Journal of Intelligent & Fuzzy Systems 31 (2016) 795–803
DOI:10.3233/JIFS-169011
IOS Press

A VLSI implementation of an SM4 algorithm resistant to power analysis

Siyang Yu[a,*], Kenli Li[b], Keqin Li[a,b], Yunchuan Qin[a] and Zhao Tong[c]
[a]*College of Information Science and Engineering, Hunan University, Changsha, China*
[b]*Department of Computer Science, State University of New York, New Paltz, New York, USA*
[c]*College of Mathematics and Computer Science, Performance Computing and Stochastic Information Processing (Ministry of Education of China), Hunan Normal University, Changsha, China*

Abstract. SM4 is a block cipher proposed by the Chinese government. Strengthening the research and extension of SM4 is significant to the development and promotion of Chinese cryptography standards. To date, research relevant to SM4 is rare. Thus, we propose the implementation of an SM4 algorithm resistant to power analysis. Ideally, a secure masking scheme is used for the SM4 cipher, which is particularly suited for implementation in the application specific integrated circuit. Moreover, the mask scheme in our chip implementation process is improved to make SM4 safer. Simulation results confirm that the use of counteractive measures resistant to power analysis is credible.

Keywords: SM4, S-box, Galois Field, mask, zero attack

1. Introduction

SM4 is a block cipher proposed by the Office of State Commercial Cipher Administration of China (OSCCA) [1]. It is the first Chinese commercial cipher algorithm. SM4 is significant to the wireless network industry and to commercial password research. Given that the SM4 cipher was only issued recently, few studies have been published regarding its security.

In traditional cryptography, a cryptographic algorithm is considered absolutely safe when it is mathematically safe. However, this ideal was refuted by differential power analysis (DPA), which was proposed by Kocher et al. [2] in 1999. Compared with traditional cryptography, DPA can obtain sensitive information by analyzing the correlation between physical information leakage and operational data. In recent years, DPA has become the most popular method of attack because of its cost-effective and time-efficient advantages.

SM4 has received serious threats with the development of DPA technology. The security of SM4 has become a common research focus. Therefore, in this paper, we proposed an approach to withstand DPA.

The remainder of this paper is organized as follows. Related work on power analysis and the countermeasures against power analysis are reviewed in Section 2. The SM4 block cipher is described briefly in Section 3. An improved SM4 implementation is proposed and described in detail in Section 4. The simulation results and discussion are presented in Section 5. The conclusions are reported in Section 6.

2. Related works

This section presents an overview of relevant work on power analysis and the countermeasures against power analysis.

*Corresponding author. Siyang Yu, College of Information Science and Engineering, Hunan University, Changsha 410082, China. E-mail: nickysy@hnu.edu.cn.

2.1. Differential power analysis

The DPA correlation equation was first proposed by Kocher et al. [2]:

$$\frac{\sum_{i=1}^{t} D(k_n, c_i)P_i[j]}{\sum_{i=1}^{t} D(k_n, c_i)} - \frac{\sum_{i=1}^{t} (1 - D(k_n, c_i))P_i[j]}{\sum_{i=1}^{t} (1 - D(k_n, c_i))}. \tag{1}$$

Equation (1) can be simplified into:

$$\Delta D[j] \approx 2\left[\frac{\sum_{i=1}^{t} D(c_i, k_n)P_i[j]}{\sum_{t=1}^{t} D(c_i, k_n)} - \frac{\sum_{i=1}^{t} P_i[j]}{t}\right]. \tag{2}$$

Let P denote the set of traces that are collected, and let P_i denote the ith trace. Let $P_i[j]$ denote the power measurement or sample at the jth time offset within the trace P_i. Let c denote the set of known plaintexts for the traces, with c_i corresponding to the ith trace. Let D denote a binary-valued selection function with plaintext c_i and the guess k_n of part of a key.

The DPA attack on the SM4 cipher can be performed as follows.

Step 1: Encrypt a set of random plaintexts P, $P = \{P_1, P_2, \ldots P_m\}$ by the same key and capture the power traces T, $T = \{T_1, T_2, \ldots T_m\}$.

Step 2: Perform a power analysis of the round key. We can divide the 32-bit round key into 8-bit blocks to be analyzed respectively. In this step, we choose an intermediate result I to be the object of the attack.

Step 3: Separate the power traces into two sets according to the selection function D. We calculate the difference between these sets using Equation (1). If a spike appears on the differential power trace, the hypothesis is correct. By this approach, we can obtain the entire key.

The value of ΔD depends on the correlation between the partitioning into sets and the information in the measured power traces. In turn, these factors depend on the guessed key k_n used by the selection function. In the case of high correlation, the two average power traces of two sets will differ at some point in time, thereby resulting in a high ΔD. However, the distributions in uncorrelated cases will not have statistically significant differences. The value of ΔD approaches zero as the number of traces increases.

2.2. Security strategy

To date, the approaches resistant to power analysis have made great progress [3–7]. The main protection technologies include the following: (1) Algorithm-level optimization. By optimizing the encryption algorithm, the process of operation is not related to the round key. (2) Mask technology. In this technology, the input and output data are masked, and the attacker finds it difficult to extract sensitive information from the power difference. (3) Balanced power. From the selection and design of the circuit logic, researchers improve the realization of the logic circuit [8–11]. In this manner, the system power consumption may not have considerable differences in various data.

Mask technology is an effective protection technology often used in the sequential approach. The basic principle of mask technology is to reduce the correlation between sensitive information and power consumption [12]. Masks include multiplication mask [13], random mask [14], fixed-value mask [15], etc. In [5], the authors introduced an anti-analysis model, which masks the intermediate results. Consequently, the selection function D cannot correctly divide the power trace, and the results cannot show the correlation. [16] proposed to apply the mask algorithm to the data encryption standard (DES) cipher. The authors designed a mask with S-BOX logic, such that the DES has better performance during anti-analysis. Research showed that each round of block cipher would be threatened by power analysis, and the mask algorithm would increase the overhead of the circuit hardware [17]. Therefore, the balance between complexity and security must be considered.

When the random mask technology used is resistant to the DPA, the random module must generate different random mask factors for each round of encrypting operation. According to the new factor, the mask logic would calculate new masks with S-BOX. The look-up tables have a nonlinear structure; and thus the design requires a great deal of storage space. This mask logic is not suitable for implementation on hardware. To solve this problem, Akkar and Giruad [12] incorporated mask technology into the AES algorithm. This technology involves exclusive or and multiplication operations. In 2002, Trichina, DeSeta and Germani simplified mask technology [18]. Nonetheless, this method can neither blur all of the intermediate results nor resist zero attack because its mask technology involves multiplication masks. Oswald, Mangard and Pramstaller [19] proposed an addition mask technology, which combines the multiplication and addition masks. Consequently, this method can effectively resist the zero attack.

The main contributions of the current study are as follows:

- This research aims to investigate the weakness of SM4 algorithms. It proposes a specific integrated circuit countermeasure against power analysis for SM4 design.
- In this paper, we introduce a mask technology applicable to the SM4 algorithm, which can effectively resist power analysis.
- We demonstrate in experimental results that our proposed SM4 design outperforms related algorithms in terms of power and hardware overhead.

3. Description of SM4

SM4 is a 32-round iterative algorithm; both the data block and the key size are fixed to 128 bits [1]. In this section, the SM4 algorithm is described in detail.

3.1. Notation

The follow notations are used throughout this paper. [1].

- Z_2^m denotes the set of m-bit vector. The element of Z_2^{32} is a word, and the element of Z_2^8 is a byte.
- $S()$ is the 8-bit substitution box (S-box).
- <<<i: is left rotation by i bits.
- $(X_i,X_{i+1},X_{i+2},X_{i+3}) \in (Z_2^{32})^4$ is the input of the i-th round ($0 \le i \le 31$).
- $(MK_0,MK_1,MK_2,MK_3) \in (Z_2^{32})^4$ is the 128-bit key. rk_i denotes the corresponding 32-bit round key in the i-th round ($0 \le i \le 31$).

3.2. Operation procedure of SM4

In the SM4 operation, let $(X_0,X_1,X_2,X_3) \in (Z_2^{32})^4$ and $(Y_0,Y_1,Y_2,Y_3) \in (Z_2^{32})^4$ present the 128-bit plaintext P and the 128-bit ciphertext C respectively. In our design, the first round is referred to as Round 0.

The encryption procedure of SM4 is as follows:

$$\begin{aligned} X_{i+4} &= F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) \\ &= (X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)) \end{aligned} \tag{3}$$

where $i = 0,1,...,31$. Finally, the 128-bit ciphertext is the output of Round 31:

$$(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32}). \tag{4}$$

In Equation (3), F is the round function, and T is a composite transformation. The transformation T is composed of the nonlinear transformation τ and the linear transformation L. The transformation could be described as follows:

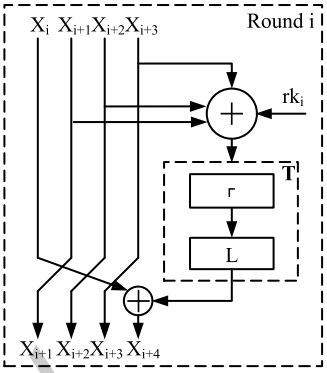

Fig. 1. The i-th round of SM4.

$$T(.) = L(\tau(.)). \tag{5}$$

The nonlinear τ is composed of four 8-bit nonlinear S-boxes in parallel. Let $A = (a_0, a_1, a_2, a_3) \in (GF(2^8))^4$ present the input of τ and $B = (b_0, b_1, b_2, b_3) \in (GF(2^8))^4$ denote the output. The τ could be described as follows:

$$\begin{aligned} B &= (b_0, b_1, b_2, b_3) \\ &= \tau(A) \\ &= (S(a_0), S(a_1), S(a_2), S(a_3)). \end{aligned} \tag{6}$$

The S-box look-up table is shown in [1], and the algebraic structure of the S-box is described in Section 4.

The input of the linear transformation L is B, which is the result of τ. This detail could be described as follows:

$$\begin{aligned} C = L(B) = B \oplus (B <<< 2) \oplus (B <<< 10) \\ \oplus (B <<< 18) \oplus (B <<< 24). \end{aligned} \tag{7}$$

In the decryption operation, SM4 could be performed in the same manner as the encryption procedure by reversing the order of the rk_i. For example, the order of the rk_i is $(rk_0, rk_1, ..., rk_{31})$ in the encryption operation, but $(rk_{31}, rk_{30}, ..., rk_0)$ in the decryption operation.

The round operation of the SM4 is shown in Fig. 1.

3.3. Key expansion

The key expansion in the SM4 cipher has the same structure as that in the encryption/decryption process,

except for the L operation. The method of generating a round key could be described as follows:

$$(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3) \tag{8}$$

and

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)) \tag{9}$$

where i = 0,1,...,31. FK_i, $i \in 0, 1, 2, 3$ represents system parameters, which are defined as 0xA3B1BAC6, 0x56AA3350, 0x677D9197, and 0xB27022DC. CK_i, $i \in 0, 1, 2, 3$ indicates key constants, which could be defined as $CK_i = (ck_{i,0}, ck_{i,1}, ck_{i,2}, ck_{i,3}) \in (GF(2^8))^4$. $ck_{i,j}$ could be calculated as follows:

$$ck_{ij} = (4i + j) * 7 (mod 256) \tag{10}$$

where i = 0,1,...,31 and j = 0,1,2,3. In Equation (9), the T' is a transformation similar to T in the encryption operation. The difference between T' and T is the linear operation, which is defined as L' and could be presented as follows:

$$L'(B) = B \oplus (B <<< 13) \oplus (B <<< 23). \tag{11}$$

4. Improved SM4 algorithm

In the following subsections, we explain in detail our design and the implementation of the SM4 algorithm resistant to power analysis. In Section 4.1, we outline the algebraic structure of the SM4 S-box. In Section 4.2, we discuss the mask technology for SM4. In Section 4.3, we develop the mask and the logic implementation of the SM4 algorithm.

4.1. Algebraic structure of S-box

In [1], the origin of the S-box is not described. Most algorithm designers use a look-up table to implement the S-box, which consists of 256 entries, each 8-bit wide. However, this design consumes too much hardware resources. By contrast, [20] proposed a algebraic structure of the SM4 S-box, which is composed of two linear affine transformations and a nonlinear modular inversion over the Galois Field (GF). The algebraic structure is more suitable for hardware implementation. The algebraic form of the S-box could be described as follows:

$$S(x) = I(x \cdot A_1 + C_1) \cdot A_2 + C_2. \tag{12}$$

The cyclic matrices A_i, $i \in (1, 2)$ in Equation (12) are

$$A_1 = A_2 = \begin{pmatrix} 1&1&1&0&0&1&0&1 \\ 1&1&1&1&0&0&1&0 \\ 0&1&1&1&1&0&0&1 \\ 1&0&1&1&1&1&0&0 \\ 0&1&0&1&1&1&1&0 \\ 0&0&1&0&1&1&1&1 \\ 1&0&0&1&0&1&1&1 \\ 1&1&0&0&1&0&1&1 \end{pmatrix} \tag{13}$$

and the row vectors are

$$C_1 = C_2 = (1, 1, 0, 0, 1, 0, 1, 1). \tag{14}$$

The operation $I(.)$ is a modular inversion over $GF(2^8)$. The irreducible polynomial is

$$M(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1. \tag{15}$$

4.2. Masking of SM4

Power analysis is based on the power consumption of a cryptographic chip, which depends on the intermediate results in the operational processes. Therefore, mask technology can be used to randomize the intermediate results and reduce or eliminate this dependence.

The advantages of mask technology are that the algorithm is straightforward and can be quickly implemented at the algorithm level. According to this theory, the power consumption characteristics of the cryptographic chip are not changed, meaning power consumption still depends on the intermediate results. However, the dependence is not correlated with the correct key. To date, mask technology has been widely applied in essential practice.

Similar to the AES block cipher, the operation of SM4 can also be summarized into three types: round key function, nonlinear S-box, and linear transformation. In the masking process, the linear transformation can be implemented by the exclusive or operation. However, modular inversion in the S-box is a nonlinear operation, and we need to design an additional logic circuit for masking.

Trichina [18] proposed a mask algorithm for AES based on multiplicative masks. The major advantage of this algorithm is that all the data are additively masked in the $GF(2^8)$ inversion, which is not safe

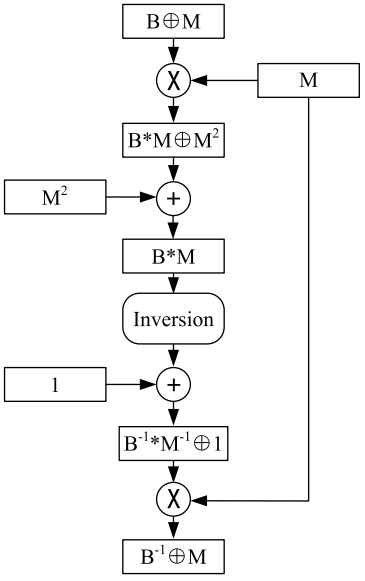

Fig. 2. The principle of mask algorithm.

under common DPA attacks. This method has easy implementation in hardware. The algorithmic details are shown in Fig.2.

As shown in Fig.2, this mask algorithm includes two multiplications and one square arithmetic. In this manner, all the intermediate results, as well as the input and output, are additively masked. Let the input data be $B + M$, where B is the original input data and M is the mask over $GF(2^8)$. Compared with the other mask algorithms proposed by Akkar [12], this mask algorithm is much faster and the hardware cost is smaller.

However, this method for AES can neither blur all of the intermediate results nor resist zero attack because mask technology involves multiplication masks [13]. For example, when $B = 0$, the input and output of the modular inversion over $GF(2^8)$ are zero. The details could be described as follows:

$$B{*}M = (B{*}M)^{-1} = 0. \tag{16}$$

In this case, the modular inversion over $GF(2^8)$ is not correlated with the mask.

4.3. Improved SM4 cipher

In this subsection, a SM4 cipher is proposed using the mask algorithm, which is described in subsection 4.2. To achieve security, we improved the SM4 cipher and the mask algorithm. The details of our algorithm are described in this subsection.

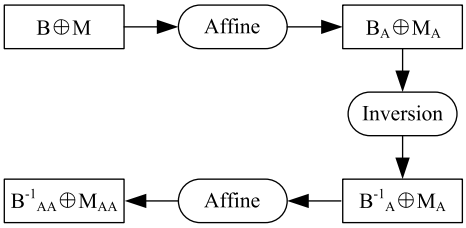

Fig. 3. The processes of mask S-box.

4.3.1. Mask S-box

In this paper, our mask principle is based on the mask algorithm, which was proposed by Trichina [18]. The S-box is implemented by the algebraic structure, details of which were described in Section 4.1. The processes of the mask S-box is shown in Fig. 3.

As previously deduced in Section 4.1, the mask processes over GF could be described as follows

$$\begin{aligned}
&(B + M) * A_1 + C_1 \\
&\rightarrow I((B + M) * A_1 + C_1) \\
&\rightarrow I(B * A_1 + C_1) + M * A_1 \\
&\rightarrow (I(B * A_1 + C_1) + M * A_1)A_2 + C_2 \\
&\rightarrow I(B * A_1 + C_1) * A_2 + C_2 + M * A_1 * A_2 \\
&\rightarrow S(B) + MA_1A_2
\end{aligned} \tag{17}$$

where M is a 32-bit random mask. The result is composed of the S-box of input added with a function of the random mask. The mask part could be removed by a simple circuit module, which generates the MA_1A_2. The round operation of the SM4 with a random mask is shown in Fig. 4.

As shown in Fig. 4, X'_i is equivalent to $X_i \oplus M$, and τ_m denotes the S-box with the mask. To guarantee that all data are masked, we first calculate the result of $X_i \oplus T(.)$. Consequently, all the intermediate data are masked and can resist power analysis. The detail of round function could be described as follows:

$$\begin{aligned}
X'_{i+4} &= X'_i \oplus L(\tau_m(X' \oplus rk_i)) \oplus L(M') \\
&= X'_i \oplus L(\tau_m(X \oplus rk_i \oplus M)) \oplus L(M') \\
&= X'_i \oplus L(\tau(X \oplus rk_i \oplus M')) \oplus L(M') \\
&= (X_i \oplus M) \oplus (T(X \oplus rk_i)) \\
&= X_{i+4} \oplus M \\
X' &= X'_{i+1} \oplus X'_{i+2} \oplus X'_{i+3}
\end{aligned} \tag{18}$$

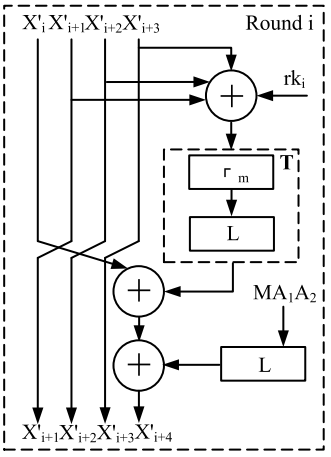

Fig. 4. The i-th round of SM4 with mask.

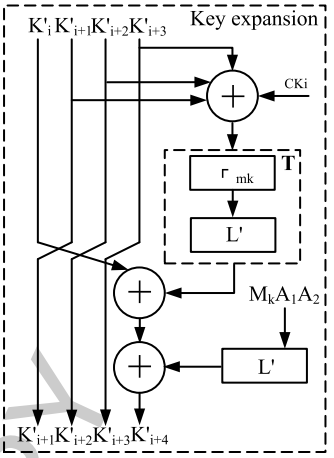

Fig. 5. The key expansion with mask.

$$X = X_{i+1} \oplus X_{i+2} \oplus X_{i+3}$$
$$M' = MA_1A_2.$$

After 32 rounds of calculations, we can obtain the cipher that has been masked

$$\begin{aligned}(Y'_0, Y'_1, Y'_2, Y'_3) &= (X'_{35}, X'_{34}, X'_{33}, X'_{32}) \\ &= (X_{35} \oplus M, X_{34} \oplus M, \\ &\quad X_{33} \oplus M, X_{32} \oplus M).\end{aligned} \tag{19}$$

After this operation, we add exclusive or logic and can obtain the unmask cipher as follows:

$$\begin{aligned}Y &= (Y_0, Y_1, Y_2, Y_3) \\ &= (X'_{35} \oplus M, X'_{34} \oplus M, X'_{33} \oplus M, X'_{32} \oplus M) \\ &= (X_{35}, X_{34}, X_{33}, X_{32}).\end{aligned} \tag{20}$$

4.3.2. Mask Key Expansion

In Section 3.3, we described the key expansion in detail. Similar to the round function, the key expansion also has the round function F, which is composed of the nonlinear transformation τ and the linear transformation L'. In the past, the main focus was the security of the encryption process, but attention must also be paid to the key expansion. In this paper, we add a mask algorithm to the key expansion.

This part uses the same mask algorithm described in Section 4.2, but a different mask, denoted as M_k, is added. The round function of key expansion is shown in Fig. 5. The key expansion has the same structure as the encryption/decryption process, except for the L operation. Consequently, we need to replace the L transformation with the L'. The K'_i is equivalent to $K_i \oplus M_k$. Finally, we could obtain K'_{i+4}, which is the round key rk_i. In this case, the round key is concealed by masking and greatly enhances the difficulty of analysis.

4.3.3. Improved Mask Logic

In this section, we propose strategies to improve the mask scheme in our design. For example, the mask algorithm [18] can neither blur all of the intermediate results nor resist zero attack because mask technology involves a multiplication mask. When the intermediate result is zero, the mask of a result is zero. The details are presented in Equation (16).

In subsections 4.3.1 and 4.3.2, we proposed a mask scheme for encryption operation and key expansion. Based on the mask algorithm proposed by Trichina [18], our algorithm could blur all of the intermediate results and only remove the mask when the last round operation is finished. Furthermore, mask updating logic is added so that we could use a different mask in every round operation. To prevent the unmasking of operands, we must introduce another additive mask before the new mask is added.

As shown in Fig. 6, an additional exclusion or logic is used to update the mask. The mask update could be obtained as follows:

$$\begin{aligned}X'' &= X' \oplus M' \\ &= X \oplus M_{new} \\ M' &= M_{old} \oplus M_{new}.\end{aligned} \tag{21}$$

With this, the mask can be updated in every round operation, and the operands would be masking.

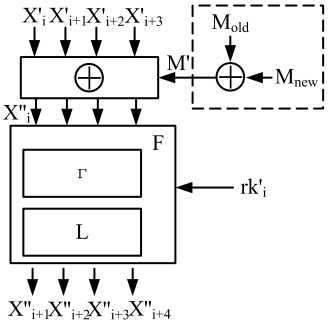

Fig. 6. The round function with update mask logic.

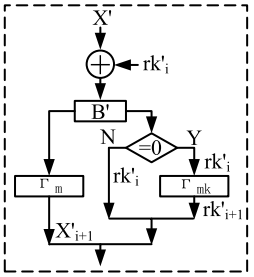

Fig. 7. Improved SM4 mask process.

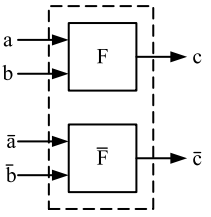

Fig. 8. The structure diagram of complementary logic.

The basic problem with the mask algorithm proposed by Trichina is that it does not mask the all-zero byte value of data. That is, the all-zero byte remains unchanged after masking [13]. On the other hand, the all-zero data bytes cannot be avoided in AES. In the first round of the masked AES, the vulnerable intermediate variables are the input byte B and the output byte $Inversion(B * M)$ of the inversion $GF(2^8)$. Note that the data byte B is given as $B = P \oplus K$, where P and K are the corresponding plain text and key byte respectively. It follows that

$$K = P \Rightarrow B = 0 \Rightarrow Inv(B * M) = 0. \qquad (22)$$

In this case, the DPA on the masked AES can be mounted in essentially the same way as the original AES without masking. The difference is that one has to target all-zero input byte or, equivalently, the all-zero output byte of $Inversion(.)$. In other words, for each of the 256 possible values of the corresponding expanded key byte K, the power consumption traces for $P = K$ are extracted and used to identify the correct key. The question is whether the described DPA can be prevented by some other implementation of the multiplicative masking.

This study proposes an approximate, non-ideal solution to the problem, which is based on balancing the power consumption. According to the description in Section 3, the key expansion in SM4 has the same S-box structure as that in the encryption process. Therefore, we could use this structure to reduce the correlation between zero and power consumption.

The Fig. 7 shows that a switching log-in is added in our implementation. First, we would determine the value of B when the function performs an S-box operation. If $B = 0$, we would simultaneously perform the round function of key expansion. Otherwise, the algorithm would directly perform the next round function. The idea of this algorithm design originates from the WDDL circuit model [21], namely, complementary logic. The structural diagram is shown in Fig. 8.

Power consumption is only drawn from the power supply when the 0 to 1 output transition occurs. Therefore, when the power dissipation of the smallest building block is constant and independent of the signal activity, no information is leaked through the power supply. Our logic flow is based on a constant power dissipating logic: in one clock cycle, the power consumption of each individual logic gate is constant and independent of its input data. In other words, when $B = 0$ and $B \neq 0$, the operation of $Inversion(.)$ would draw the same power consumption. To balance the power consumption, two conditions must be satisfied in our design. First, the complementary logic has the same logic form of implementation as the τ function in encryption. Second, the circuit must perform the same operation.

As previously described, similar to the round function, the key expansion also has a round function F, which is composed of the nonlinear transformation τ and the linear transformation L'. Therefore, the key expansion is used as complementary logic, as shown in detail in Fig. 7. If $B = 0$, the circuit logic performs an additional τ operation, such that it could reduce the power consumption difference generated by the zero byte.

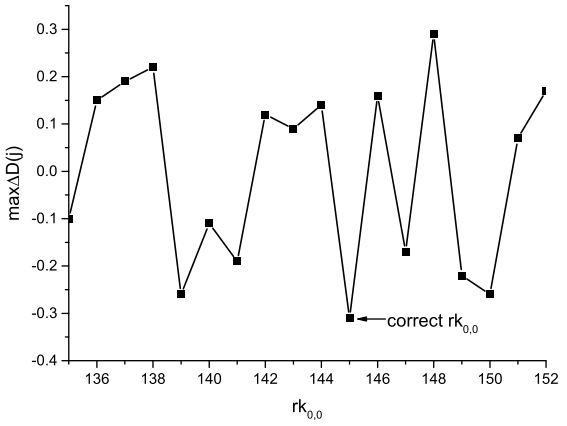

Fig. 9. DPA attack on the ordinary SM4 algorithm.

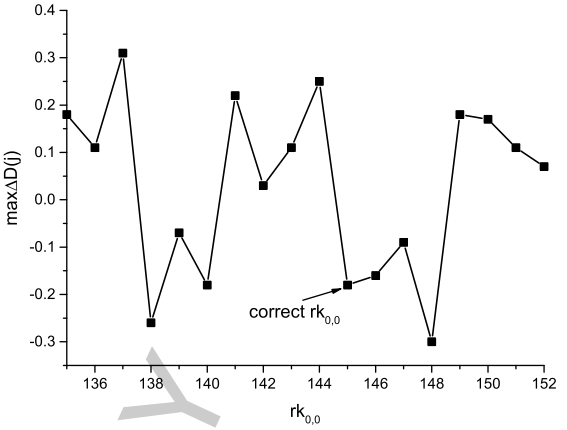

Fig. 10. DPA attack on the improved SM4 algorithm.

Table 1
Area of SM4

	original	improved
Number of ports	235	235
Number of nets	4163	4264
Number of cells	3543	3653
Number of references	76	76
Combinational area	149249	173869
Nocombinational area	143694	143528
Total cell area	292943	317397

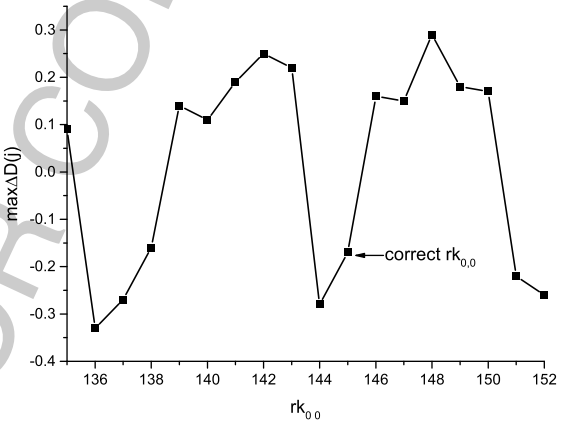

Fig. 11. Zero-attack on the improved SM4 algorithm.

Table 2
Power of SM4

	original	improved
Cell internal power	2.038mW	2.049mW
Net switching power	0.114mW	0.114mW
Total dynamic power	2.152mW	2.163mW

5. Experiment results

The experiment results are described in this section.

The SM4 design described above was implemented on an application specific integrated circuit (ASIC) chip particularly designed for power analysis evaluation. Instantaneous power consumption traces are collected by a series of Synopsys software. First, we utilize the Synopsys design compiler software to synthesize the ASIC chip and obtain the netlist file of the chip. Second, we use a Synopsys Verilog-compiled simulator to simulate the netlist file. This simulation can generate the value change dump (VCD) file that documents detailed information on the SM4 operation process. Third, we input the VCD file into prime time software. The fast signal database (FSDB) file is produced through analysis and Huffman encoding. Fourth, we employ CosmosScope software to extract power information from the FSDB file.

Using the GSMC 0.18 μm CMOS technology, we complete the logic synthesis and physical design of the ordinary SM4 module and the improved SM4 module, where the S-box of the ordinary SM4 module is implemented as a look-up table. Let the encryption key be the same as that used in [1], then $MK = (0x0123456789abcdeffedcba9876543210)$.

This paper analyzes the S-box because it denotes the only nonlinear SM4 function. According to the DPA method described in Section 2, DPA attacks the four bytes of the round key. In our experiment, we collected 1 million power traces. The results of our algorithm are performed as follows.

Fig. 9 and Fig. 10 show the correlation coefficients along the sample for every hypothesis. As shown in Fig. 9, the correct assumption is $rk_{0,0} = 0x91$ because the correlation coefficient at this value is greater than the other values. However, given the

improved SM4 cipher proposed in this paper, no fluctuations are obvious in Fig. 10. As expected and previously described, this design can resist the DPA. The experimental results shown in Fig. 11 indicate that this design can resist zero-value attack because the correct key has a lower peak than the other values.

This paper presents a method for securing a SM4 cipher resistant to the DPA. Comparisons of the ordinary SM4 cipher and the improved SM4 cipher are shown in Table 1 and 2 The proposed design is resistant to DPA, although it increases the hardware implementation area to a little extent. Moreover, the design is resistant to zero-value attack because we added switching logic to our design.

6. Conclusion

In this paper, we analyzed the structure of the SM4 cipher and proposed a logic design as a countermeasure to power analysis. Unlike the multiplicative mask, our algorithm can resist the zero-value attack. In addition, module reuse and composite field computation were performed to blur all of the intermediate results. The theoretical analysis and simulation results of our chip suggest that the proposed approach is efficient.

Future work will examine the weak points and immunity of these methods to higher-order power analysis.

Acknowledgments

The research was partially funded by the Key Program of National Natural Science Foundation of China (Grant Nos.61133005, 61432005), the National Natural Science Foundation of China (Grant Nos. 61370095, 61472124, 61402400), the International Science and Technology Cooperation Program of China (Grant No. 2015DFA11240) and Hunan Normal University of Youth Science Foundation (Grant No.11404).

References

[1] 2012, Office of State Commercial Cipher Administration of China, GM/T 0002-2012 SM4 block cipher[S].

[2] P. Kocher, J. Jaffe and B. Jun, Differential power analysis, *Advances in CryptologyąłCRYPTOąŕ99,* 1999, pp. 388–397.

[3] H. Kim, S. Hong and J. Lim, A fast and provably secure higher-order masking of aes s-box, *Cryptographic Hardware and Embedded SystemsÍCCHES 2011*, 2011, pp. 95–107.

[4] T. Güneysu and A. Moradi, Generic side-channel countermeasures for reconfigurable devices, *Cryptographic Hardware and Embedded SystemsÍCCHES 2011*, 2011, pp. 33–48.

[5] M. Rivain and E. Prouff, Provably secure higher-order masking of aes, *Cryptographic Hardware and Embedded Systems, CHES 2010*, 2010, pp. 413–427.

[6] M. Avital, H. Dagan, O. Keren and A. Fish, Randomized multitopology logic against differential power analysis, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* **24**(3) (2015), 702–711.

[7] A. Moradi, A. Poschmann, S. Ling, C. Paar and H. Wang, Pushing the limits: A very compact and a threshold implementation of aes, *Advances in Cryptology–EUROCRYPT 2011*, 2011, pp. 69–88.

[8] D. Suzuki, M. Saeki, K. Shimizu, A. Satoh and T. Matsumoto, A design methodology for a dpa-resistant circuit with rsl techniques, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **93**(12) (2010), 2497–2508.

[9] J.-L. Danger, S. Guilley, S. Bhasin and M. Nassar, Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors, *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, 2009, pp. 1–8.

[10] M. Bucci, L. Giancane, R. Luzzi and A. Trifiletti, A flip-flop for the dpa resistant three-phase dual-rail pre-charge logic family, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* **20**(11) (2012), 2128–2132.

[11] A.-T. Hoang and T. Fujino, Hybrid masking using intra-masking dual-rail memory on lut for sca-resistant aes implementation on fpga, *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2013, pp. 266–267.

[12] M.-L. Akkar and C. Giraud, An implementation of des and aes, secure against some attacks, *Cryptographic Hardware and Embedded Systems, CHES 2001*, 2001, pp. 309–318.

[13] J.D. Golić and C. Tymen, Multiplicative masking and power analysis of aes, *Cryptographic Hardware and Embedded Systems-CHES 2002*, 2002, pp. 198–212.

[14] T.S. Messerges, Securing the aes finalists against power analysis attacks, *Fast Software Encryption, Springer Berlin Heidelberg*, 2001, pp. 150–164.

[15] H. Chang and K. Kim, Securing aes against second-order dpa by simple fixed-value masking, *Computer Security Symposium*, 2003, pp. 145–150.

[16] J.-S. Coron, Higher order masking of look-up tables, *Advances in Cryptology-EUROCRYPT 2014*, 2014, pp. 441–458.

[17] J. Zeng, Y. Wang, C. Xu and R. Li, Improvement on masked s-box hardware implementation, *Advances in Cryptology-EUROCRYPT 2014*, 2012, pp. 113–116.

[18] E. Trichina, D. De Seta and L. Germani, Simplified adaptive multiplicative masking for aes, *Cryptographic Hardware and Embedded Systems-CHES 2003*, 2003, pp. 187–197.

[19] E. Oswald, S. Mangard, N. Pramstaller and V. Rijmen, A sidechannel analysis resistant description of the aes s-box, *Fast Software Encryption*, 2005, pp. 413–423.

[20] F. Liu, W. Ji, L. Hu, J. Ding, S. Lv, A. Pyshkin and R.-P. Weinmann, Analysis of the sms4 block cipher, *Information Security and Privacy*, 2007, pp. 158–170.

[21] E. Amouri, H. Mehrez and Z. Marrakchi, Impact of dual placement and routing on wddl netlist security in fpga, *International Journal of Reconfigurable Computing* (2013), 8.