electronics

MDPI

Article

The Design of Compact SM4 Encryption and Decryption Circuits That Are Resistant to Bypass Attack

Fang Zhou [1,2], Benjun Zhang [1], Ning Wu [1,*] and Xiangli Bu [1]

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; zfnuaa@nuaa.edu.cn (F.Z.); zhangbenjun@nuaa.edu.cn (B.Z.); amubu@foxmail.com (X.B.)

[2] Science and Technology on Electronic Information Control Laboratory, Chengdu 610036, China

* Correspondence: wunee@nuaa.edu.cn; Tel.: +86-139-5189-3307

Received: 18 May 2020; Accepted: 4 July 2020; Published: 6 July 2020

Abstract: In order to achieve the purpose of defending against side channel attacks, a compact SM4 circuit was designed based on the mask and random delay technique, and the linear transformation module was designed with random insertion of the pseudo operation method. By analyzing the glitch data generated by the S-box of SM4 with different inputs, the security against glitch attacks was confirmed. Then, the DPA (Differential Power Analysis) was performed on the designed circuit. The key could not be successfully obtained even in the case of 100,000 power curves, so that the safety of SM4 against DPA is verified. Finally, using Synopsys DC (Design Compiler, Mountain View, CA94043DC, USA) to synthesize the designed circuit, the results show that the area of the designed circuit in the SMIC 0.18 process is 82,734 μm^2, which is 48% smaller than results reported in other papers.

Keywords: SM4; cryptographic circuit; mask; resist power analysis

1. Introduction

The SM4 algorithm is a block symmetric cipher algorithm announced by Chinese National Cipher Management Committee Office in January 2006 and it has been widely used in various fields of information security in China, such as wireless local area network (WLAN), WLAN Authentication and Privacy Infrastructure (WAPI), storage device and the smart card system. As the SM4 algorithm is mostly used in high-speed and resource constrained applications, it is very necessary to design and implement the compact circuit of SM4. As a standard cipher algorithm, SM4 has been widely used in the field of information security for its short build time and low memory requirements [1–3]. However, Differential Power Analysis (DPA), which has been developed in recent years, has brought great challenges to the security of SM4 circuits. DPA is a typical SCA (side channel attack) method which performs a correlation analysis by collecting the power consumption of the operation. According to the correlation between sensitive information in the operation and the instantaneous power consumption of the CMOS circuit, DPA can quickly recover the key of SM4. It has the advantages of simple implementation, high efficiency, and short attack time. Therefore, it has posed a serious threat to the security of an integrated circuit. The goal of this work is to study compact SM4 circuits resistant to SCA for resource constrained applications.

Since Kocher proposed PA (Power Analysis, USA) technology in 1998, the research on PA and defense measures of cryptographic circuits have increased [4]. In recent years, ML (machine learning, USA) and PCA (principal component analysis) have also been applied to PA because of the large amount of data needed to be statistically analyzed [5,6]. Based on the principle of PA, the correlation between

the power consumption of the circuit and the intermediate value of the cryptographic operation can be weakened or shielded, so that the attacker cannot recover the key through the power consumption information. The defense strategy of PA includes the masking method and randomization method.

1.1. Masking Method

Because the intermediate value of cryptographic algorithm operations has certain correlation with power consumption, the masking method has become the most commonly used defense method and is widely used in algorithms such as AES (Advanced Encryption Standard), and SM4 [1,7,8]. The principle of the masking method is to allow the key to be "hidden" by using a random or fixed mask value before the plaintext and key data are calculated. The intermediate value after the operation with the plaintext is a random amount. In this way, the power consumption related to the intermediate value is also random. An attacker cannot define a distinguishing function. It is difficult to collect useful power consumption curves. It is impossible to analyze the key based on the power consumption curve.

Akkar et al. [1,8,9] proposed an AES cryptographic circuit based on the random data masking method, but it does not really eliminate the vulnerability that can be attacked in the circuit because of the finite field inversion, which is the only non-linear part in its S-box that is not masked randomly. As a result, the circuit may be subjected to a differential Power Analysis or a high-order differential Power Analysis. A first-order masking-based countermeasure to defeat DPA and CPA (correlation power analysis) for SM4 was proposed in [7]. However, there was no area optimization of mask S-boxes in this work. There was also no analysis of the area resource in the experimental results. Tan Ruineng et al. [10] used the multiplication mask method to implement the mask S-box of SM4. Because the implementation of the multiplication mask is complicated and there are "zero value attack" defects, it cannot completely resist DPA attacks. In [11], for the first time, the author took the inverse part of the finite field as a whole, introduced a new quantity "∞", and defined Inv(X) over $GF(2^8) \cup \infty$, which is to replace "∞" with "0". Using this method to re-mask implementation can resist DPA. Reference [12] implemented the S-box of AES using Boolean mask method, which successfully resisted DPA attacks, but there was a problem of excessive circuit resource overhead. Mangard et al. [13] proposed an attack method for AES circuits that have implemented mask defense, namely glitch attacks. By analyzing the number of glitches in the S-box of AES, the correlation between the intermediate value and power consumption was successfully cracked. Liang et al. [14] designed a masked S-box of SM4 using composite field masking method, which can resist first-order DPA attacks, but it also cannot resist glitch attacks. The paper [15] designed a SM4 cryptographic circuit with a full-mask round transform structure based on the random mask method, but the problem of excessive resource overhead still exists. Therefore, how to reduce the circuit resource is still a content that needs to be researched while ensuring the safety of the circuit.

1.2. Randomization Method

The randomization method aims to destroy the correlation between the median value and power consumption by increasing the redundant power consumption or random noise. Randomization methods usually include random pseudo operations inserting and random delay. Herbst et al. [12] designed an AES cryptographic circuit based on randomization method. Since the AES algorithm has a width of 128 bits and the execution order of the bytes is not related, the operation process can be randomized during the operation. Inserting pseudo-operations disturbs the power consumption information of the circuit and achieves the purpose of resisting power consumption attacks. Kocher et al. [16] adopted a random insertion delay method for the clock of the cryptographic circuit, which destroyed the fixedness of the median operation time point and prevented the attacker from finding the power consumption data position corresponding to the attack point. This randomization method has the advantages of small circuit area and simple implementation, and is commonly used for resisting power consumption attacks.

In general, the masking method is strong in security and easy to implement, but it has the disadvantage of being unable to resist glitch attacks. Although the randomization method has a weak

ability to defend against attacks, its resource overhead is small. In order to achieve a circuit resistant to DPA for resource-constrained applications, this work will design and implement a compact SM4 encryption and decryption circuit based on masking and randomization method.

The rest of this manuscript is organized as follows: Section 2 introduces the overall structure design of SM4. Section 3 details the sub-module design of SM4. Section 4 analyzes the security of the proposed SM4 to resist glitch attacks and DPA. In Section 5, we study the synthesized results of the SM4 circuit. Section 6 summarizes the conclusions of this work.

2. Overall Structure Design

Because the attacker can use the power consumption information of the cryptographic circuit at any time to crack the original key of the circuit, it is not safe for the entire SM4 encryption and decryption circuit to take defensive measures against only some modules of the circuit. The security design should protect SM4 circuit during each operation of the algorithm. From [1], the attacked positions of SM4 are the input/output of the S-box, and the output of the linear transformation. Thus, the attackers should choose the input/output of the S-box as the position of Power Analysis. Reference [1,10] respectively selected the input and output of the S-box to attack the SM4 encryption circuit, and successfully obtained the key. This paper intends to use a masking method to protect the T-box. Because the composite field mask S-box is easy attacked by glitch attacks, this paper further proposes the random delay method to change the time delay of the input data of the composite field mask S-box. The linear transformation performs different cyclic shifts on the output of the S-box to achieve the purpose of diffusion. Therefore, the attack of this position, a byte distinguishing function method cannot be adopted, only a word (32 bit) attack method can be used to implement the attack. In this case, the attacker needs to try at least 2^{32} data guesses, and the attack is very difficult. Therefore, we use other parallel structures in the circuit which do not need to work and insert random operations to generate randomized power consumption to disrupt the overall power of the circuit. Because the key expansion and round transformation module of the SM4 encryption and decryption circuit designed in this paper reuses the T-box part, there is no need to protect the key expansion separately. Using the above design ideas, the SM4 encryption and decryption circuit based on the random mask and randomization method is designed, as shown in Figure 1.

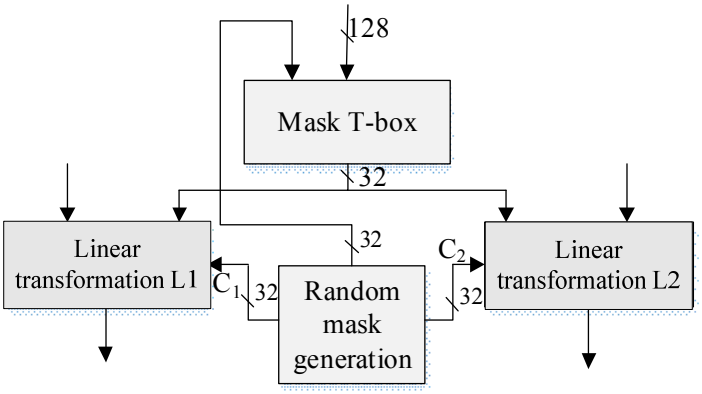

Figure 1. Schematic diagram of SM4 module resistant to Power Analysis.

Figure 1 includes a mask T-box, a linear transformation L1, a linear transformation L2, and a random mask generation module. Among them, the mask T-box occupies more than 40% of the combined logic overhead of the total circuit, and is an important part of round transformation and key expansion in the SM4 encryption and decryption circuit. The composite field mask S-box is also an important design for the T-box module. Based on the above design ideas, the following section will design a compact SM4 encryption and decryption circuit based on the mask and randomization method.

3. Sub-Module Design

This section will study the design and optimization of the mask T-box, random linear transformation L1, random linear transformation L2, and random mask generation module based on random delay, mask, and random insertion pseudo-operation method.

3.1. Design of the Mask T-Box

The designed circuit structure of the mask T-box is shown in Figure 2, which is mainly divided into three parts: XOR operation module, random delay module and mask S-box. Among them, at the beginning of the T-box operation, the mask M_I and the key rk or the fixed parameter CK are operated to mask the intermediate value result. Then the mask is input into the random delay module along with the round data D gained by the exclusive XOR operation to delay the data path. The delayed data enters the mask S-box for calculation. Finally, the mask S-box outputs the non-linearly transformed data S and updated mask M_S.

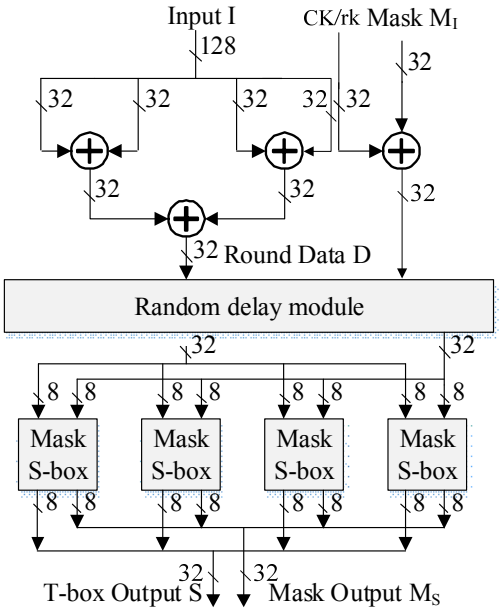

Figure 2. Mask T-box circuit structure diagram.

Since the composite field mask S-box cannot resist glitch attacks, we use a composite field mask method based on random delay to improve the security of the S-box. From the principle of the glitch attack, it is known that the number of glitches in the circuit is related to the power consumption of the circuit. When different values are inputted in the S-box, the number of glitches in the S-box operation circuit is different. Thus, the attacker can establish the connection between the input value of the S-box and the power consumption. The key of the circuit was successfully attacked through a differential Power Analysis [13]. Therefore, as long as the correlation between the number of glitches and the input of the S-box is destroyed, the purpose of resisting glitch attacks can be achieved. Based on the above analysis, the structure of the designed random delay module is shown in Figure 3.

In Figure 3, each triangle represents a buffer unit, which is made up of four identical NOT gates connected in a series. The selector determines the delay of the data path by several units and outputs it according to the mask input of the random delay module. Because the round data D to S-box is 32-bit data, it needs 32 random delay modules as shown in Figure 3. Among them, since each selector requires a 3-bit control signal, the 32-bit mask MI can only be used to control 10 selectors. The glitch attack is an attack against a single S-box. Even if the delay between the S-box is the same, the security of the S-box against the glitch attack will not be affected. We use the lower 24 bits of the mask MI to control the outputs of the 32 selectors of the data path.

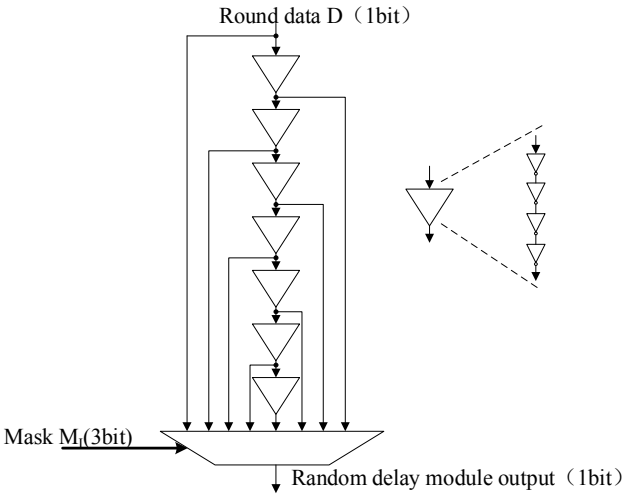

Figure 3. Random delay module diagram.

The most complicated part of the T-box is the mask S-box. Based on the composite field mask method, the mask S-box module structure is shown in Figure 4. Because the round data to S-box is 32-bit data, it needs 4 mask S-box modules, as shown in Figure 4.

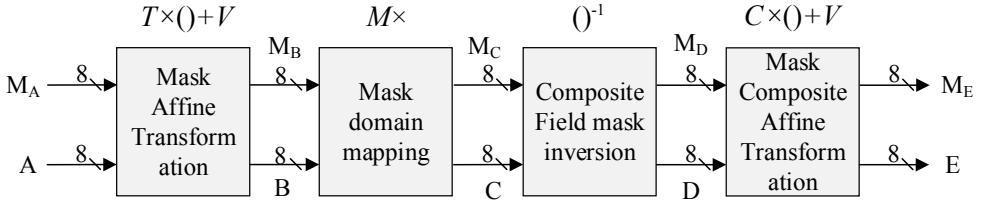

Figure 4. Composite Field mask S-box circuit structure diagram.

In Figure 4, M_A indicates 8-bit mask input, A indicates 8-bit input data, and E indicates 8-bit output data. After the mask affine transformation module, mask field mapping module, composite field mask inversion module, and mask composite affine transformation module of the SM4 S-box, the masks are M_B, M_C, M_D, and M_E, respectively. In the case of determining prime polynomials over $GF((2^4)^2)$ and $GF(2^4)$, the mask design and optimization of each module in Figure 4 will be studied next.

3.1.1. Optimization Design of Mask Affine Transformation Module

Suppose A is the input of the mask affine transformation module, B is the output of the mask affine transformation module, and M_A and M_B are the mask input and output of the mask affine transformation module, respectively. B and M_B can be expressed in the forms shown in Equation (1).

$$\begin{cases} B = TA + V = T(M_A + X) + V \\ M_B = TM_A \end{cases} \tag{1}$$

where X is the input data without mask operation.

$$T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, V = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{2}$$

The DACSE (delay-aware common sub-expression elimination) optimization method is used to substitute Equation (2) into Equation (1), and the circuit logic expressions of the mask output and output data of the optimized mask affine transformation module are shown in (3) and (4), respectively.

$$M_B = \begin{cases} m_{B7} = (m_1 \oplus m_0) \oplus z_7 \\ m_{B6} = (m_3 \oplus z_0) \oplus z_4 \\ m_{B5} = (m_5 \oplus m_2) \oplus z_7 \\ m_{B4} = z_2 \oplus z_6 \\ m_{B3} = z_5 \oplus z_6 \\ m_{B2} = z_3 \oplus z_8 \\ m_{B1} = (m_3 \oplus z_5) \oplus z_2 \\ m_{B0} = z_4 \oplus z_8 \end{cases}, \begin{cases} z_0 = m_7 \oplus m_6 \\ z_1 = m_7 \oplus m_2 \\ z_2 = m_6 \oplus m_1 \\ z_3 = m_4 \oplus m_3 \\ z_4 = m_5 \oplus m_0 \\ z_5 = m_2 \oplus m_0 \end{cases}, \begin{cases} z_6 = m_5 \oplus z_3 \\ z_7 = m_4 \oplus z_0 \\ z_8 = m_1 \oplus z_1 \end{cases} \tag{3}$$

$$B = \begin{cases} b_7 = (x_1 \odot x_0) \oplus p_7 \\ b_6 = (x_3 \oplus p_0) \oplus p_4 \\ b_5 = (x_5 \oplus x_2) \oplus p_7 \\ b_4 = p_2 \oplus p_6 \\ b_3 = p_5 \oplus p_6 \\ b_2 = p_3 \oplus p_8 \\ b_1 = (x_3 \oplus p_5) \oplus p_2 \\ b_0 = p_4 \oplus p_8 \end{cases}, \begin{cases} p_0 = x_7 \oplus x_6 \\ p_1 = x_7 \oplus x_2 \\ p_2 = x_6 \odot x_1 \\ p_3 = x_4 \oplus x_3 \\ p_4 = x_5 \odot x_0 \\ p_5 = x_2 \oplus x_0 \\ p_6 = x_5 \oplus p_3 \\ p_7 = x_4 \oplus p_0 \\ p_8 = x_1 \oplus p_1 \end{cases} \tag{4}$$

where the symbol $\oplus$ represents XOR operation, the symbol $\odot$ represents XNOR operation.

3.1.2. Optimized Design of Mask Field Mapping Module

Assume that B is the input of the mask field mapping module, C is the output of the mask field mapping module, and M_B and M_C are the mask input and output of the mask field mapping module, respectively. According to (5), C can be expressed as the Equation (6).

$$\begin{cases} B = M_B + X \\ C = MB = M(M_B + X) \\ M_C = MM_B \end{cases} \tag{5}$$

$$C = \begin{cases} c_7 = b_3 \oplus p_6 \\ c_6 = p_3 \oplus p_0 \\ c_5 = b_7 \oplus p_4 \\ c_4 = p_4 \\ c_3 = p_0 \oplus p_1 \\ c_2 = p_6 \\ c_1 = p_5 \oplus b_2 \\ c_0 = b_0 \oplus p_5 \end{cases}, \begin{cases} p_0 = b_7 \oplus b_1 \\ p_1 = b_6 \oplus b_5 \\ p_2 = b_4 \oplus b_2 \\ p_3 = b_2 \oplus b_3 \end{cases}, \begin{cases} p_4 = b_5 \oplus p_3 \\ p_5 = b_1 \oplus p_1 \\ p_6 = p_1 \oplus p_2 \end{cases} \tag{6}$$

The mask field mapping module performs the same field mapping operation on the masked data B and the mask M_B, so it can be directly implemented using two field mapping modules.

3.1.3. Optimized Design of Mask Composite Affine Transformation Module

Similar to the mask affine transformation module, suppose M_D and M_E are the mask input and output of the mask composite affine transformation module, respectively, and D and E are the input and output of the mask composite affine transformation module, respectively. The mask output M_E

and output data E of the mask composite affine transformation module can be deduced, as shown in the Equation (7)

$$\begin{aligned} M_E &= CM_D \\ E &= CD + V \end{aligned} \tag{7}$$

$$C = TM^{-1} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{8}$$

Substituting (8) into (7) and optimizing with the DACSE optimization method, the circuit logic expressions of the mask operation and data operation of the composite affine transformation module are shown in (9) and (10), respectively.

$$M_E = \begin{cases} m_{E7} = z_4 \oplus m_2 \\ m_{E6} = z_4 \oplus (m_3 \oplus m_5) \\ m_{E5} = z_3 \oplus m_5 \\ m_{E4} = z_3 \oplus z_1 \\ m_{E3} = (m_7 \oplus m_0) \oplus z_1 \\ m_{E2} = (m_7 \oplus m_3) \\ m_{E1} = z_2 \\ m_{E0} = z_4 \end{cases}, \begin{cases} z_0 = m_6 \oplus m_1 \\ z_1 = m_6 \oplus m_5 \\ z_2 = m_4 \oplus m_0 \\ z_3 = m_2 \oplus m_4 \end{cases}, \{z_4 = z_0 \oplus z_2 \tag{9}$$

$$Z = \begin{cases} z_7 = p_4 \oplus n_2 \\ z_6 = p_4 \oplus (n_3 \oplus n_5) \\ z_5 = p_3 \oplus n_5 \\ z_4 = p_3 \odot p_1 \\ z_3 = (n_7 \oplus n_0) \oplus p_1 \\ z_2 = (n_7 \oplus n_3) \\ z_1 = p_2 \\ z_0 = p_4 \end{cases}, \begin{cases} p_0 = n_6 \oplus n_1 \\ p_1 = n_6 \oplus n_5 \\ p_2 = n_4 \odot n_0 \\ p_3 = n_2 \oplus n_4 \\ p_4 = p_0 \oplus p_2 \end{cases} \tag{10}$$

3.1.4. Optimal Design of the Inverse Operation of the Mask GF($(2^4)^2$)

The constituent modules of the $GF((2^4)^2)$ inversion circuit are all operations over $GF((2^4)^2)$. The four mask operations over $GF(2^4)$ are designed in detail below.

(a) Mask GF(2^4) Add Operation

Since the mask operation is consistent with the masked data operation, only two $GF(2^4)$ add operations can be used to achieve the masked $GF(2^4)$ add operation. The output data and mask of the masked $GF(2^4)$ add operation are shown in (11) and (12), respectively.

$$Z = X + Y = \begin{cases} z_3 = x_3 \oplus y_3 \\ z_2 = x_2 \oplus y_2 \\ z_1 = x_1 \oplus y_1 \\ z_0 = x_0 \oplus y_0 \end{cases}, \begin{array}{c} X = (x_3, x_2, x_1, x_0), \\ Y = (y_3, y_2, y_1, y_0), \\ Z = (z_3, z_2, z_1, z_0). \end{array} \tag{11}$$

$$M_Z = M_X + M_Y = \begin{cases} m_{Z3} = m_{X3} \oplus m_{Y3} \\ m_{Z2} = m_{X2} \oplus m_{Y2} \\ m_{Z1} = m_{X1} \oplus m_{Y1} \\ m_{Z0} = m_{X0} \oplus m_{Y0} \end{cases}, \begin{array}{l} M_X = (m_{X3}, m_{X2}, m_{X1}, m_{X0}), \\ M_Y = (m_{Y3}, m_{Y2}, m_{Y1}, m_{Y0}), \\ M_Z = (m_{Z3}, m_{Z2}, m_{Z1}, m_{Z0}). \end{array} \tag{12}$$

Among them, X and Y represent the input data of the mask $GF(2^4)$ add operation, and M_X and M_Y represent the input masks of the mask $GF(2^4)$ add operation. Z and M_Z represent the output data and output mask of the masked $GF(2^4)$ add operation.

(b) Mask GF(2^4) Multiplication

Suppose that X' and Y' represent the input data of the mask $GF(2^4)$ multiplication operation after demasking, and Z' is the output data of the mask $GF(2^4)$ multiplication operation after demasking, the Equation (13) can be obtained.

$$Z' = X'Y' = (X + M_X)(Y + M_Y) = XY + XM_Y + YM_X + M_XM_Y \tag{13}$$

If the output mask of the module is M_Z and the output data is Z, $M_Z = Z \oplus Z'$. In order to avoid exposing the intermediate results with direct operation of the input data and the mask, the mask operation equation and data operation equation of the mask $GF(2^4)$ multiplication operation are obtained, as shown in (14).

$$\begin{cases} Z = X((Y + M_X) + M_Y) \\ M_Z = M_X((X + M_Y) + Y) \end{cases} \tag{14}$$

(c) Mask GF(2^4) Squared Constant Operation

Suppose that X and X' represent the masked and non-mask input data of the masked $GF(2^4)$ squared constant operation, M_X is the input mask. The calculation of Z', which is the output data of Mask $GF(2^4)$ squared constant operation, is shown in (15).

$$Z' = X'^2 \times k = (X + M_X)^2 \times k = X^2 \times k + {M_X}^2 \times k \tag{15}$$

It can be known from Equation (15) that this operation is composed of two $GF(2^4)$ squared constant operations. When a constant matrix is used, the $GF(2^4)$ squared constant operation has no area overhead, so the mask $GF(2^4)$ squared constant operation also requires no resource overhead.

(d) Mask GF(2^4) Inversion Operation

Suppose that X and X' represent the masked and non-mask input data of the masked $GF(2^4)$ inversion operation, M_X is the input mask. The calculation of Z', which is the output data of Mask $GF(2^4)$ inversion operation, is shown in (16).

$$\begin{aligned} Z' &= X'^{14} = \left(X^2 + {M_X}^2\right)\left(X^2 + {M_X}^2\right)^2\left(\left(X^2 + {M_X}^2\right)^2\right)^2 \\ &= \left(X^2 + {M_X}^2\right)\left(X^4 + {M_X}^4\right)\left(X^8 + {M_X}^8\right) \\ &= X^{14} + X^{12}{M_X}^2 + X^{10}{M_X}^4 + X^8{M_X}^6 \\ &+ X^6{M_X}^8 + X^4{M_X}^{10} + X^2{M_X}^{12} + {M_X}^{14} \end{aligned} \tag{16}$$

If the operation is directly implemented in the manner of Equation (16), multiple $GF(2^4)$ multiplication and addition modules need to be consumed. This not only causes a huge area overhead, but also increases the critical path delay of the circuit. If the mask output of the module is M_Z and the data output is Z, then exists. After analysis, in order to avoid the direct operation of the input

data and input mask to expose the intermediate results, the mask operation formula and data operation formula of the inversion operation of the mask $GF(2^4)$ are obtained, as shown in Equation (17).

$$\begin{cases} Z = X^8 Q \\ M_Z = {M_X}^8 Q \end{cases}, Q = (X^6 + X^4 {M_X}^2 + X^2 {M_X}^4 + {M_X}^6) \tag{17}$$

Since the expression in (17) still requires multiple $GF(2^4)$ multiplication operations, the expression can be simplified and set to obtain the simplified expression, as shown in (18).

$$Q = \begin{cases} q_3 = a_2(m_1 + m_3) + (a_1 + m_3)(m_2 + m_3) + a_3(m_2 + m_1) + m_1(a_0 + m_3) \\ \quad + a_1(a_2 + a_0) + m_1(m_2 + m_0) + a_3(a_1 + a_2) + (a_3 + a_1 m_0) \\ q_2 = a_2(m_0 + m_3) + a_3(m_2 + m_1) + a_0(m_2 + m_3) + (a_1 m_3 + a_3 m_0) \\ \quad + a_2(a_2 + a_0) + (a_3 a_1 + m_1 m_3) + a_3(a_2 + a_0) + (m_2 + m_3)(m_2 + m_0) \\ q_1 = a_0(m_2 + m_3) + (a_0 m_1 + a_2 m_0) + a_1(m_0 + m_3) + a_3(m_0 + m_1) \\ \quad + a_1(a_3 + a_1) + a_0(a_1 + a_2) + (a_3 a_0 + m_0 m_2) + (m_0 + m_1)(m_1 + m_3) \\ q_0 = a_0(m_2 + m_1) + (a_1 + m_0)(m_2 + m_0) + a_2(m_1 + m_3) + (a_2 m_0 + a_3 m_2) \\ \quad + a_2(a_3 + a_1) + a_0(a_1 + a_2) + m_2(m_1 + m_3) + (a_0 + m_1 m_0) \end{cases} \tag{18}$$

Let $T = X^8, M_T = {M_X}^8$, we can get the optimized Q output, as shown in Equation (19).

$$\begin{cases} Q = \begin{cases} q_3 = F_{amm213} \oplus X_{am13} X_{m23} \oplus F_{amm312} \oplus (A_{am01} \oplus A_{m13}) \\ \quad \oplus a_1 X_{a02} \oplus m_1 X_{m02} \oplus a_3 X_{Na12} \oplus a_1 m_0 \\ q_2 = a_2 X_{m03} \oplus F_{amm312} \oplus F_{amm023} \oplus (a_1 m_3 \oplus a_3 m_0) \\ \quad \oplus X_{a23} X_{a02} \oplus (a_1 a_3 \oplus A_{m13}) \oplus X_{m23} X_{m02} \\ q_1 = F_{amm023} \oplus (A_{am01} \oplus A_{am20}) \oplus a_1 X_{m03} \oplus a_3 X_{m01} \\ \quad \oplus a_1 X_{a13} \oplus a_0 X_{a12} \oplus (a_0 a_3 \oplus m_0 m_2) \oplus X_{m01} X_{m13} \\ q_0 = a_0 X_{m12} \oplus X_{am10} X_{m02} \oplus F_{amm213} \oplus (A_{am20} \oplus a_3 m_2) \\ \quad \oplus a_2 X_{a13} \oplus a_0 X_{Na12} \oplus m_2 X_{m13} \oplus m_0 m_1 \end{cases} \\ T = X^8 = \begin{cases} t_3 = X_{a13} \\ t_2 = a_3 \\ t_1 = X_{a23} \\ t_0 = X_{a03} \end{cases}, M_T = {M_X}^8 = \begin{cases} m_{T3} = X_{m13} \\ m_{T2} = m_3 \\ m_{T1} = X_{m23} \\ m_{T0} = X_{m03} \end{cases} \end{cases}, \begin{cases} X_{m13} = m_1 \oplus m_3 \\ X_{m23} = m_2 \oplus m_3 \\ X_{m03} = m_0 \oplus m_3 \\ X_{m12} = m_1 \oplus m_2 \\ X_{m02} = m_0 \oplus m_2 \\ X_{m01} = m_0 \oplus m_1 \\ X_{a13} = a_1 \oplus a_3 \\ X_{a03} = a_0 \oplus a_3 \\ X_{a23} = a_2 \oplus a_3 \\ X_{a02} = a_0 \oplus a_2 \\ X_{Na12} = a_1 \odot a_2 \\ X_{a12} = a_1 \oplus a_2 \end{cases}, \begin{cases} X_{am13} = a_1 \oplus m_3 \\ X_{am03} = a_0 \oplus m_3 \\ X_{am10} = a_1 \oplus m_0 \\ A_{am01} = a_0 m_1 \\ A_{am20} = a_2 m_0 \\ A_{m13} = m_1 m_3 \\ F_{amm213} = a_2 X_{m13} \\ F_{amm023} = a_0 X_{m23} \\ F_{amm312} = a_3 X_{m12} \end{cases} \tag{19}$$

It can be known from Equation (19) that the mask $GF(2^4)$ multiplication operation requires two $GF(2^4)$ multiplication operations in addition to the above operations.

The mask $GF((2^4)^2)$ inversion circuit mainly includes two mask $GF(2^4)$ adds, three mask $GF(2^4)$ multiplications, one mask $GF(2^4)$ squared constant and one mask $GF(2^4)$ inversion. Based on the SMIC 0.18 μm process, the comparison before and after optimization of the mask $GF((2^4)^2)$ inversion operation circuit is shown in Table 1.

After optimized design, the area overhead of the mask $GF((2^4)^2)$ inversion circuit implemented in this section is $231A_{XOR}$ + $159A_{AND}$ + $1AX_{NOR}$, which is 8298.81 μm^2, and its critical path delay is $19T_{XOR}$ + $4T_{AND}$. Before optimization, 414XOR gates and 322 AND gates are required, with an area of 15,302.36 μm^2, and the critical path delay is $21T_{XOR}$ + $4T_{AND}$. Compared with the situation before optimization, the critical path delay of the circuit is reduced by 8.3%, and the circuit area is reduced by about 45.83%.

Based on the above analysis and the SMIC 0.18 μm process, the mask S-box implemented in this section is synthesized. The comparison between before and after the optimization is shown in Table 2.

Table 1. Comparison of area and delay before and after optimization of mask $GF((2^4)^2)$ inversion.

-			AND	XOR	XNOR	Value
Mask addition *2	Before optimization	Area	-	8	-	212.88 μm^2
		Delay	-	1	-	0.21 ns
	After optimization	Area	-	8	-	212.88 μm^2
		Delay	-	1	-	0.21 ns
Mask multiplication *3	Before optimization	Area	64	68	-	2661.32 μm^2
		Delay	1	5	-	1.21 ns
	After optimization	Area	32	46	-	1649.98 μm^2
		Delay	1	5	-	1.21 ns
Mask squared constant	Before optimization	Area	50	42	-	1783.12 μm^2
		Delay	1	3	-	0.79 ns
	After optimization	Area	-	-	-	0
		Delay	-	-	-	0
Mask inversion	Before optimization	Area	80	152	-	5109.52 μm^2
		Delay	2	9	-	2.21 ns
	After optimization	Area	63	77	1	2914.11 μm^2
		Delay	2	7	0	1.79 ns
Total	Before optimization	Area	322	414	-	15,302.36 μm^2
		Delay	4	21	-	5.05 ns
	After optimization	Area	159	231	1	8289.81 μm^2
		Delay	4	19	0	4.63 ns

Table 2. Comparison of the area and delay of each module before and after the mask S-box optimization.

Module			AND	XOR	XNOR	Total
Mask Affine Transformation	Before optimization	Area	-	69	-	1836.09 μm^2
		Delay	-	3	-	0.63 ns
	After optimization	Area	-	39	4	1144.23 μm^2
		Delay	-	3	0	0.63 ns
Mask field mapping	Before optimization	Area	-	48	-	1277.28 μm^2
		Delay	-	3	-	0.63 ns
	After optimization	Area	-	28	-	745.08 μm^2
		Delay	-	2	-	0.42 ns
Mask composite field inversion	Before optimization	Area	322	414	-	15302.36 μm^2
		Delay	4	21	-	5.05 ns
	After optimization	Area	159	231	1	8289.81 μm^2
		Delay	4	19	0	4.63 ns
Mask Composite Affine Transformation	Before optimization	Area	-	49	-	1303.89 μm^2
		Delay	-	3	-	0.63 ns
	After optimization	Area	-	24	2	691.86 μm^2
		Delay	-	3	0	0.63 ns
Total	Before optimization	Area	322	580	-	19719.62 μm^2
		Delay	4	30	-	6.94 ns
	After optimization	Area	159	322	7	10870.98 μm^2
		Delay	4	27	0	6.10 ns

The SM4 S-box based on composite field mask implemented in this paper consumes 322 AND gates and 580 XOR gates before optimization. The area is 19,719.62 μm^2 and the critical path delay is $30T_{XOR} + 4T_{AND}$. After optimization, the critical path delay is reduced by 12.1%, and the circuit area is $7A_{XNOR} + 159A_{AND} + 322A_{XOR}$, which is equal to 10,870.98 μm^2, which is a reduction of 44.87%.

3.2. Design of Random Linear Transformation

According to the analysis above, the attack point of the SM4 cryptographic circuit is generally input or output of the S-box. Since the linear transformation part needs to be in the form of a word attack, the attack is difficult. Therefore, we use the method of randomly inserting pseudo operations to defend against power consumption attacks on the module. Since the round transformation and key expansion are performed serially, the linear transformation L2 of the round transformation is idle when the key expansion part of the calculation is performed. When the round transformation module is performed, the key expansion linear transformation L1 is also idle. Considering the idle modules, the structure of the designed random linear transformation module is shown in Figure 5.

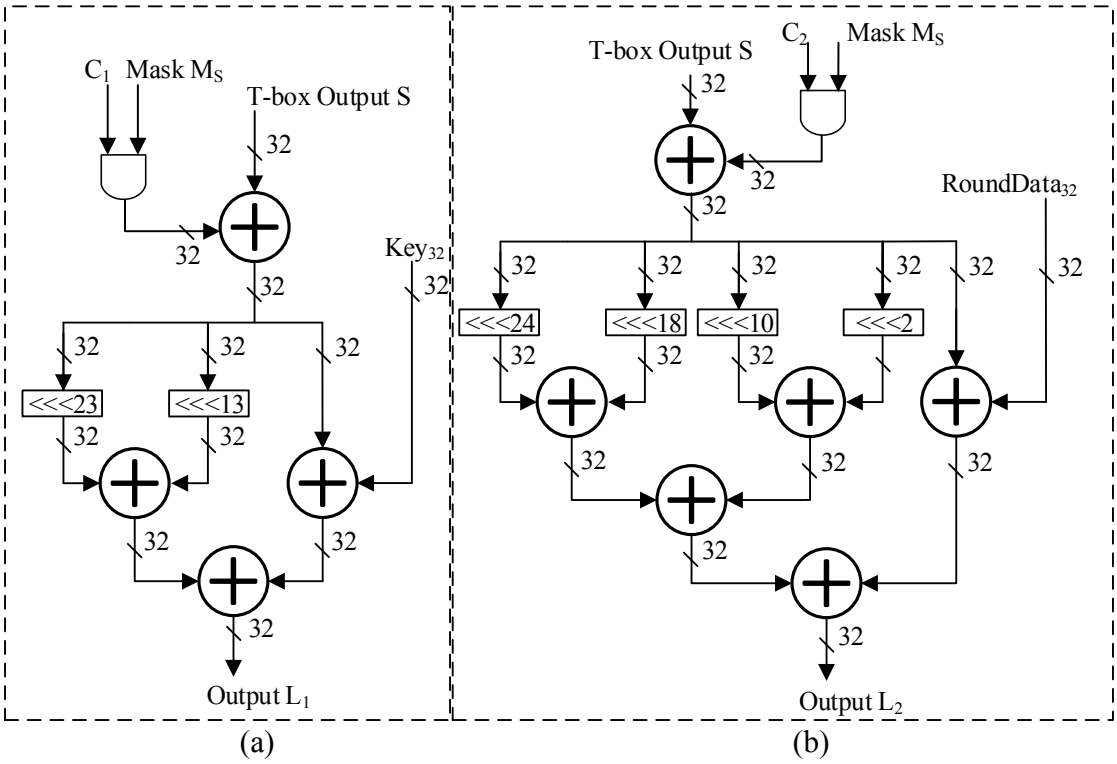

Figure 5. Random linear transformation operation module circuit structure diagram. (**a**) shows the random linear transformation L1 of the key extension, and (**b**) shows the random linear transformation L2 of the round transformation.

C1 and C2 come from the random mask generation module. When the round transformation operation is being performed, C1 is 0 and C2 is 32'hffffffff. At the same time, the linear transformation with mask data is performed in the random linear transformation L1, which disturbs the power consumption generated by the random linear transformation L2 of the round transformation being performed. Similarly, when the key expansion operation is performed to the random linear transformation operation, the random linear transformation module of the round transformation will also generate corresponding random power consumption, thereby increasing the difficulty of power consumption attack.

3.3. Design of Random Mask Generation Module

The random mask generation module is used to generate the random mask required by the mask T-box, the operands C1 and C2 required by the mask linear transformation module. Therefore, the random mask generation module designed in this section is shown in Figure 6.

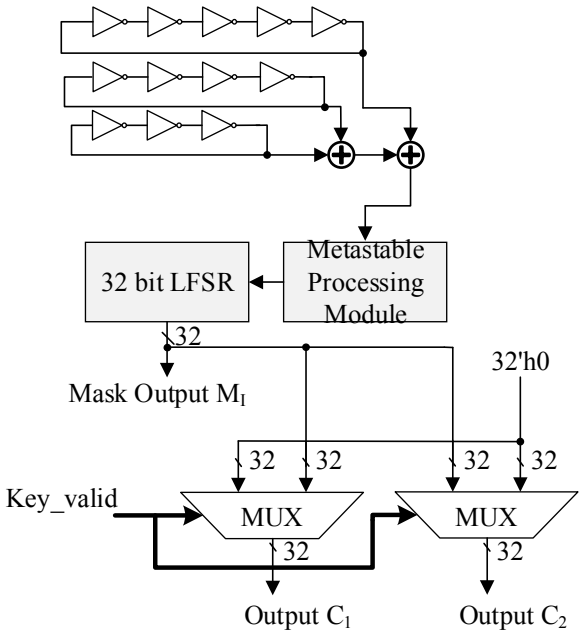

Figure 6. Circuit diagram of random mask generation module.

Because LFSR (linear feedback shift register) is a common method for generating pseudo-random numbers, a 32-bit LFSR is used in the random mask generation module. Figure 6 contains the ring oscillator operation unit, metastable processing unit, 32-bit LFSR, and selection unit. Among them, the loop oscillator operation unit is used to generate random numbers based on circuit characteristics. The metastable processing module is made up of two D flip-flops connected in series. It can synchronize the data generated by the loop oscillator operation unit to the clock field of the SM4 encryption and decryption circuit and eliminate the metastable state. LFSR relies on the input single-bit random data based on the characteristics of the circuit, so that the output random number tends to be true random in order to ensure the security of the mask output. When the Key_valid signal is 0, the C1 output mask is selected and C2 to output 0; otherwise, C1 outputs 0 and C2 outputs the mask.

4. Security Analysis

This section first performs a security analysis on the random delay mask S-box to verify its ability to resist glitch attacks, and then verifies the anti-DPA attack performance of the compact SM4 encryption and decryption circuit designed in this paper.

4.1. Security Analysis of Random Delay Mask S-box

The principle of the glitch attack is to attack the key based on the correlation between the number of glitches in the S-box operation and the input of the S-box. Therefore, as long as the number of glitches in the S-box operation is random, the glitch attack can be resisted. Table 3 lists the relationship between the input value of the random delay mask S-box and the number of glitches in the case of some mask inputs.

Table 3. Relation table between random delay mask S-box input value and the number of glitches.

SboxIn / MaskIn	0xff	0xaa	0x44	0x56
0xedcba9	35	33	30	26
0xff24dc	33	39	31	32
0x156894	29	32	21	31
0xffffff	22	24	26	29
0x002456	34	29	23	26
0x380cd9	42	32	28	48
0x6ad0c3	31	24	28	29

In Table 3, SboxIn represents the input of the random delay mask S-box. MaskIn is used to control the random delay module to generate different delays in different data paths. It can be seen from Table 3 that, under different MaskIn, for the same SboxIn, the number of glitches generated by the S-box operation is different and tends to be random. The random delay mask S-box can effectively resist glitch attacks.

4.2. Security Analysis of SM4 Encryption and Decryption Circuit

To better evaluate the security of SM4 encryption and decryption circuit, there are two sets of tests performed on FPGA and ASIC, separately.

Firstly, the circuit designed in this paper is implemented on FPGA. The differential Power Analysis platform designed in [17] is used for data collection and Power Analysis. During the attack, the middle value of the selected attack is the high 8-bit output of the first round of the byte replacement operation, and the corresponding round key rk0 is 32'h15263748. After analyzing the collected 100,000 power consumption curves through Matlab software, the output Power Analysis results are shown in Figure 7.

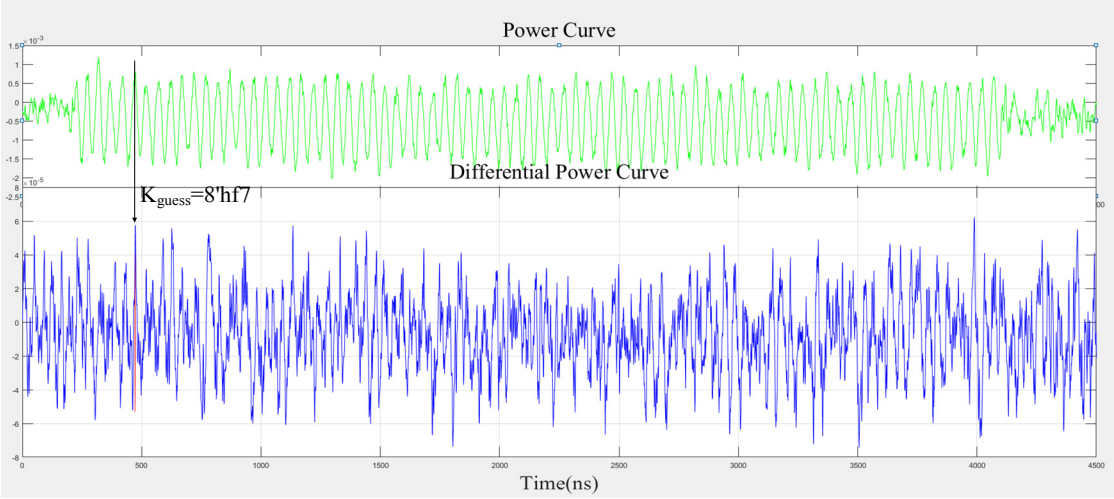

Figure 7. Differential Power Analysis (DPA) curve of the high 8-bit output of the first round of transformation byte replacement operation.

At the time of the intermediate value calculation output, the key K_{guess} corresponding to the DPA curve with the highest peak value is 8'hf7, and the guess key is wrong. Therefore, in the case of collecting 100,000 power consumption curves, the method of using a differential Power Analysis cannot crack the key of the circuit.

Secondly, Synopsys VCS, DC, Prime Time-PX and the other EDA software are used to simulate the operation process of the cryptographic circuit, and calculate the corresponding power consumption data according to the turnover rate of the cryptographic circuit in the simulation process. We take the

low 8-bit output of the second round of byte replacement operation as an example. The second round key rk1 is 32'h2937ac24. After analyzing the collected 100,000 power consumption curves, the output Power Analysis results are shown in Figure 8.

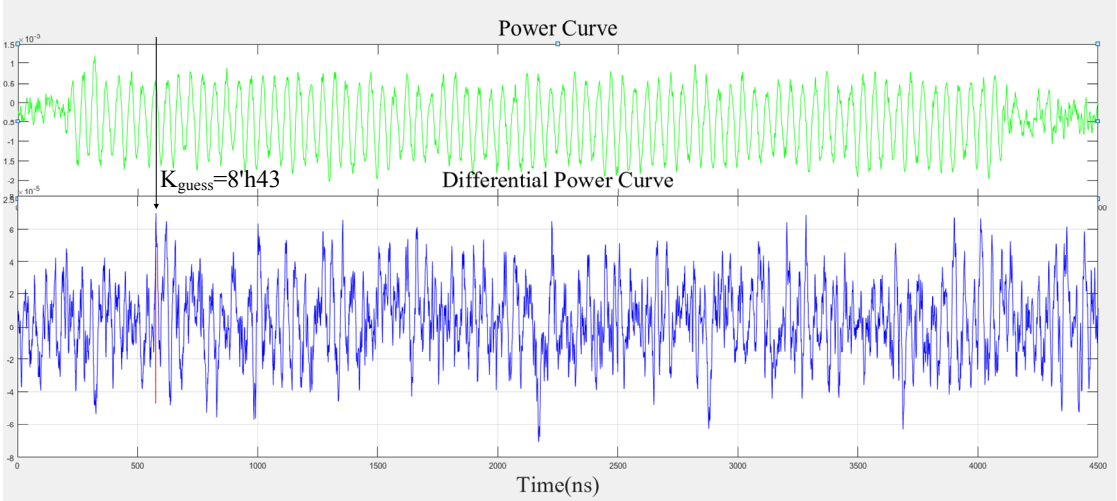

Figure 8. DPA curve of the low 8-bit output of the second round of transformation byte replacement operation.

At the time of the intermediate value calculation output, the key K_{guess} corresponding to the DPA curve with the highest peak value is 8'h43, and the guess key is also wrong. Therefore, it is proved again that the random mask and randomization scheme proposed in this paper ensured the security of the compact SM4 encryption and decryption circuit against DPA attacks.

5. Synthesize Results

Based on the Xilinx Zynq-7000 (XC7Z020CLG484) FPGA platform, the circuit designed in this paper is synthesized in the software of Vivado 2017.4, and then the circuit is implemented after adding constraints. Table 4 shows the resource consumption and performance evaluation of the circuit designed in this paper on FPGA. The SM4 encryption and decryption circuit based on random mask and randomization method designed in this paper achieved a throughput of 99.56 Mbps with a resource overhead of 968 LUTs and 536 FFs.

Table 4. Resource overhead and performance evaluation on FPGA.

Type	Name	Value
Resource overhead	LUT (Look up table)	968/53,200 (1.82%)
	FF (Register)	536/106,400 (0.38%)
Performance evaluation	Maximum clock frequency	60.67 MHz
	Critical path delay	16.482 ns
	Throughput	99.56 MHz/s

In addition, tools such as Synopsys DC and Prime Time-PX were used to synthesize and sequence the compact SM4 encryption and decryption circuit designed in this paper. Under the SMIC 0.18 μm process, the circuit area consumption was 82,734 μm^2. Considering that the area of the same circuit synthesized in different SIMC processes is different, we take NAND gate (9.9792 μm^2) as the standard gate and convert the circuit area to the number of NAND gates. The number of NAND gates in this circuit is 8290, and the critical path delay is 8.93 ns. A comparison of the characteristic parameters and resource overhead of the circuits designed in this paper with other circuits is shown in Table 5.

Table 5. Comparison of performance parameters and resource overhead of ASIC implementation.

Parameter	This Article	[1]	[10]
Process	SMIC 0.18 μm	SMIC 0.13 μm	SMIC 0.18 μm
Clock frequency (MHz)	110.1	50	50
Area (μm^2)	82,734	-	-
Equivalent gate (gate)	8290	16,000	36,000
Throughput (Mb/s)	207	200	200
Throughput/gate (Mb·s-1·kgate-1)	24.97	12.5	5.56
Attack cost (number of curves)	100,000+	500,000+	500,000+
Circuit characteristics	Resistant to DPA (including Glitch Attack), support encryption and decryption	Resistant to DPA, only support encryption	Resists DPA, only supports encryption

As can be seen from the table, the SM4 encryption circuit designed in this paper has better throughput/gate parameters. At the same time, the circuit in the paper [1] cannot resist glitch attacks, and only supports SM4 encryption operations, so it is not as good as the circuit designed in this paper in terms of security and practicability.

6. Conclusions

This paper focuses on the compact implementation of SM4 encryption and decryption circuits that are resistant to bypass attacks. In view of the inability to resist differential Power Analysis, a SM4 encryption and decryption circuit based on mask and randomization method is proposed. A mask S-box is designed using a composite field masking technique, so that the composite field inverse operation in the mask S-box can be truly masked. The random delay method is used to control the delay of each bit in the input signal to resist glitch attacks. The random linear transformation module is implemented by using a random insertion pseudo operation, which increases the difficulty of DPA to this module. Next, the security of the SM4 S-box against glitch attack is analyzed, and two bypass attack verifications of the designed circuit are performed using Power Analysis platform based on FPGA and ASIC. The attack cannot be successful with 100,000 curves. Finally, based on the SMIC 0.18 μm process, Synopsys DC are used to synthesize the design circuit. The area consumption is 82,734 μm^2, which is 48% smaller than other papers. The compact SM4 encryption and decryption circuit based on the inverse operation comparison mechanism implemented in this paper has lower circuit resource overhead and higher security, and is a better implementation solution.

Author Contributions: Conceptualization and Data curation, F.Z.; Resources, B.Z.; Supervision, N.W.; Writing-original draft, F.Z.; Writing-review and editing, X.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for Central Universities (No. NS2017023 and No. NP2019102), Aeronautical Science Foundation of China (No. 201943052001), Project of Science and Technology on Electronic Information Control Laboratory.

Acknowledgments: The authors would like to thank Jinbao Zhang for his beneficial suggestions and comments.

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

References

1. Chao, P.E.I. A Method of masking SM4 and analysis against DPA attacks. *J. Cryptol. Res.* **2016**, *3*, 79–90.
2. Di, W.; Wu, L.; Zhang, X. Key-leakage hardware Trojan with super concealment based on the fault injection for block cipher of SM4. *Electron. Lett.* **2018**, *54*, 810–812.

3. Niu, Y.; Jiang, A. The low power design of SM4 cipher with resistance to differential power analysis. In Proceedings of the International Symposium on Quality Electronic Design, Santa Clara, CA, USA, 2–4 March 2015.
4. Yuanman, T. Research on Key Techniques of Design and Implementation of Power Analysis. Ph.D. Thesis, National University of Defense Technology, Changsha, China, 2008.
5. Feng, H.; Lin, W.; Shang, W.; Cao, J.; Huang, W. MLP and CNN-based Classification of Points of Interest in Side-channel Attacks. *Int. J. Networked Distrib. Comput.* **2020**, *8*, 108–117. [CrossRef]
6. Wang, Z.; Zhang, W.; Ma, P.; Wang, X.A. Power consumption attack based on improved principal component analysis. In Proceedings of the 2019 International Conference on Broadband and Wireless Computing, Communication and Applications, Antwerp, Belgium, 7–9 November 2019.
7. Bae, D.; Nam, S.; Ha, J. Side channel attack on block cipher SM4 and analysis of masking-based countermeasure. *J. Korea Inst. Inf. Secur. Cryptol.* **2020**, *30*, 39–49.
8. Liling, D. The Optimization and Research for AES Cipher Chips with Power Attack Resistance. Master's Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2016.
9. Akkar, M.-L.; Giraud, C. An implementation of DES and AES, secure against some attacks. In Proceedings of the Cryptographic Hardware and Embedded Systems (CHES 2001), Nara, Japan, 28 September–1 October 2001.
10. Ruineng, T.; Yuanyuan, L.; Jiaoling, T. SM4 multi-path multiplicative masking method against side-channel attack. *Comput. Eng.* **2014**, *40*, 103–108.
11. Courtois, N.T.; Goubin, L. An algebraic masking method to Prltect AES against power attacks. In Proceedings of the ICISC, Seoul, Korea, 1–2 December 2005.
12. Herbst, C.; Oswald, E.; Mangard, S. An AES smart card implementation resistant to power analysis attacks. In Proceedings of the ACNS, Singapore, 6–9 June 2006.
13. Mangard, S.; Pramstaller, N.; Oswald, E. Successfully attacking masked AES hardware implementations. In Proceedings of the CHES 2005, Edinburgh, UK, 29 August–1 September 2005.
14. Liang, H.; Wu, L.; Zhang, X.; Wang, J. Design of a masked S-box for SM4 Based on composite field. In Proceedings of the 2014 Tenth International Conference on Computational Intelligence and Security, Kunming, China, 15–16 November 2014.
15. Jing, L. Side-Channel Analysis and Implementation of FPGA based Cryptographic Algorithms. Master's Thesis, Hunan University, Changsha, China, 2011.
16. Kocher, P.; Jaffe, J. Using Unpredictable Information to Minimize Leakage from Smartcards and Other Cryptosystems. U.S. Patent 6,327,661, 4 December 2001.
17. Zhang, Y.; Wu, N.; Zhou, F.; Zhang, J.; Yahya, M.R. A Countermeasure against DPA on SIMON with an Area-Efficient Structure. *Electronics* **2019**, *8*, 240. [CrossRef]