Order Infection Mechanism. In Proceedings of the Word Congress on Engineering and Computer Science (WCECS), San Francisco, CA, USA, 19–21 October 2016; pp. 19–21.
30. Fournier, J.; Rigaud, J.B.; Bouquet, S.; Robisson, B.; Tria, A.; Dutertre, J.M.; Agoyan, M. Design and characterisation of an AES chip embedding countermeasures. *Int. J. Intell. Eng. Inform.* **2011**, *1*, 328–347. [CrossRef]
31. Shan, W.; Fu, X.; Xu, Z. A Secure Reconfigurable Crypto IC with Countermeasures against SPA, DPA, and EMA. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1201–1205. [CrossRef]