Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5945312, 11 pages
https://doi.org/10.1155/2018/5945312

Research Article

A New Type of Countermeasure against DPA in Multi-Sbox of Block Cipher

Shuaiwei Zhang and Weidong Zhong

Key Laboratory of Network & Information Security of People's Armed Police, Engineering University of People's Armed Police, Xi'an 710086, China

Correspondence should be addressed to Shuaiwei Zhang; zsw36277@163.com

Received 7 March 2018; Accepted 12 April 2018; Published 28 June 2018

Academic Editor: Ximeng Liu

The Internet of Things (IoT) provides the network for physical devices, like home appliances, embedded with electronics, sensors, and software, to share and exchange data. With its fast development, security of IoT has become a crucial problem. Among the methods of attack, side-channel attack has proven to be an effective tool to compromise the security of different devices with improving techniques of data processing, like DPA and CPA. Meanwhile, many countermeasures have risen accordingly as well, such as masking and noise addition. However, their common deficiency was that every single countermeasure might not be able to protect the key information completely after statistical analysis. Sensitive information will be disclosed during differential power analysis of Sbox, since it is the only nonlinear component in block cipher. Thus, how to protect Sbox effectively was the highlight of researches. Based on Sbox-reuse concept proposed by Bilgin, this paper put forward a new type of a countermeasure scheme against DPA in multi-Sbox of block cipher. We first converted the multi-Sbox into 4×4 permutations and then reused permutation with the algebraic degree of more than one so as to turn it into a special reusable Sbox and then numbered 4×4 permutation input. Finally, we made these inputs of permutations completely random by masking. Since it was necessary to make the collected power consumption curve subject to alignment process in DPA by chosen-plaintext attack, this scheme combined the concept from DPA countermeasures of masking and noise addition. After the experiment with the proposed implementation, successful prevention of the attacker from accurately aligning the power consumption curve of the target Sbox has been proven, and the level of security has been improved by adding more random noise to protect key information and decrease the accuracy of statistical analysis.

1. Introductions

The Internet of Things (IoT) has been undergoing a fast and vast development in recent decades, which improved the efficiency and accuracy of many tasks in our life and brings more economic benefit. However, it also gives rise to the issue of security [1–4] especially in electronic devices [5]. Since 1996, when Paul Kocher proposed the side-channel attack [6], which will make the IoT applications unsecured and vulnerable, many improvements of attack method have induced the researches in countermeasures. Not only the range of cryptology security has extended from the initial security simply based on mathematical theory to comprehensive security of mathematical theory together with cryptography implementation, but also a huge thwart to the security of hardware device needed to be overcome in IoT. From the beginning of the 20th century until now, research achievements in this field emerge endlessly, such as power analysis [7, 8], timing analysis [9], electromagnetic analysis [10, 11], fault injection [12, 13], more advanced template attack [14, 15], Glitch attack [16, 17], and machine learning attack [18–23], among which power analysis has become the research emphasis for its easy implementation, lower costs, and higher successful attacking rate especially in lightweight block cipher [24]. Power analysis consists of simple power analysis, differential power analysis, and high-order power analysis, which are all based on the concept of recovering key with power difference generated by logic circuit composed with CMOS when processing "0" or "1" bit. Thanks to the vigorous development of attack theory, researches looking into countermeasures theory against power attack have also been in full swing. Over the years of study on countermeasures, the theories are basically divided

into two categories. One is the countermeasure scheme based on algorithm, such as random masking, shuffling, and hiding, characterized by low costs but low security [25–27]. The other is based on circuit level technique, featuring higher security, and more implementation costs, including two major technologies: sense amplifier based logic (SABL) [28] and wave dynamic differential logic (WDDL) [29]. In 2006, Svetla proposed the secret sharing and multiparty secure computation-based threshold implementation scheme[30], a well-developed scheme that can resist high-order DPA attack and Glitch attack [31–33], which possesses higher security and lower implementation costs. Inspired by threshold implementation and based on the concept of reused Sbox of block cipher, Bilgin proposed a design with compact implementation of multi-Sbox in 2015 [34], which greatly reduced the cost in implementation of DES.

Based on the study mentioned above, our paper puts forward a new type of a countermeasure scheme against DPA attack using concept of reused Sbox in [34]. We first convert the multi-Sbox into 4×4 permutation and reuse the permutation with the algebraic degree of more than one in order to turn it into a special and reusable Sbox and then number the 4×4 permutation input. Finally, each group of 4×4 permutation enters into Sbox after random masking; the power consumption curve is randomized by scrambling the data input from Sbox to have a higher probability of invalidating DPA. The security and feasibility of this scheme are verified by DES algorithm in our experiment.

The novel contributions of this paper are as follows.

(1) In this paper, we put forward a new type of countermeasure against DPA and it is divided into two phases. The first phase is converting the multi-Sbox into 4×4 permutations and reusing the permutation with the algebraic degree of more than one to turn it into a special reusable Sbox. The next phase is generating random input, which makes input data of Sbox completely random.

(2) Compared to other DPA masking techniques, the proposed scheme uses the value of masking as a selector and controls the sequence of data input of the multi-Sbox, instead of applying XOR or modular multiplication onto value of masking and original data. This not only results in reduced number of masking, but also increases the difficulty of aligning each power consumption curve for the attacker, which indirectly increases the noise for resisting DPA attacks.

(3) The proposed scheme can be applied to many other cryptographic algorithms based on multi-Sbox; the only difference is that, in the first phase of converting Sbox, different principles of generating permutations from Sbox that correspond to different algorithms should be considered in order to have a special and reusable Sbox and then proceed with the phase of generating random input.

This paper is organized as follows. Section 2 includes preliminaries of DPA procedures, physical basis of power attack, and concept of compact implementation. Section 3 introduces our countermeasure scheme. In Section 4, the results of the experiments are presented for validation of our scheme. Section 5 shows the security analysis of our countermeasure scheme. Section 6 is dedicated to conclusions.

2. Preliminaries

2.1. Differential Power Analysis. Differential power analysis (DPA) [7] is a side-channel attack scheme in DES algorithm put forward by Paul Kocher in 1999, whose model is based on hamming weight. The author believes that register requires different power when storing "0" and "1", which leads to the disclosure of power information. Compared with simple power analysis, differential power analysis recovers keys with statistical differential technology instead of requiring algorithm details. However, it has to collect much more consumption curves. This paper offers a conclusion of the typical process of DES algorithm differential power analysis as follows.

(1) Choose m sets of plaintexts $M_1, M_2, M_3, ..., M_m$ and encrypt each of them with the same key K to measure each set of consumption curve and mark it as $T_i[j]$; among which, i refers to the sets of plaintexts measured ($1 \le i \le m$) and j means the sampling sites.

(2) A distinguisher $D(M_i, b, K_s)$ is chosen to represent b of the median at the end of the first group of Sbox, among which M represents plaintext and $0 \le K_s \le 2^6$ stands for 6-bit key entering into the Sbox corresponding to bit b.

(3) According to the predicted K_s and the speculated value of distinguisher $D(M_i, b, K_s)$, all the consumption curves with the distinguisher value of 0 and 1 are averaged to record differential power curve, as revealed in

$$\Delta D[j] = \frac{\sum_{i=1}^{m} D(M_i, b, K_s) T_i[j]}{\sum_{i=1}^{m} D(M_i, b, K_s)} - \frac{\sum_{i=1}^{m} (1 - D(M_i, b, K_s)) T_i[j]}{\sum_{i=1}^{m} (1 - D(M_i, b, K_s))} \quad (1)$$

(4) During the observation of the current differential power curve, if an obvious large peak appears, the speculation about 6-bit key is considered as correct; if there is no remarkable peak, such speculation is incorrect and should continue.

(5) The 6-bit key that corresponds to other Sbox is predicted with the same scheme; the last 8 checking bits are obtained by brute force.

2.2. Physical Basis of Power Attack. Due to the improved manufacturing process, logic gates made by CMOS process possess lower power consumption, less costs, and stronger antijamming capability compared to TTL circuit. Almost all the mainstream cipher chips and equipment adopt devices of CMOS process to construct circuit. For the convenience of analysis, the following part offers an introduction to the physical property of CMOS device regarding its power consumption. Take inverter as an example with its internal structure shown in Figure 1.

As shown in Figure 1, this structure consists of two enhanced MOSFET, namely, N channel structure and P channel structure. When the low logic level is input, P channel conducts and N channel is cut off with high logic level output; when the high logic level is input, N channel conducts and P channel is cut off with low logic level output. The total power

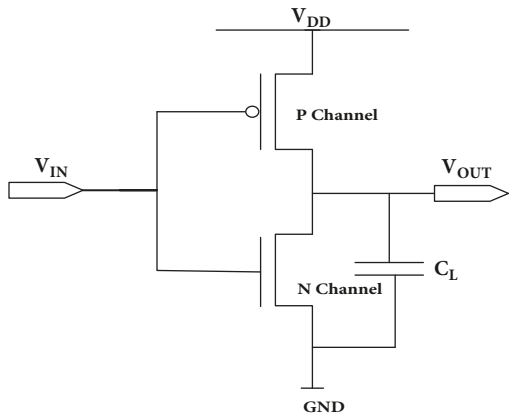

Figure 1: The internal structure of inverter.

consumption refers to the sum of static power and dynamic power which is

$$P_{total} = P_{stat} + P_{dyn} \quad (2)$$

When input V_{IN} of inverter stabilizes, the output V_{OUT} is also stable; under such circumstances, there are the conduction and the cut-off between P channel and N channel. It is found in actual measurement that a small amount of leak current I_{leak} is conveyed through the cut-off channel. Therefore, P_{stat} static power can be calculated according to the following:

$$P_{stat} = {I_{leak}}^2 V_{DD} \quad (3)$$

When the input V_{IN} of inverter changed, the output V_{OUT} changed accordingly. At this time, the dynamic power generated usually consists of two parts: one is P_{chrg}, power consumption of load capacitor C_L, while charge and discharge account for 85%; the other is P_{sc}, power consumption of top-down short-circuit current generated by the two concurrently conducting channels within very short period of time when the input level reaches $V_{DD}/2$ (accounts for 15%). Table 1 represents the constitution of the total power consumption of inverter with different inputs. Other logic gates based on CMOS process also have the above-mentioned consumption properties with much more complicated structure. Multielectrode MOS hopping superposition has made the generated dynamic power more obvious. Therefore, attackers can easily align the power consumption with the key, which serves as the principle of power attack after the hardware implementation of cryptographic algorithm.

2.3. Compact Implementation

2.3.1. Introduction. Sbox compact implementation is proposed by Bilgin based on threshold implementation in 2015 [34]. In threshold implementation, Sbox with algebraic degree of two will be implemented with at least three shares while Sbox is with algebraic degree of three with at least four shares. The circuit scale grows exponentially with the increasing number of shares. Therefore, researchers hope to replace the Sbox of higher algebraic degree with several serial Sbox of lower algebraic degree so as to ensure less resource consumption and less reduction of speed thanks to the employment of pipeline technology. Bilgin adopted the affine-equivalence technology to seek the public high-degree permutation of the eight Sbox in DES algorithm for reuse and then implemented the residual parts with algebraic degree of 1, thus reducing the hardware resources of Sbox by 50% [34].

2.3.2. Scheme Implementation. This scheme is dedicated to the 4×4 Sbox. As it can be seen as the permutations are of 4 bits, some of its properties deserve further study.

One permutation of n bits constitutes a symmetric group. An affine equivalence is defined as follows.

Definition 1. If there is a pair of affine permutation $A(x)$ and $B(x)$ which also meets $S_1 = B \circ S_2 \circ A$, $S_1(x)$ and $S_2(x)$ can be called affine equivalence.

The permutations that form affine equivalence in n bits permutations constitute a class. In this class, a permutation can be regarded as the representation element. The permutations in one class have the same algebra degree. At the same time, all the permutations are represented with $\mathscr{A}_{2^n}$ or $\mathscr{S}_{2^n} \backslash \mathscr{A}_{2^n}$.

Literature reveals that in 4-bit permutations, there are one affine class, six quadratic classes, and 295 cubic classes, among which all the affine class and quadratic classes all belong to $\mathscr{A}_{16}$; however, 144 out of 295 cubic classes belong to $\mathscr{A}_{16}$ and the remaining 151 are categorized into $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$.

$\mathscr{M} = \{\mathscr{Q}_{004}, \mathscr{Q}_{012}, \mathscr{Q}_{293}, \mathscr{Q}_{294}, \mathscr{Q}_{299}, \mathscr{Q}_{300}\}$ is a set for 6 quadratic classes. It is proven in [19] that, in $\mathscr{A}_{16}$, permutations with any algebra degree can be represented by the elements from $\mathscr{M}$. The cubic class permutation in $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$ can be represented by one or many secondary permutations in $\mathscr{A}_{16}$ and one-third of permutations in $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$; however, the third permutation in $\mathscr{N} = \{\mathscr{Q}_{001}, \mathscr{Q}_{003}, \mathscr{Q}_{013}, \mathscr{Q}_{301}\}$ is often chosen to represent all because they possess some fine properties. Therefore, we aim to decompose different Sbox such that minimum number of nonlinear permutations is used to jointly describe all Sbox. Refer to [34] for more specific implementation of scheme.

3. Our Countermeasure Scheme

3.1. Classification of DPA Countermeasures Methods. DPA can speculate the key by subjecting the collected consumption curve to statistical difference. Therefore, the protection of any of the links can reduce the possibility of successful attack. Currently, the countermeasure methods for DPA usually fall into the following three categories.

(1) Countermeasures for the Leaked Information. In light of the low power consumption and fast speed, the mainstream hardware platforms all use chips based on CMOS process. It is defined by the working principle of CMOS gates that different power consumption will be generated when processing bit "0" and "1". Therefore, the countermeasures targeted the nature of disclosed information which is changing the processed "0" and "1" bit through certain technologies, such as adding mask.

Table 1: Constitution of the total power consumption of inverter with different inputs.

Initial State	Final State	Constitution of Total Power Consumption
0	0	P_{stat}
1	1	P_{stat}
0	1	$P_{stat} + P_{chrg} + P_{sc}$
1	0	$P_{stat} + P_{chrg} + P_{sc}$

(2) Countermeasures for the Implementation of Circuit Environment. As DPA is a method based on chosen-plaintext attack, it has high requirements for precision of measured consumption curve. If Signal to Noise Ratio (SNR) reduces, it will give rise to the high number of power consumption curves in attack and even result in the failure of attack. Therefore, the countermeasures for the implementation of circuit environment are to artificially introduce noise to the circuit in order to enhance attack difficulty and reduce the probability of successful attacks.

(3) Countermeasures for the Data Postprocessing. Data of the collected power consumption curve need to be aligned during the data postprocessing of DPA. The alignment is carried out by keeping the leaking points, which leak the sensitive information from different power consumption curves, aligning at the same point of time, to recover the key with a higher efficiency. The countermeasure of scrambling is employed to increase the difficulty of aligning different power consumption curves, in order to protect the circuit from leaking sensitive information.

This scheme is a combined countermeasure that includes countermeasures for the leaked information, the implementation of circuit environment, and data postprocessing. By utilizing the Sbox-reuse technology and randomly inputting data with masking, it can resist DPA because of raising random noise and preventing attackers from aligning the consumption curves corresponding with the key data with high probability in the data postprocessing.

3.2. Scheme Flow. In accordance with Nikova's theory, when the bit digit input $n \geq 4$, such permutation is secure. It is also noted that, in the existing cryptography scheme, the smallest Sbox is 4×4; under such circumstances, the minimum permutation of 4×4 in the Sbox framework turns out to be logical. The specific scheme flow is listed as follows.

(1) n independent parallel Sboxes are replaced by a special and reusable Sbox framework S', using the compact algorithm. The 4×4 Sbox in S' is numbered

$$[S_0(m_0), S_1(m_1) \cdots S_{n-1}(m_{n-1})] \Longrightarrow S', \tag{4}$$

in which m_{n-1} stands for the input of the $(n\text{-}1)^{\text{th}}$ 4-bit Sbox permutation, $S_{n-1}(m_{n-1})$ is the output of the $(n\text{-}1)^{\text{th}}$ 4-bit Sbox permutation, and S' is a special and reusable Sbox framework.

(2) A random number R_1 appears before the Sbox algorithm of circuit

$$R_1 = \left(r_1, r_2, \cdots r_{g(n)}\right) \tag{5}$$

Among which, $0 \leq R_1 \leq n-1$ and $g(n)$ stands for the binary bit digit that corresponds to n, the number of 4×4 Sbox participating in algorithm.

(3) The first 4×4 Sbox permutation entering S' is chosen based on R_1 value; the permutation is S_{R_1}.

(4) The random number R_1 and the input of 4×4 Sbox permutation entering S' are subjected to XOR operation with the input data as the random number of the next 4×4 Sbox permutation

$$m_{R_1} \oplus R_1 = R_2 \tag{6}$$

(5) Repeat Step (3) and Step (4); if the 4×4 Sbox that corresponds to the newly generated random number R_i has been chosen, then execute Step (6).

(6) R_i is subjected to XOR operation bit by bit, $R_i{}^*$ is obtained. Namely,

$$R_i{}^* = r_{g(n)\bullet(i-1)+1} \oplus r_{g(n)\bullet(i-1)+2} \oplus \cdots \oplus r_{g(n)\bullet(i-1)+g(n)} \tag{7}$$

(7) Choose a distinguisher $f(R_i{}^*)$.

$$f\left(R_i{}^*\right) = \begin{cases} S_{(R_i-1+n) \bmod n} & \text{if } R_i{}^* = 0 \\ S_{(R_i+1) \bmod n} & \text{if } R_i{}^* = 1 \end{cases} \tag{8}$$

If $R_i{}^*$, the result of bit-by-bit XOR operation of R_i is "0", the permutation $S_{(R_i-1+n) \bmod n}$ is chosen; if the result is "1", the permutation $S_{(R_i+1) \bmod n}$ is chosen. If the result is the selected 4×4 Sbox permutation, execute Step (7) until the 4×4 Sbox that has never been chosen appears and returns to Step (3).

(8) Repeat the above-mentioned steps until all n 4×4 Sbox permutations have all been chosen and entered the S'; Figure 2 is the flow of our scheme.

4. Experiments

This part mainly introduces the scheme implementation by using DES algorithm Sbox. Although it is known that DES algorithm of 56-bit key has been proven insecure in many applications, Triple-DES has been proven secure for its 112-bit key and widely applied to many electronic devices [35].

4.1. Implementation Steps of DES Algorithm Sbox Scheme. According to DES algorithm, its Sbox consists of *eight* parallel 6×4 Sboxes; in each Sbox, the first and sixth of its 6-bit input are used to determine four 4×4 permutaions. The 4-bit input consists of the second, third, fourth, and fifth of the 6-bit input; therefore, the *eight* 6×4 Sboxes actually consist of *thirty-two* 4×4 multi-Sbox. The DES algorithm Sbox is

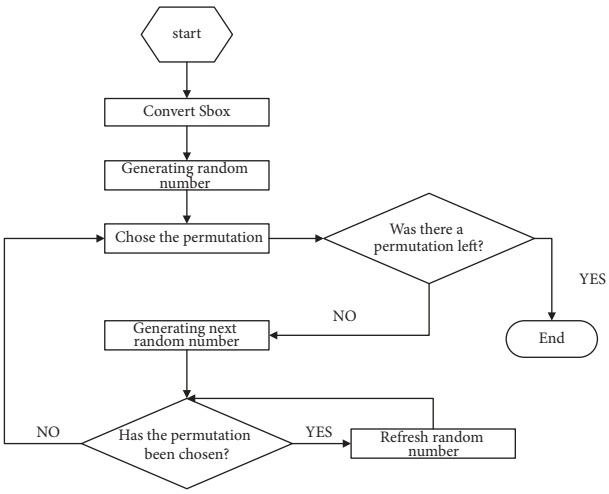

Figure 2: The flow of our scheme.

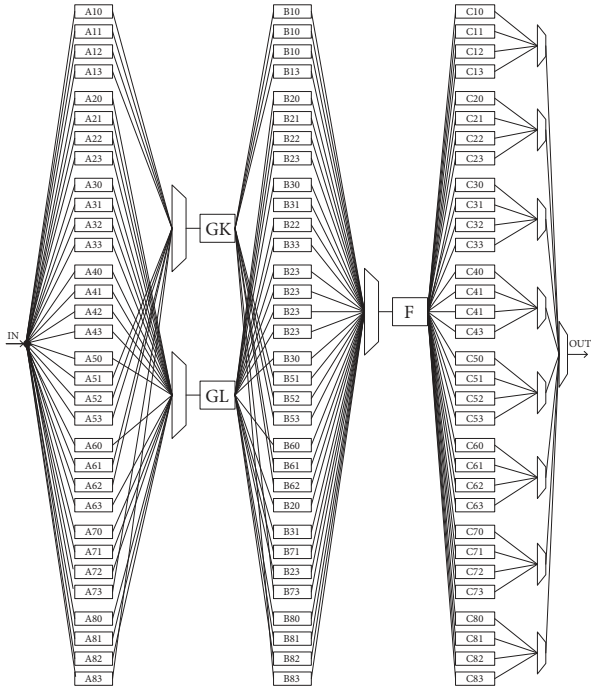

Figure 3: Special and reusable Sbox framework S'.

implemented according to the flows introduced in 3.2 with specific steps listed as follows.

(1) The *eight* 6×4 Sboxes in DES algorithm are converted into *thirty-two* 4×4 permutations. As suggested by Bilgin's reuse concept, n independent parallel Sboxes are converted into a special and reusable Sbox framework S'.

$$[S_0(m_0), S_1(m_1) \cdots S_7(m_7)] \Longrightarrow S' \tag{9}$$

The logic diagram after conversion is listed in Figure 3:

GK, GL, F, *Aij*, *Bij*, and *Cij* are known permutations. Refer to [34] for the specific permutations.

(2) As there are 8 Sboxes of 4×4 participating in DES algorithm, therefore, $n = 8$ and $g(n) = 3$. To satisfy the following algorithm requirements, we make $g(n)' = g(n) + 1 = 4$. $R_1 = (r_1, r_2, \cdots r_{g(n)'}) = (r_1, r_2, r_3, r_4)$ is the random number generated, $0 \le R_1 \le 15$.

(3) Suppose ${R_1}' = (r_2, r_3, r_4)$; the first 4×4 Sbox permutation entering S' is chosen based on the value of ${R_1}'$.

(4) The random number R_1 and the input of 4×4 Sbox permutation entering S' are subjected to XOR operation; the results obtained serve as the random number for the selection of the next 4×4 Sbox permutation.

$$m_{R_1'} \oplus R_1 = R_2 \tag{10}$$

(5) Repeat Step (3) and Step (4); if the 4×4 Sbox that corresponds to the newly generated random number R_i has been chosen, then execute Step (6).

(6) R_i is subjected to XOR operation bit by bit to obtain ${R_i}^*$.

$${R_i}^* = r_{3(i-1)+1} \oplus r_{3(i-1)+2} \oplus r_{3(i-1)+3} \oplus r_{3(i-1)+4} \tag{11}$$

(7) Choose a distinguisher function $f({R_i}^*)$

$$f({R_i}^*) = \begin{cases} S_{({R_i}'+7) \bmod 8} & \text{if } {R_i}^* = 0 \\ S_{({R_i}'+1) \bmod 8} & \text{if } {R_i}^* = 1 \end{cases} \tag{12}$$

If ${R_i}^*$, the result of bit-by-bit XOR operation of R_i is "0"; the permutation $S_{({R_i}'+7) \bmod 8}$ is chosen; if the result is "1", the permutation $S_{({R_i}'+1) \bmod 8}$ is chosen. If the result is the selected 4×4 Sbox permutation, execute Step (7) until the 4×4 Sbox that has never been chosen appears and returns to Step (3).

(8) Repeat the above-mentioned steps until all *eight* 4×4 Sboxes permutations have all been chosen and entered the S'. Finally, output all the parts of S simultaneously. The pseudocode of scheme is listed as Algorithm 1 where $S_{R_i'} = in$ means 4×4 Sbox $S_{R_i'}$ has never been chosen.

4.2. Experimental Results. The experiment environment of this scheme is presented in Table 2.

In accordance with 3.2, this scheme is subjected to experiment with the results listed as follows.

4.2.1. Resource and Operating Speed Result. On one hand, Tables 3 and 4 are the resources consumed by the algorithm in the FPGA platform between the scheme proposed in this paper and original scheme. It can be seen that the total logic elements of this scheme are 33k, which is roughly eightfold the original scheme. But considering the whole resources in FPGA chip (about 114480 logic elements), our scheme is still practical to operate.

On the other hand, the speed of our countermeasure implementation is up to *80M* and an average number of periods of 41 are needed to process one group of plaintext.

4.2.2. Security Result. Figures 4 and 5 are the DPA result comparison between original DES algorithm and our countermeasure scheme for each Sbox within right key (both are using fourth-order cumulate to make result more obviously). Apparently, after 800 power traces of DPA, we found that

Input: R_1, muti-Sbox
Output: S
(1) ***function***$(R_1$,*muti-Sbox, S)*
(2) *Convert muti-Sbox to* S'
(3) *Number the* 4×4 *Sbox start at* S_0
(4) ***input*** *Random masking* $R_1(r_1, r_2, r_3, r_4)$
(5) ***for*** $i = 1$ *to* 8
(6) ***do*** $R_i' \longleftarrow (r_{3i-1}, r_{3i}, r_{3i+1})$
(7) *Chose* $R_i th$ 4×4 *Sbox* $S_{R_i'}$
(8) ***if*** $(S_{R_i'} = in)$
(9) *Save* $S_{R_i'}$
(10) $R_i \longleftarrow (m_{R_i'} \oplus R_i)$
(11) ***go to*** *Line (7)*
(12) ***else***
(13) $R_i^* \longleftarrow (r_{3i-2}, r_{3i-1}, r_{3i}, r_{3i+1})$
(14) ***if*** $(R_i^* = 0)$
(15) *Chose* $S_{(R_i+7) \bmod 8}$
(16) ***go to*** *Line (8)*
(17) ***else if*** $(R_i^* = 1)$
(18) *Chose* $S_{(R_i+1) \bmod 8}$
(19) ***go to*** *Line (8)*
(20) ***end if***
(21) ***end if***
(22) ***end for***
(23) $S \longleftarrow [S_{R_1'} \| S_{R_2'} \| S_{R_3'} \| S_{R_4'} \| S_{R_5'} \| S_{R_6'} \| S_{R_7'} \| S_{R_8'}]$
(24) ***return*** S
(25) ***end function***

Algorithm 1: The pseudocode of scheme.

Table 2: Experimental environment.

Tools	Pattern
PC	Lenovo Thinkpad x240 core i7
System	Windows7
Software	Quatus prime 15.1, Modelsim15.1
FPGA	Altera EP4CE115F2317
Oscilloscope	Tektronix MSO5204B
Differential probe	Tektronix TDP3500
Regulated power supply	DH-1719

Table 3: Total logic elements of original scheme.

Number	Parameters	Values
1	Total logic elements	4137
2	Total combinational function	3856
3	Dedicated logic registers	1144
3	Total registers	1144
4	Total pins	194

Table 4: Total logic elements of this scheme.

Number	Parameters	Values
1	Total logic elements	33602
2	Total combinational function	30997
3	Dedicated logic registers	7385
3	Total registers	7385
4	Total pins	187

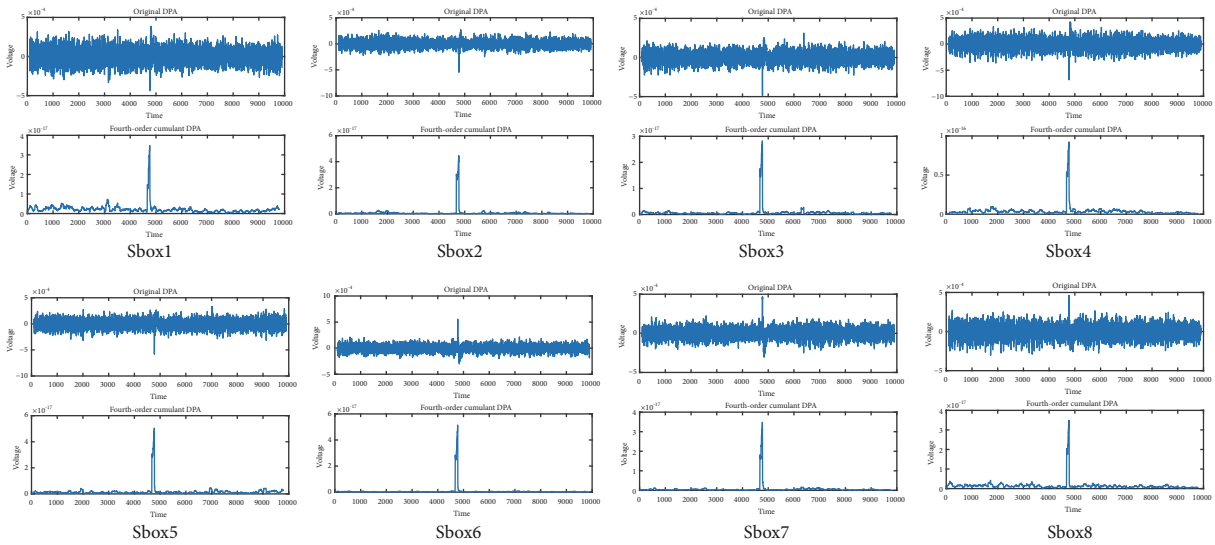

Figure 4: DPA using original scheme with 800 traces.

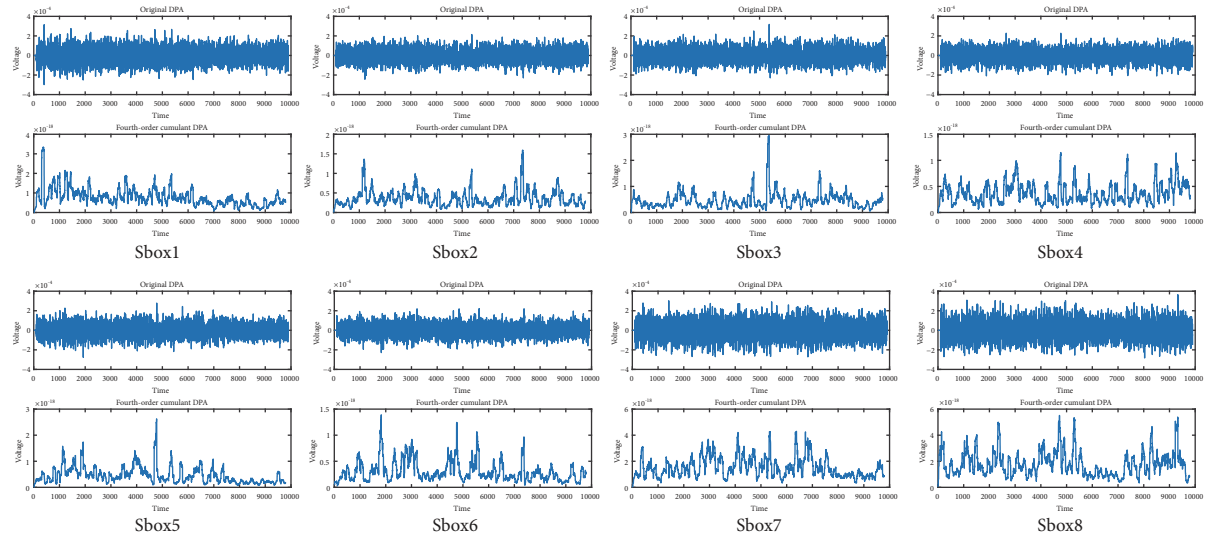

Figure 5: DPA using our scheme with 5000 traces.

there was one obvious peak in original DPA of DES algorithm for each Sbox. On the contrary, several peaks in our scheme with 5000 traces we found in Figure 5 were "ghost" peaks, which leads to wrong key corresponding to the target Sbox. Therefore, we conclude that our countermeasure scheme in Sbox of DES can improve the security of implementation against DPA.

5. Security Analysis

5.1. Theory of DPA Power Analysis. The DPA power attack is target at the output of register corresponding to the Sbox in cryptographic algorithms circuit. Although sensitive information might leak from the logic circuits inside the Sbox and be used by attackers for Glitch attack, we mainly focus on DPA, and our scheme is offering protection to registers.

Take 4 × 4 Sbox as an example with the specific circuit diagram shown in Figure 6, in which power region is at where attackers want to collect power consumption.

X_i represents the input of Sbox, Y_i stands for output of Sbox as well as the input of register, and Q_i is the output of register.

The internal structure of one register is shown as Figure 7.

One register consists of a few control components and one D trigger; the D trigger is composed of 6 NAND gates shown in Figure 8.

Therefore, in line with the analysis of 2.2, when an obvious large hopping takes place after D is input, CMOS transistors within *eight* NAND gates, *one* OR gate, and *one* NOT gate will instantaneously generate dynamic power consumption. Attackers can attack the device according to the power consumption collected and by means of DPA.

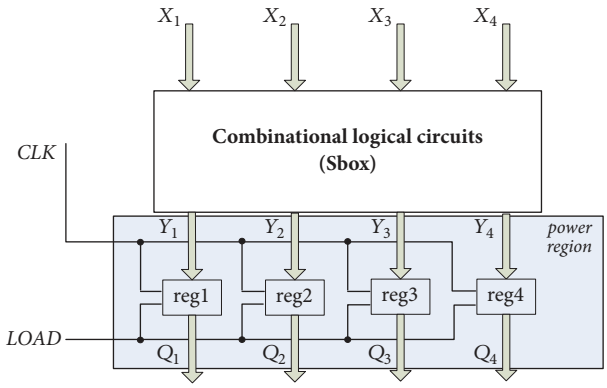

Figure 6: The corresponding register of Sbox.

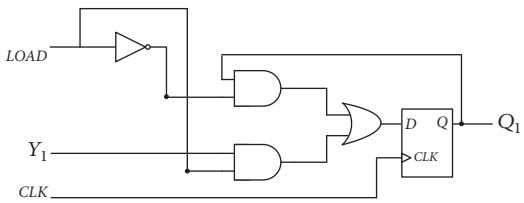

Figure 7: Internal structure of one register.

5.2. Analysis of the Security of Traditional Power Model. It is shown in 2.2 that, in cryptographic calculation circuit, the total power consumption is the sum of dynamic power and static power:

$$P_{total} = P_{stat} + P_{dyn} \tag{13}$$

Due to the output of register, different hopping corresponds to different power consumption and is represented by $P_{0\longrightarrow1}$, $P_{1\longrightarrow0}$, $P_{0\longrightarrow0}$, and $P_{1\longrightarrow1}$; and, obviously, $P_{0\longrightarrow0} = P_{1\longrightarrow1} = P_{stat}$. Therefore, as shown by 5.1,

$$P_{0\longrightarrow1} = a\left(P_{AND} + P_{OR} + P_{NOT}\right) + n + P_{stat} \tag{14}$$

$$P_{1\longrightarrow0} = a\left(P_{AND}{}' + P_{OR}{}' + P_{NOT}{}'\right) + n + P_{stat}, \tag{15}$$

in which a is a constant coefficient, P_{AND}, P_{OR}, and P_{NOT} are dynamic power consumption in logic gates, and n is noise. As abundant facts have proven that $P_{0\longrightarrow1} > P_{1\longrightarrow0}$, it is believed that

$$P_{0\longrightarrow1} = P_{1\longrightarrow0} + \varepsilon \tag{16}$$

As hamming weight model is adopted in DPA, therefore,

$$P_0 = \frac{\left(P_{1\longrightarrow0} + P_{0\longrightarrow0}\right)}{2} \tag{17}$$

$$P_1 = \frac{\left(P_{0\longrightarrow1} + P_{1\longrightarrow1}\right)}{2} \tag{18}$$

The following part offers an analysis of the DPA security. If attackers succeed in guessing the key, refer to Table 5.

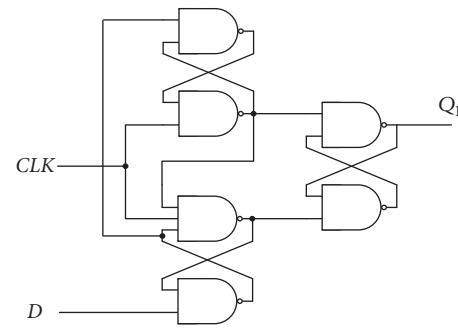

Figure 8: Internal structure of D trigger.

Table 5: Situation when attackers succeed in guessing the key.

	Guess value	True value	power
Possibility1	0	0	P_0
Possibility2	1	1	P_1

In accordance with DPA principle, power consumption with the guessed value of 1 minus the power consumption with the guessed value of 0 is represented as follows:

$$DP = P_1 - P_0 = \frac{\left(P_{0\longrightarrow1} + P_{1\longrightarrow0}\right)}{2} = \frac{\varepsilon}{2} \tag{19}$$

If attackers fail to guess the key, refer to Table 6.

Power consumption with the guessed value of 1 minus the power consumption with the guessed value of 0 is represented as follows:

$$DP = \left(P_0 + P_1\right) - \left(P_0 + P_1\right) = 0 \tag{20}$$

Therefore, the possibility of guessing the key correctly for the attackers is 1/16.

5.3. Analysis of the Security in Our Scheme. The proposed scheme combines the methods of conversion of Sbox and randomizes the input to resist DPA. Table 7 lists the situation of guessing key in our scheme.

As it is shown in the table, the attackers can only locate the position of leaking point on the power consumption curve of target Sbox, when the sequence of speculating Sbox and the key to the corresponding Sbox are both correct. In other cases, the positions of leaking points are random. Compared to conventional masking schemes, there are 3 advantages.

(1) Multi-Sboxes will rely on each other, due to existence of the selector for value of masking.

Keys of conventional cryptographic algorithms can be successfully recovered by DPA because their multi-Sboxes are parallel independently; DPA is able to successfully recover key from each single Sbox to get the corresponding key. However, the proposed scheme utilizes a special reusable Sbox, having random sequence of encrypting data in Sboxes each time, resulting in different success rate of recovering key from different Sboxes, shown in Table 8. Also depicted in Figure 9, the success rate of recovering key from corresponding Sbox with proposed scheme is decreasing exponentially compared to conventional method.

Table 6: Situation when attackers fail to guess the key.

	Guess value	True value	power
Possibility1	0	0	P_0
Possibility2	1	1	P_1
Possibility3	0	1	P_1
Possibility4	1	0	P_0

Table 7: Situation of guessing key in our scheme.

	Guess Sbox	Guess value	True value	power
Possibility1	correct	correct	sure	sure
Possibility2	correct	wrong	random	random
Possibility3	wrong	correct	random	random
Possibility4	wrong	wrong	random	random

Table 8: The success rate of recovering key corresponding n^{th} Sbox.

	Sequence of speculating Sbox	Guessing the value of key	Success rate
$Sbox^{1st}$	1/8	1/64	$(1/2)^9$
$Sbox^{2nd}$	1/8	$(1/64)^2$	$(1/2)^{15}$
$Sbox^{3rd}$	1/8	$(1/64)^3$	$(1/2)^{21}$
$Sbox^{4th}$	1/8	$(1/64)^4$	$(1/2)^{27}$
$Sbox^{5th}$	1/8	$(1/64)^5$	$(1/2)^{33}$
$Sbox^{6th}$	1/8	$(1/64)^6$	$(1/2)^{39}$
$Sbox^{7th}$	1/8	$(1/64)^7$	$(1/2)^{45}$
$Sbox^{8th}$	1/8	$(1/64)^8$	$(1/2)^{51}$

(2) It is difficult to align power consumption curves increases during data postprocessing.

Since the principle of DPA is to align the position of leaking points for sensitive information, the statistical differential method is then applied to recover the key. However, the positions of leaking points for sensitive information on different power consumption curves are not located within one period with a high possibility for proposed scheme; additional measures need to be applied to move power consumption curves during data postprocessing for attacks.

(3) Increased noise exists for DPA attack.

Since the noise generated during DPA attack can be eliminated with statistical differential method, the noise will be on superposition randomly while processing data of each single Sbox during process of encryption for the proposed scheme, as the method for inputting data is based on Sbox in series randomly. Moreover, this noise cannot be eliminated by statistical differential method; thus, even if the attackers moved the power consumption curves precisely and successfully recovered the keys corresponding to Sboxes, the attack will still end up in failure because of the interference of the noises in the result.

6. Conclusions

This paper proposed a countermeasure scheme of multi-Sbox against DPA attack, based on the multi-Sbox-reuse concept and random input for IoT applications security. Compared to other DPA masking techniques, the proposed scheme uses the value of masking as a selector and controls the sequence of data input of the multi-Sbox, instead of applying XOR or modular multiplication onto value of masking and original data. This not only results in reduced number of masking, but also increases the difficulty of aligning each power consumption curve for the attacker, which indirectly increases the noise for resisting DPA attacks. With the experiments, our scheme is supported correctly and accurately by experimental evidence of power data for DES algorithm processing in our DPA platform as Figure 10 has shown.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant no. 2017YFB0802000), National Natural

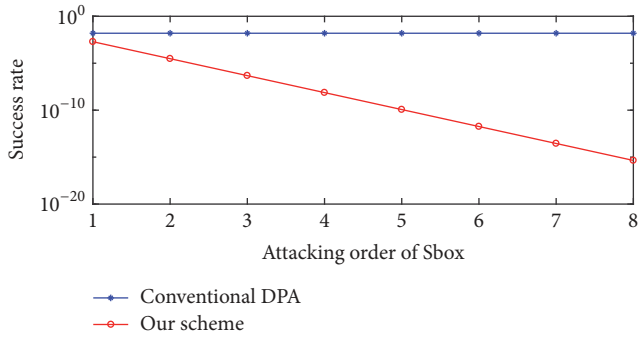

Figure 9: Comparison between conventional DPA and our scheme in success rate.

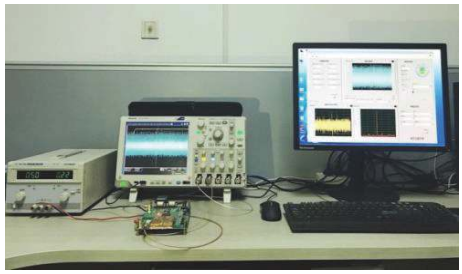

Figure 10: DPA platform.

Science Foundation of China (Grant nos. U1636114, 61772550, and 61572521), and National Cryptography Development Fund of China (Grant no. MMJJ20170112).

References

[1] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 68–75, 2011.

[2] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[4] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[5] A. A. Pammu, K.-S. Chong, W.-G. Ho, and B.-H. Gwee, "Interceptive side channel attack on AES-128 wireless communications for IoT applications," in *Proceedings of the 2016 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2016*, pp. 650–653, Republic of Korea, October 2016.

[6] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Advances in Cryptology-CRYPTO '96*, Lecture Notes in Computer Science, pp. 104–113, Springer, Berlin, Germany, 1996.

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*, pp. 388–397, Springer, Berlin, Germany, 1999.

[8] W. Wang, Y. Yu, F. Standaert, J. Liu, Z. Guo, and D. Gu, "Ridge-Based DPA: Improvement of Differential Power Analysis For Nanoscale Chips," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1301–1316, 2018.

[9] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proceedings of the third International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2001*, vol. 2162, pp. 309–318, Springer, Berlin, Germany, May 2001.

[10] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, pp. 251–261, Springer, Berlin, Germany, 2001.

[11] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration, the VLSI Journal*, vol. 40, no. 1, pp. 52–60, 2007.

[12] G. Piret and J. Quisquater, "A differential fault attack technique against spn structures, with application to the AES and khazad," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, vol. 2779 of *Lecture Notes in Computer Science*, pp. 77–88, Springer, Berlin, Germany, 2003.

[13] C. H. Kim and J.-J. Quisquater, "Faults, injection methods, and fault attacks," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 544-545, 2007.

[14] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 13–28, Springer, Berlin, Germany, 2003.

[15] L. Lerman, R. Poussier, O. Markowitch et al., "Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version," *Journal of Cryptographic Engineering*, pp. 1–13, 2017.

[16] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," in *nternational Workshop on Cryptographic Hardware and Embedded Systems*, vol. 2005, pp. 157–171, Springer, Berlin, Germany.

[17] S. Mangard, T. Popp, and B. M. Gammel, "Side-channel leakage of masked CMOS gates," in *Topics in cryptology–CT-RSA 2005*, pp. 351–365, Springer, Berlin, Germany, 2005.

[18] S. Zhang, X. Yang, W. Zhong, and Y. Wei, "An improved combinational side-channel attack on S-box in block cipher," *Journal of Internet Technology*, vol. 17, no. 1, pp. 157–166, 2016.

[19] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: A first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 293–302, 2011.

[20] E. Cagli, C. Dumas, and E. Prouff, "Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures," in *International Conference on Cryptographic Hardware and Embedded Systems*, vol. 2017, pp. 45–68, Springer International Publishing, Champa.

[21] S. Hou, Y. Zhou, H. Liu, and N. Zhu, "Wavelet support vector machine algorithm in power analysis attacks," *Radioengineering*, vol. 26, no. 3, pp. 890–902, 2017.

[22] L. Lerman, Z. Martinasek, and O. Markowitch, "Robust profiled attacks: Should the adversary trust the dataset?" *IET Information Security*, vol. 11, no. 4, pp. 188–194, 2017.

[23] W. Shan, S. Zhang, and Y. He, "Machine learning based side-channel-attack countermeasure with hamming-distance redistribution and its application on advanced encryption standard," *IEEE Electronics Letters*, vol. 53, no. 14, pp. 926–928, 2017.

[24] S. Tang, W. Li, and J. Wu, "Power analysis attacks against FPGA implementation of KLEIN," *Security and Communication Networks*, 2017.

[25] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proceedings of the third International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2001*, vol. 2162, pp. 309–318, Springer, May 2001.

[26] M. Akkar, R. Bévan, and L. Goubin, "Two Power Analysis Attacks against One-Mask Methods," in *International Workshop on Fast Software Encryption*, vol. 2004, pp. 332–347, Springer, Berlin, Germany.

[27] A. A. Ding, L. Zhang, Y. Fei, and P. Luo, "A Statistical Model for Higher Order DPA on Masked Devices," in *nternational Workshop on Cryptographic Hardware and Embedded Systems*, pp. 147–169, Springer, Berlin, Germany, 2014.

[28] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC '02)*, pp. 403–406, IEEE, September 2002.

[29] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 246–251, IEEE Computer Society, February 2004.

[30] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *nternational Conference on Information and Communications Security*, pp. 529–545, Springer, Berlin, Germany, 2006.

[31] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Trade-Offs for Threshold Implementations Illustrated on AES," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1188–1200, 2015.

[32] A. Shahverdi, M. Taha, and T. Eisenbarth, "Lightweight side channel resistance: threshold implementations of Simon," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 66, no. 4, pp. 661–671, 2017.

[33] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-order threshold implementations," in *nternational Conference on the Theory and Application of Cryptology and Information Security*, vol. 8874, pp. 326–343, Springer, Berlin, Germany, 2014.

[34] B. Bilgin, M. Knežević, V. Nikov, and S. Nikova, "Compact Implementations of Multi-Sbox Designs," in *International Conference on Smart Card Research and Advanced Applications*, pp. 273–285, Springer, 2015.

[35] Y. Ren, L. Wu, H. Li et al., "Key recovery against 3DES in CPU smart card based on improved correlation power analysis," *Tsinghua Science and Technology*, vol. 21, no. 2, pp. 210–220, 2016.