Journal of Electronic Testing (2023) 39:435–445
https://doi.org/10.1007/s10836-023-06076-5

New Second-order Threshold Implementation of Sm4 Block Cipher

Tianyi Shao[1,2] · Bohua Wei[2,3] · Yu Ou[1,2] · Yongzhuang Wei[1,2] · Xiaonian Wu[1,2]

Received: 22 December 2022 / Accepted: 10 July 2023 / Published online: 4 August 2023

Abstract
As SM4 block cipher has become an ISO/IEC international encryption standard in June 2020, the security of SM4 against side-channel analysis (SCA) is highly valued by academic community. Threshold implementation (TI) scheme is a common countermeasure against SCA. However, the implementation of a high-order TI scheme can be costly. How to improve the resistance of SM4 implementation against high-order SCA without significant increasing the cost appears to be an important task. In this article, a new SM4 second-order TI scheme is proposed based on the tower field decomposition of 8-bits inverter. In more detail, by performing the tower field decomposition twice in the SM4 S-box, the inverse and multiplication operations on finite field are transformed into inverse and multiplication operations on tower field, thus reducing the algebraic order of the decomposed S-box from 7 to 2. Then, the design and implementation of our scheme with 3 shares is illustrated based on the decomposed S-box. Compared with the best-known TI of the S-box in the SM4, our scheme uses smaller number of register stages. The circuit area of S-box is reduced by 48.6%. The number of fresh randomness required in a single round operation is 96 bits. Moreover, both the second-order t-test with 10 million power traces and the correlation power analysis are performed, thus verifying the second-order security of this scheme.

Keywords Block cipher · Side-channel analysis · SM4 block cipher · Threshold implementation · Fresh randomness

1 Introduction

Side-channel analysis (SCA) proposed by Kocher et al. has brought serious threats to cryptographic devices [13]. The core idea of SCA is that any adversaries can gain the sensitive information by monitoring the physical leakage of a cryptographic implementation and the secret key can be further recovered with the sensitive information. Indifference to the traditional attacks, the SCA can be performed on any encryption algorithm implementation or hardware device. For instance, simple power analysis (SPA), differential power analysis (DPA) [13], and correlation power analysis (CPA) [4]. Recently, some high-order power analysis (HOPA) methods are being increasingly used in SCA [26].

During the past 2 decades, with the rapid development of SCA, various countermeasures against SCA are proposed. At chip level, random noise is introduced into the chip's power supply [8]. At algorithm level, masking schemes are used to randomize the intermediate values [1]. At gate level, the dual-rail logic circuits are performed to balance the energy consumption of different operations during the computation [10]. Actually, among the side-channel protection methods, the Boolean masking has been widely applied. Since Boolean masking is performed on algorithm implementation, which has no effect on the circuit synthesis, layout, or any other processes, it is more suitable for hardware devices [2, 25]. Later, in 2003, ISW masking for the probing security model is presented by Ishai et al. [11], where the trichina gate for measuring the order of SCA attacks are considered. However, probing model could be invalid in combinatorial circuit due to the phenomenon (called glitches). To fix this problem, Nikova et al. proposed the threshold implementation (TI), which is the first probing secure masking implementation. In particular, the impact of glitches are taken into

Responsible Editor: S. Bhunia

✉ Bohua Wei
wei_bh@163.com

[1] Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, Guangxi, China

[2] Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

[3] Guangxi Wangxin Information Technology Co., Ltd., Nanning 530000, Guangxi, China

account [18]. In fact, the TI is based on the secure techniques for multi-party computation and secret sharing, where sensitive information is divided into multiple sharings such that the leaked information independent of the sensitive data. To guarantee its security, three properties were introduced to TI by Nikova at 2011 [17]. In classical TI, at least $t \times d + 1$ of input shares are required to ensure its security, where t is defined as the algebraic degree of the Boolean function, and d stands for the security order of masking. On the other hand, by introducing fresh randomness into the scheme, a scheme with $d + 1$ input shares can reach the same security order as a classical one does [9, 20]. To reduce the needs of fresh randomness, an approach named "changing of the guards" is proposed to reuse the randomness [6]. Later, an improvement of changing of the guards is given by Dhooghe [7]. Moreover, Bilgin extended the 1st-order TI to higher-order TI in [3]. Recently, a higher-order TI with almost no fresh randomness is proposed by Shahmirzadi in [23].

The SM4 block cipher was released for encryption data in wireless networks environment in 2006 [12]. Later, it is further selected as an ISO/IEC international encryption standard in June 2020. The security of the SM4 block cipher against SCA is extensively received attention by academic community. In the first place, the algebraic structure of SM4 is described by Liu et al. in 2007, where the similarity of the S-box between SM4 and AES is verified [16]. Then the TI methods used to protect AES can also be applied to SM4 cipher. Based on the composite field theory and TI, a new masked scheme for the S-box of SM4 is proposed [15]. Actually, this TI of SM4 is still limited to the first-order protection. Moreover, the second-order security S-box against SCA is investigated by Li et al. in 2018 [14]. Recently, an unbalanced sharing of TI for SM4 cipher is also proposed to reduce the number of input shares [28]. Notice that the first-order protection of the SM4 implementation is still verified to be vulnerable via some higher-order SCA attacks (e.g., high-order correlation power analysis). How to improve the resistance of the SM4 implementation against higher-order SCA without significant increasing the implementation consumption appears to be an important task. In particular, there is still no second-order protection suggested for the whole SM4 block cipher so far.

In this article, a masking scheme of SM4 block cipher is proposed for second-order security against SCA. Based on the theory of tower field decomposition, the element over $GF(2^8)$ is mapped onto $GF(((2)^2)^2)^2$ to reduce the difficulty of design. A 3-shared S-box with fresh randomness is designed and the linear transform is masked with shared-wise approach. As a result, this scheme requires about 16 k gate equivalence (kGE) with UMC 130 standard cell library, and the consumption of fresh randomness is 96 bits. As the implementation is designed with round-based method, it requires 256 clock cycle for one encryption. 10 million power traces from this second-order protection of SM4 cipher are collected for both the t-test and correlation power analysis. The experimental results illustrate that our work reaches the second-order security level.

1.1 Organization

This article is organized as follows: In Sect. 2, the introduction of definitions, SM4 cipher and threshold implementation is provided. In Sect. 3, The encryption circuit of masked SM4 and its decomposition are described. In Sect. 4, the specific masking scheme of the S-box of SM4 cipher is proposed. To illustrate the advantages of our scheme, an evaluation is carried out in Sect. 5. The t-test and CPA experiments illustrate the second-order security of SM4 cipher. The conclusion of this work is given in Sect. 6.

2 Preliminaries

In this section, the preliminary notations and definitions are introduced. The SM4 block cipher and threshold implementation are described as well.

2.1 Notations and Definitions

The binary variables over $GF(2)$ is denoted with lowercase italic fonts (e.g. $x \in GF(2)$) and the vectors over $GF(2^n), n > 2$ is denoted with upper-case italic fonts (e.g. $X \in GF(2^n)$). The subscript of variables indicates its position in the vector. A vector can be written as a bitstring and the most significant variable is in the leftmost position, i.e., $X = (x_{n-1}, \cdots, x_1, x_0)$. The superscript of variables and vectors like X^i indicates the i-th sharing of vector X. $X <<< n$ is defined as a cyclic shift of vector X by n bits to the left.

2.2 SM4 Block Cipher

SM4 cipher uses an unbalanced Feistel structure with the 128-bits data block and 128-bits secret key, which is composed of 32 round function operations [16]. In the i-th round, a 32-bits subkey K_i is applied, and the internal state can be represented by $S = (S_1, S_2, S_3, S_4)$, where the $S \in GF(2^{128}), S_i \in GF(2^{32})$.

The round function F of i-th round can be defined as:

$$\begin{aligned} &F : GF(2)^{128} \times GF(2)^{32} \rightarrow GF(2)^{128} \\ &(S_i, S_{i+1}, S_{i+2}, S_{i+3}, K_i) \rightarrow (S_{i+1}, S_{i+2}, \\ &S_{i+3}, S_i \oplus L(\tau(S_{i+1} \oplus S_{i+2} \oplus S_{i+3} \oplus K_i))) \end{aligned} \tag{1}$$

The linear transformation L can be defined as:

$$
\begin{aligned}
&L : GF(2)^{32} \rightarrow GF(2)^{32} \\
&X \rightarrow X \oplus (X <<< 2) \oplus (X <<< 10) \oplus \\
&(X <<< 18) \oplus (X <<< 24)
\end{aligned} \tag{2}
$$

The input X of linear transform L is generated by mixing up the state (S_2, S_3, S_4) with subkey K_i:

$$X = S_2 \oplus S_3 \oplus S_4 \oplus K_i \tag{3}$$

Function τ represents the S-box of SM4:

$$
\begin{aligned}
&\tau : GF(2)^{32} \rightarrow GF(2)^{32} \\
&X \rightarrow (Sbox(X_{[31\cdots24]}), Sbox(X_{[23\cdots16]}), \\
&Sbox(X_{[15\cdots8]}), Sbox(X_{[7\cdots0]}))
\end{aligned} \tag{4}
$$

To keep the consistence of encryption and decryption, an inverse transform is performed at the end of encryption as:

$$(X_{33}, X_{34}, X_{35}, X_{36}) \rightarrow (X_{36}, X_{35}, X_{34}, X_{33}) \tag{5}$$

2.3 Threshold Implementation

Combined with secret sharing and multi-party computation, the concept of TI was firstly proposed in 2006 [18], where the input variables are masked with random numbers. The masking order is defined in such a way that the d-order masking should be resisted to d-order differential power attacks, i.e., most of the d probes can be put onto the gate output. Any combination of them should not reveal any information about the secret [22]. To ensure the resistance against side-channel analysis, a TI scheme should satisfy the three properties below.

2.3.1 Correctness

For the same input variables, the result of an unmasked cipher implementation and the result of a masked one should be consistent. In other words, the sum of all the input shares should be the input variables, at the same time, the sum of all the output shares should be the output variables.

2.3.2 Non-Completeness

To resist the d-order power analysis, any combination of d output of component functions should be independent of at least one input variable. When the security order against power analysis increases, the number of input shares will increase as well.

2.3.3 Uniformity

A masking scheme is said to be uniform if each of its output shares can be generated with the same probability for both the input and output variables.

In the classical threshold implementation without fresh randomness, a masking scheme can satisfy the three properties for d-order side-channel security if the number of input shares is:

$$S_{in} \geq t \times d + 1 \tag{6}$$

while t is the algebraic degree of the Boolean function.

The number of output shares should be:

$$S_{out} \geq \binom{S_{in}}{t} \tag{7}$$

Shahmirzadi et al. pointed out that we can design a d-order SCA security scheme with $d + 1$ input shares by partitioning the Boolean function into several isolated component functions and introducing fresh random numbers [24].

On the other hand, since the basic component of modern hardware devices is CMOS, the glitches are inevitable. The designer needs to set the register layers to store the intermediate value to stop the glitches from propagating along the circuit.

High-order masking schemes can resist SCA attacks effectively. However, higher-order masking schemes often means more circuit area, randomness, and clock cycles. How to make the trade-off between area, randomness requirements and latency appears to be an important task.

3 Encryption Circuit Design

In this section, the design of the second-order TI protection scheme for SM4 encryption circuit is presented.

3.1 Overview of Hardware Circuit

The circuit architecture for the second-order masked SM4 cipher is shown in Fig. 1.

The encryption circuit of SM4 cipher can be decomposed into some modules: finite state machine, key and state registers, round function, parameter generator, randomness generator, multiplexer, and reverse output, where the finite state machine is used to generate control signals. The flow of control signals is indicated by dashed lines with arrows in the figure, and the flow of data is indicated by solid lines with arrows.

3.2 Masking State and Key

The 128-bits state and key are shared with three-shared components, requiring a total of $128 \times 2 \times 2 = 512$ bits of randomness. After each round of encryption, the intermediate values need to be stored in registers. The value of key

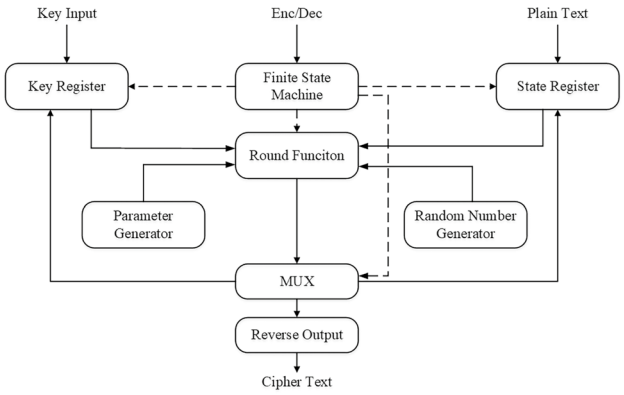

Fig. 1 Overview of the second-order masked SM4 hardware circuit

and state registers update according to the control signals generated by the finite state machine.

3.3 Round Function

In the encryption circuit of SM4 cipher, the length of input for the round function is 32, so four identical 8-bits S-boxes are applied to execute in parallel to improve the efficiency of operation. For the unmasked SM4, the input data $S_{in} = x + y + z + rk_i$, where x, y, z represents the 32-bits plaintext of SM4, and rk_i is the 32-bits round key for round i. The formula for the input data is a linear function and the same function can be applied independently to each shared component. In this case, it can be masked by a share-wise approach with the masking scheme shown in (8).

$$\begin{aligned} S_{in}^1 &= x^1 + y^1 + z^1 + rk_i^1 \\ S_{in}^2 &= x^2 + y^2 + z^2 + rk_i^2 \\ S_{in}^3 &= x^3 + y^3 + z^3 + rk_i^3 \end{aligned} \tag{8}$$

Note that the algebraic expression for the SM4 S-box is given by Liu et al. in (9).

$$S_{out} = I(S_{in} \cdot A_1 + C_1)A_2 + C_2 \tag{9}$$

The symbol I represents the inverse operation on the finite field $GF(2^8)$, which is the only nonlinear operation in SM4 cipher. It is usually difficult to design the scheme for a nonlinear Boolean function. The specific scheme designed for the S-box of SM4 will be further introduced in section 4.

3.4 Key Schedule

The masking for the key schedule is similar to the masking of the encryption operation, i.e., linear components are masked using the three-input shared-wise method and the S-box is masked using a three-input component with fresh randomness.

3.5 Randomness Generator

In our new protection scheme, random numbers are used to share sensitive information so that the uniformity of output during the operation can be ensured. In general, the masking scheme dynamically generates random numbers with 31-bits linear feedback shift registers built into Xilinx FPGAs [23]. In this section, a 256-bits nonlinear feedback shift register is used to generate random numbers with better properties, which are seeded by a true random number generator from the FPGA when it is set up.

4 Masked S-box Design

The S-box is the only nonlinear component in SM4 cipher, which is the primary target for power analysis as well. Due to its high algebraic degree of the 8-bits S-box, it poses a great challenge for masking designing, especially for high-order masking protection. To overcome this difficulty, an optimized structure of the SM4 S-box is given. Based on the tower field implementation, the S-box is decomposed into some small components. Moreover, the masking protection scheme is constructed by basing on these small components.

4.1 Optimization of S-box Circuit

Note that Canright proposed the S-box implementation approach with tower field decomposition of S-box [5]. Based on the isomorphic mapping relationship between the finite field and the tower field, the core idea of this approach is described below:

1. Looking for an appropriate tower field so that it is isomorphic to the finite field used by the cipher algorithm.
2. Mapping the input element over the finite field into the tower field, and then the inverse operation of tower field is used to calculate the inverse operation of the finite field.
3. Mapping the results in the second step into the original finite field.

Generally, the complexity of the inverse operation over a tower field is much lower than the complexity of the inverse operation over a finite field.

Since there are common factors in the high 4-bits and low 4-bits during the calculation on the tower field, Bai et al. further optimized the circuit of the S-box implemented in the tower field by extracting the common factors and changing the operation order in [2]. The optimized

circuit is shown in Fig. 2. The symbols $\odot$ in the figure indicate that the high and low 4-bits are taken for splitting or combining.

The optimized S-box circuit mainly consists of three components: 4-bits multiplier, 4-bits inverter, and square-v-scaler. For each component, instead of considering its specific design and implementation, we use multiple of them for noise amplification.

Between the components, there are register stages for saving intermediate values to prevent the propagation of glitches.

4.2 Design of 4-bits Multiplier

During the tower field implementation of S-box circuit, the 8-bits input is decomposed into two 4-bits values x and y, which respectively represents the high 4-bits and low 4-bits. Through the linear mapping (affine transformation and isomorphic mapping), the 4-bits output z is obtained by the multiplication operation $z = xy$ on the tower field.

A remasking operation is performed with fresh randomness so that the outputs satisfy the uniformity requirement. Registers are used to store the results of the operation. The output of the square-v-scaler is added to the first component of each output share. The algebraic expression with three-shared components of the masked 4-bits multiplier is given in (10).

$$\begin{cases} z^0 = x^0y^0 + r_1 + [const_1] \\ z^1 = x^0y^1 + r_1 + r_2 \\ z^2 = x^0y^2 + r_2 \end{cases}$$
$$\begin{cases} z^3 = x^1y^0 + r_3 + [const_2] \\ z^4 = x^1y^1 + r_3 + r_4 \\ z^5 = x^1y^2 + r_4 \end{cases} \tag{10}$$
$$\begin{cases} z^6 = x^2y^0 + r_5 + [const_3] \\ z^7 = x^2y^1 + r_5 + r_6 \\ z^8 = x^2y^2 + r_6 \end{cases}$$

The circuit design of the 4-bits multiplier mask scheme is shown in Fig. 3. $(r_1, \cdots, r_6)$ represents 6 groups of fresh randomness with the length of 4 bits, where these fresh randomness are used to ensure that the outputs meet the uniformity requirement. Remasking structure is further designed, where each group uses 3 shares. Due to 6 groups of fresh randomness are used, then the consumption of fresh randomness is reduced by 12 bits.

4.3 Design of 4-bits Inverter

The 4-bits inverter in the tower field S-box implementation can be regarded as a 4×4 S-box whose permutation table is determined by (11)

$$\begin{aligned} y_0 &= x_2x_1x_0 + x_3x_0 + x_1 + x_0 \\ y_1 &= x_2x_1x_0 + x_3x_1x_0 + x_2x_1 + x_3x_0 + x_1 \\ y_2 &= x_2x_1x_0 + x_3x_2x_0 + x_3x_1 + x_0 \\ &\quad + x_1 + x_2 \\ y_3 &= x_2x_1x_0 + x_3x_1x_0 + x_3x_2x_0 + x_3x_2x_1 \\ &\quad + x_2x_0 + x_2x_1 + x_3x_0 + x_1 + x_2 + x_3 \end{aligned} \tag{11}$$

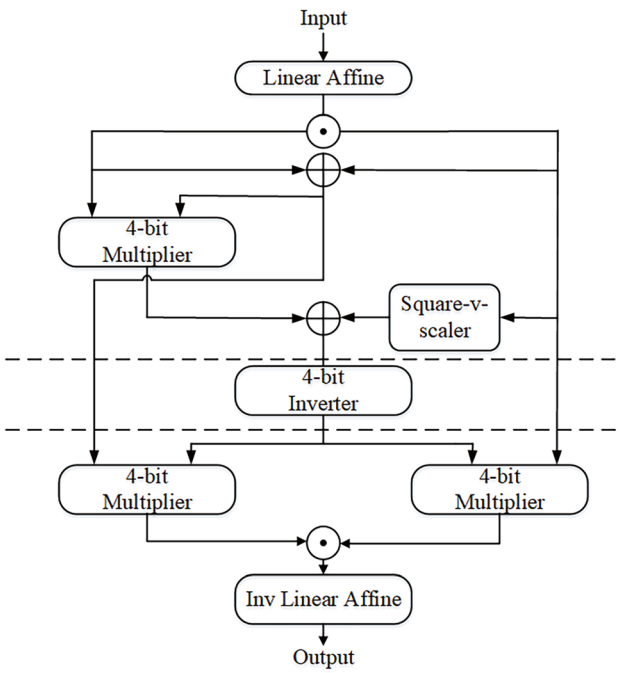

Fig. 2 Optimization of S-box circuit

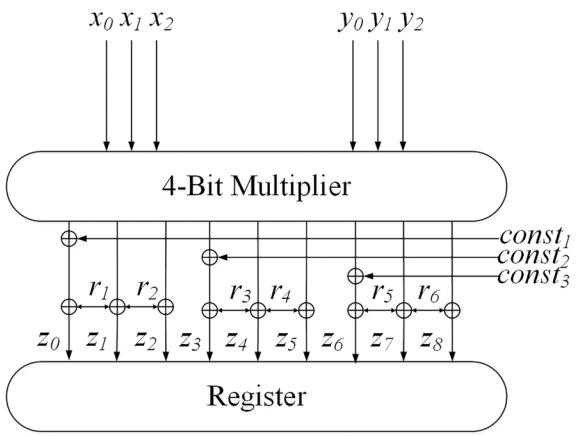

Fig. 3 Structure of 4-bits multiplier in masked S-box

For a smaller S-box such as a 4×4 one, there are some reasonable methods for masking design. To guarantee the second-order security of this scheme, the 4-bits inverter is decomposed by mapping it into a smaller field, thus the circuit design is simplified.

4.4 Design of Square-v-scaler

The square-v-scaler is a 4-bits linear affine which maps the 4-bits input $(x_3, \cdots, x_0)$ to output $(_3, \cdots, x_0)$. By enumerating its inputs, the algebraic expression of the square-v-scaler can be obtained as (12).

$$\begin{aligned} y_0 &= x_3 + x_2 + x_0 \\ y_1 &= x_3 + x_1 + x_0 \\ y_2 &= x_3 + x_2 \\ y_3 &= x_3 \end{aligned} \tag{12}$$

Therefore, the square-v-scaler can be masked by share-wise approach as (13). Three square-v-scalers perform the identical operation in parallel.

$$\begin{cases} y_0^0 = x_3^0 + x_2^0 + x_0^0 \\ y_1^0 = x_3^0 + x_1^0 + x_0^0 \\ y_2^0 = x_3^0 + x_2^0 \\ y_3^0 = x_3^0 \end{cases}$$
$$\begin{cases} y_0^1 = x_3^1 + x_2^1 + x_0^1 \\ y_1^1 = x_3^1 + x_1^1 + x_0^1 \\ y_2^1 = x_3^1 + x_2^1 \\ y_3^1 = x_3^1 \end{cases} \tag{13}$$
$$\begin{cases} y_0^2 = x_3^2 + x_2^2 + x_0^2 \\ y_1^2 = x_3^2 + x_1^2 + x_0^2 \\ y_2^2 = x_3^2 + x_2^2 \\ y_3^2 = x_3^2 \end{cases}$$

5 Implementation and Experiments

In this section, the second-order masking circuit of SM4 cipher is implemented by Verilog, where Xilinx Vivado 2017.4 is used for simulation and verification. In order to illustrate the advantage of our design, some parameters regarding to the circuit area, latency, and fresh randomness of the implementation are compared with previous works. Finally, the t-test and CPA experiments are performed to prove the second-order side-channel security of this design.

5.1 Simulation and Verification

The test vector for SM4 cipher via ECB encryption mode is given in Table 1, all strings are presented in hexadecimal notation. The labels "Plaintext" and "Ciphertext" indicate the standard plaintext and ciphertext of SM4-ECB cipher, while the "Input Block" and "Output Block" mean the input and output strings of our 2nd-order TI scheme. It directly shows that the encryption of this scheme is correct.

Table 1 Test vector for SM4-ECB

Test Vector	Value
Key	0123456789abcdeffedcba9876543210
Plaintext	0123456789abcdeffedcba9876543210
Input Block	0123456789abcdeffedcba9876543210
Output Block	681edf34d206965e86b3e94f536e4246
Ciphertext	681edf34d206965e86b3e94f536e4246

5.2 Performance

In Table 2, the performance of our scheme is compared with the related works (also see [14, 15, 19, 23, 27, 28]).

5.2.1 Fresh Randomness

The consumption of fresh randomness is often ignored when evaluating the performance of scheme. However, the generation of fresh randomness commonly leads to the increase of circuit area. As the True Random Number Generators (TRNGs) have the feature of low throughput, high cost, and dependency to physical devices, we choose the Pseudo-Random Number Generators (PRNGs) instead of TRNGs. Generally speaking, each fresh randomness needs 3 LUTs to build in FPGAs: 2 LUTs for Shift-Register and 1 LUT for the nonlinear feedback function. In this scheme, $6 \times 4 \times 4 = 96$ bits of fresh randomness are required for a single round operation, which means 288 LUTs is used to generate the fresh randomness.

Table 2 Performance comparison of different second-order masking designs

Design	Performance			
	Protection Orders	Fresh Randomness (Bits)	Clock Cycle	Circuit Area (kGE)
[15]	1st	72	192	25
[28]	1st	8	192	2
[19]	1st	2056	192	16
[23]	2nd	0	72	231.5
[14]	2nd	108	198	22
[27]	2nd	116	266	4.3
New	2nd	96	256	11

5.2.2 Clock Cycles

The clock cycles indicate the throughput of the encryption scheme. The use of sequential logic circuits and registers results in the increase in the number of clock cycles. However, the introduction of clock to the scheme is necessary. On the one hand, register stages avoid the propagation of glitches. On the other hand, the use of registers reduces the required time in critical path, thus a higher clock frequency chip can be applied to increase the cipher throughput.

During the operation of the tower field implementation, four register layers are inserted to store the current intermediate values. Therefore, the second-order masking SM4 has a latency of 4 clock cycles for each round. Moreover, for the SM4 encryption with 32 rounds, taking the key schedule into consideration, 256 clock cycles are consumed for one encryption operation.

5.2.3 Circuit Area

The circuit area is a common metric of hardware cost. A larger circuit requires more materials such as silicon wafers or transistors, which all contribute to the overall cost. Additionally, a larger circuit may require more complex processes and equipment for manufacturing. This design consumes 11 kGE in circuit area for the masked S-box.

It can be seen in Table 2, the three-share 2nd-order TI of SM4 achieves a balanced result in the three metrics: fresh randomness, clock cycle and circuit area. The designs are synthesized using UMC 130 standard cell library.

As a comparison to the state of the art, this design has a 48.6% reduction in circuit area compared with the S-box masking scheme SM4 used in [14]. On the other hand, the fresh randomness used outperforms reference [14] and [27]. The reduction in circuit area and fresh randomness totally leads to reduction of overhead. In addition, this scheme is designed round-based, which means a lower consumption in circuit area and a higher consumption in clock cycles, i.e., the scheme can be further optimized in latency with pipeline architecture at the cost of increasing the area.

5.3 Security Evaluation

In this section, the 2nd-order TI scheme is performed on the SAKURA-G board, and then 10 million power traces of unmasked and masked SM4 implementations are separately collected by a digal oscilloscope monitoring (Model: LeCroy WaveRunner 610Zi, sampling frequency: 250 MSamples/sec). It is obvious that there are 32 spikes in the traces of the unmasked SM4 while the traces of the masked SM4 is rather uniform, (also see Figs. 4 and 5).

5.3.1 T-test

T-test has been widely used to evaluate the side-channel vulnerability of a hardware device. It provides a general testing method of the cipher implementation and relaxes the dependency between the evaluations and the device's underlying architecture.

In this work, the first- and second-order t-tests are conducted with the method proposed by Schneider and Moradi [21]. 5 million power traces of masked SM4 cipher with fixed plaintext are collected and represented

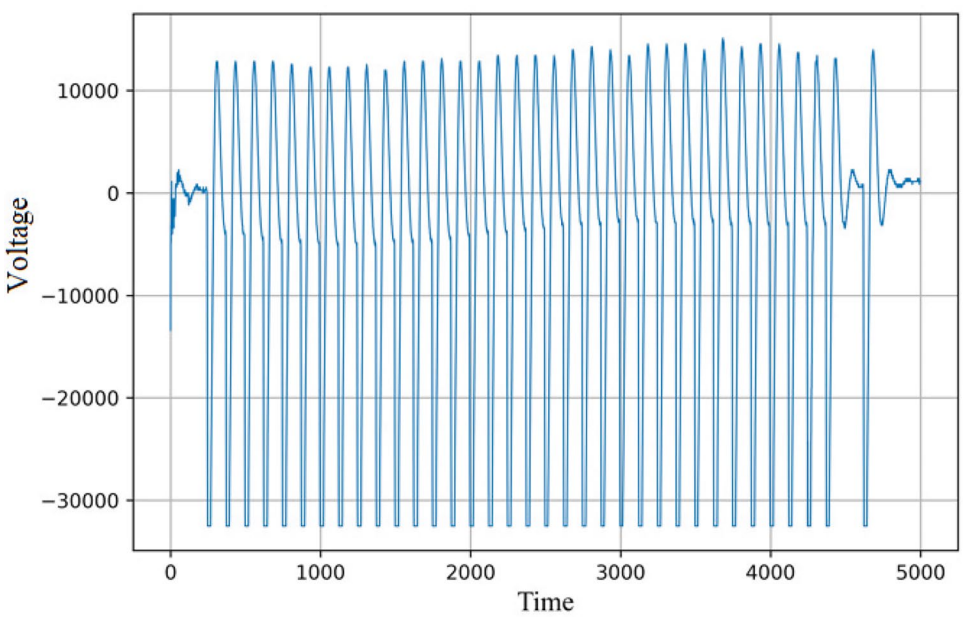

Fig. 4 The trace of SM4 cipher without any protection

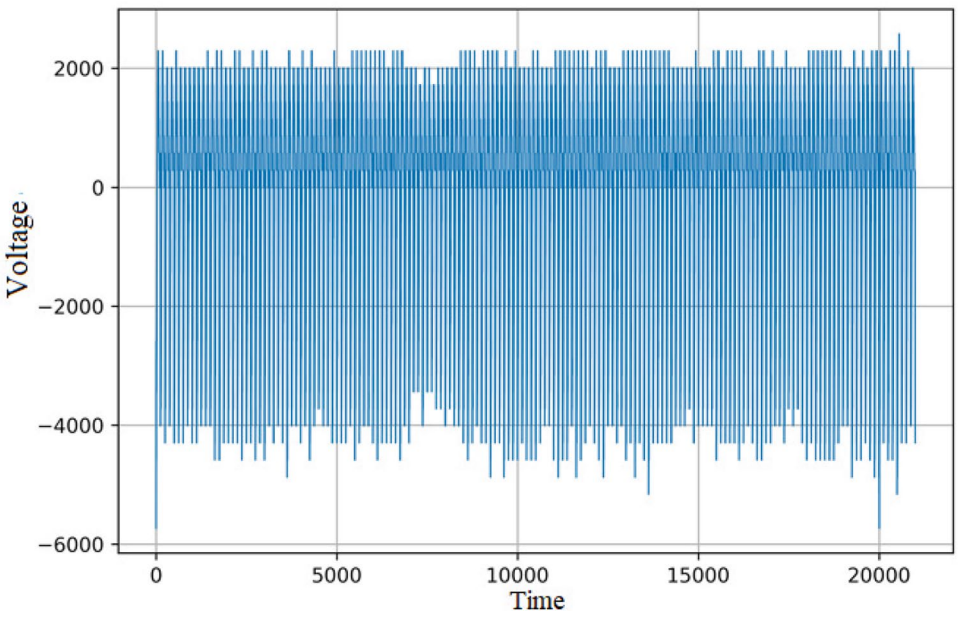

Fig. 5 The trace of SM4 cipher with 2nd-order TI

as vector X, while 5 million power traces of masked SM4 cipher with random plaintext are collected and represented as vector X'.

The first-order t-statistics can be defined as:

$$t = \frac{M_1 - M_1'}{\sqrt{\frac{(M_2 - M_1)^2}{n} + \frac{(M_2' - M_1')^2}{n'}}} \tag{14}$$

while M_n represents the n-th moment of X:

$$M_n = E(X^n) \tag{15}$$

The second-order t-statistics can be defined as:

$$t = \frac{(M_2 - M_1^2) - (M_2' - M_1'^2)}{\sqrt{\frac{(CM_4 - CM_2^2)^2}{n} + \frac{(CM_4' - CM_2'^2)^2}{n'}}} \tag{16}$$

while CM_2 and CM_4 represents the 2- and 4-th central moment of X:

$$\begin{aligned} CM_2 &= M_2 - M_1^2 \\ CM_4 &= M_4 - 4M_3M_1 + 6M_2M_1^2 - 3M_1^4 \end{aligned} \tag{17}$$

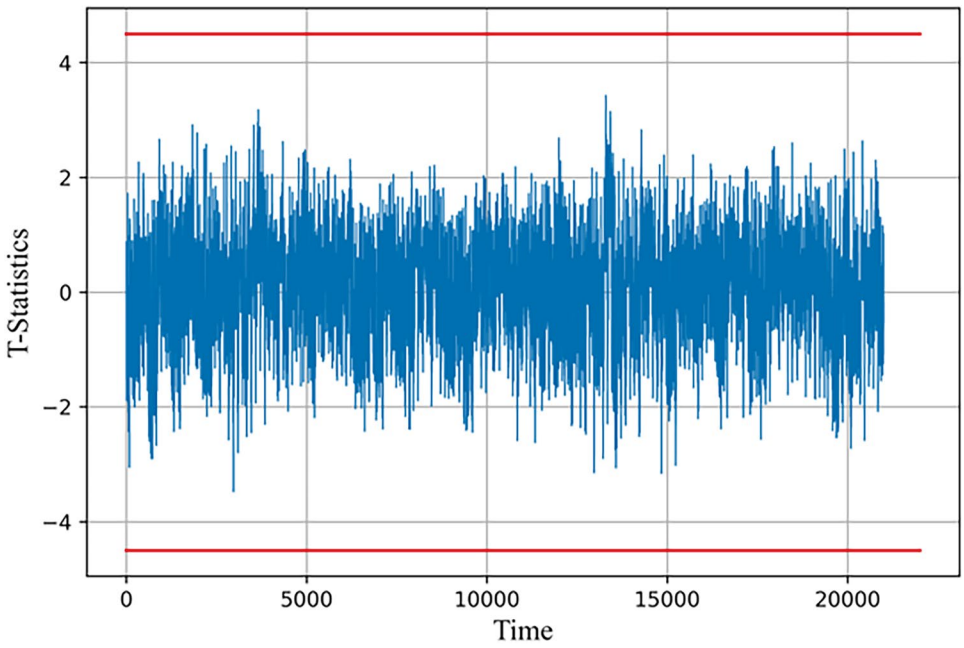

Fig. 6 First-order leakage evaluation of masked SM4 cipher

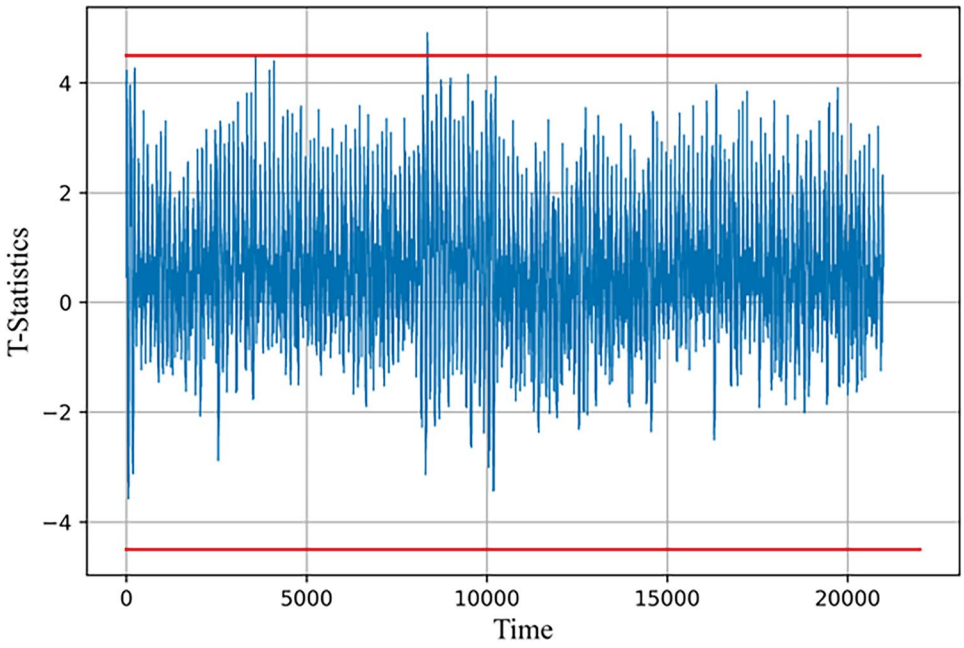

Fig. 7 Second-order leakage evaluation of masked SM4 cipher

The null hypothesis of t-test is set as X and X' are not distinguishable. In general, a threshold $|t|$ 4.5 is defined to reject the null hypothesis.

The first- and second-order t-test results are shown in Figs. 6 and 7. The fluctuation of t-statistics is within 4.5, which means the power traces between fixed plaintext and random plaintext are not distinguishable, i.e., this scheme serves as a good confusion. The t-test results confirmed the second-order security of this new design.

5.3.2 CPA

CPA is a threating SCA approach where the correlation coefficient is used to measure the possibility of a guessed key. To perform the CPA attack with Hamming distant leakage model, the trace of the first 4 rounds of the SM4 S-box is collected. The CPA attack results of the first round are shown in Fig. 8.

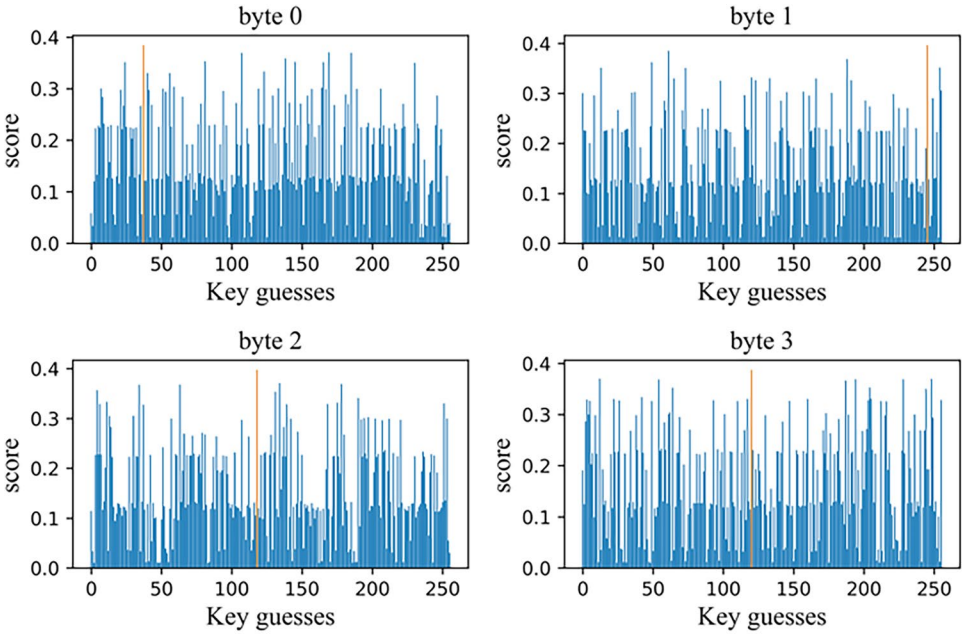

Fig. 8 CPA attack against the 1st round of encryption

By Fig. 8, we found the correlation coefficients of all the guessed key bytes are kept in a small region (less than 0.4), which also means that the correct key cannot be recovered well. More precisely, the value 0x25 (marked in orange) has a higher correlation coefficient than the others for byte 0, but it is not the true key bit value (0xF1 for the first round) of byte 0. Thus, this design provides a high resistance against 2nd-order SCA.

6 Conclusion

In this article, 2nd-order TI masking schemes for SM4 block cipher are presented. The schemes for linear and nonlinear components are designed separately using 3 shares. These are optimized for randomness as 96 bits of fresh randomness is required in a single round operation. The area overhead of this design is 16 kGE, including 11 kGE used in the masked S-box. Compared with the state-of-the-art 2nd order TI S-box of SM4 proposed by Li et al. [14], the circuit area of S-box in this design is reduced by 48.6%. Moreover, both the 2nd-order t-test and CPA experiments are performed on this implementation with 10 million traces collected, which show an outstanding resistance against 2nd-order SCA of the design.

Funding This work is supported by the Innovation Research Team Project of Guangxi Natural Science Foundation (2019GXNSFGA245004), the National Natural Science Foundation of China (62062026 and 62162016) and the Guangxi Key Research and Development Program (Guike AB23026131).

Data Availability The datasets generated during and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Akkar ML, Giraud C (2001) An implementation of DES and AES, secure against some attacks. In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems, pp. 309–318. Springer
2. Bai XF, Guo L, Li T (2008) Differential power analysis attack on SMS4 block cipher. In: Proc. 2008 4th IEEE International Conference on Circuits and Systems for Communications, pp. 613–617. IEEE
3. Bilgin B, Gierlichs B, Nikova S, Nikov V, Rijmen V (2014) Higher-order threshold implementations. In: Proc. International Conference on the Theory and Application of Cryptology and Information Security, pp. 326–343. Springer
4. Brier E, Clavier C, Olivier F (2004) Correlation power analysis with a leakage model. In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems, pp. 16–29. Springer
5. Canright D (2005) A very compact S-box for AES. In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems, pp. 441–455. Springer
6. Daemen J (2017) Changing of the guards: a simple and efficient method for achieving uniformity in threshold sharing. In: Proc. International Conference on Cryptographic Hardware and Embedded Systems, pp. 137–153. Springer
7. Dhooghe S, Nikova S, Rijmen V (2021) First-order hardware sharings of the AES. IACR Cryptol ePrint Arch 734
8. Geng H, Wu J, Liu JM, Choi M, Shi YY (2012) Utilizing random noise in cryptography: where is the tofu? In: Proc. 2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 163–167. IEEE
9. Gross H, Mangard S, Korak T (2016) Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In: Proc. 2016 ACM Workshop on Theory of Implementation Security, p. 3. Association for Computing Machinery, New York, NY, USA
10. He W, Jap D (2015) Dual-rail active protection system against side-channel analysis in FPGAs. In: Proc. 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP), pp. 64–65. IEEE
11. Ishai Y, Sahai A, Wagner D (2003) Private circuits: Securing hardware against probing attacks. In: Proc. Annual International Cryptology Conference, pp. 463–481. Springer
12. ISO/IEC: 18033-3 (2021) Information technology-Security techniques-Encryption algorithms-Part 3: Block ciphers-Amendment 1: SM4. Available at https://www.iso.org/standard/81564.html
13. Kocher P, Jaffe J, Jun B (1999) Differential power analysis. In: Proc. Annual International Cryptology Conference, pp. 388–397. Springer
14. Li XC, Zhong WD, Zhang SW, Yang XY (2018) A threshold implementation scheme for the SM4 S-box. J Cryptologic Res 6(5):641–650
15. Liang H, Wu LJ, Zhang XM, Wang JB (2014) Design of a masked S-box for SM4 based on composite field. In: Proc. 2014 Tenth International Conference on Computational Intelligence and Security, pp. 387–391. IEEE
16. Liu F, Ji W, Hu L, Ding JT, Lv SW, Pyshkin A, Weinmann RP (2007) Analysis of the SMS4 block cipher. In: Proc. Australasian Conference on Information Security and Privacy, pp. 158–170. Springer
17. Nikova S, Rijmen V, Schläffer M (2011) Secure hardware implementation of nonlinear functions in the presence of glitches. J Cryptol 24(2):292–321
18. Nikova S, Rechberger C, Rijmen V (2006) Threshold implementations against side-channel attacks and glitches. In: Proc. International Conference on Information and Communications Security, pp. 529–545. Springer
19. Pei C (2016) A method of masking SM4 and analysis against DPA attacks. J Cryptologic Res 3(1):79–90
20. Reparaz O, Bilgin B, Nikova S, Gierlichs B, Verbauwhede I (2015) Consolidating masking schemes. In: Proc. Annual Cryptology Conference, pp. 764–783. Springer
21. Schneider T, Moradi A (2015) Leakage assessment methodology - a clear roadmap for side-channel evaluations. In: Proc. Cryptographic Hardware and Embedded Systems, pp. 495–513. Springer, Berlin, Heidelberg
22. Shahmirzadi, A.R., Božilov, D., Moradi, A (2021) New first-order secure AES performance records. IACR Trans Cryptogr Hardw Embed Syst 2021(2), 304–327. https://doi.org/10.46586/tches.v2021.i2.304-327
23. Shahmirzadi AR, Moradi A (2021) Second-order SCA security with almost no fresh randomness. IACR Trans Cryptogr Hardw Embed Syst 2021(3):708–755

24. Shahmirzadi AR, Moradi A (2021) Re-consolidating first-order masking schemes: Nullifying fresh randomness. IACR Trans Cryptogr Hardw Embed Syst 2021(1):305–342
25. Trichina E (2003) Combinational logic design for AES subbyte transformation on masked data. IACR Cryptol ePrint Arch 236
26. Waddle J, Wagner D (2004) Towards efficient second-order power analysis. In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems, pp. 1–15. Springer
27. Wei YZ, Yao F, Pasalic E, Wang A (2019) New second-order threshold implementation of AES. IET Inf Secur 13(2):117–124
28. Wei M, Sun SW, Wei ZH, Hu L (2021) Unbalanced sharing: a threshold implementation of SM4. Sci China Inf Sci 64(5):1–3

Tianyi Shao is a postgraduate student at the College of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include side-channel analysis and Boolean masking.

Bohua Wei is the associate director of Guangxi Wangxin Information Technology Co., Ltd. He is conducting research in the field of information security assessment.

Yu Ou is studying for his Ph.D. degree in Guilin University of Electronic Technology, Guilin, China. His research interests include side-channel analysis and deep learning.

Yongzhuang Wei received the M.S. and the Ph.D. degrees in cryptology from Xidian University, Xi'an, China, in 2004 and 2009, respectively. Since September 2014, he joined the Guangxi Key Laboratory of Cryptography and Information Security at Guilin University of Electronic Technology, where he is currently employed as a full professor. He is now a member of Chinese Association for Cryptologic Research (CACR). His current research interests include Boolean functions, stream ciphers, block ciphers, and Hash functions.

Xiaonian Wu received the M.S. degree in National University of Defense Technology, Hunan, China in 2004. He is currently an associate Professor at the College of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include side-channel analysis and distributed computing.