IIoTBC: A Lightweight Block Cipher for Industrial IoT Security

Juanli Kuang[1, 2, 3], Ying Guo[2, 3*], and Lang Li[2, 3*]
[1] Faculty of Innovation Engineering, Macau University of Science and Technology
Macau, 999078 China
[e-mail: kuangjuanli0917@126.com]
[2] College of Computer Science and Technology, Hengyang Normal University
Hengyang, 421002 China
[e-mail: guoying129@foxmail.com, lilang911@126.com]
[3] Hunan Provincial Key Laboratory of Intelligent Information Processing and Application
Hengyang Normal University, Hengyang, 421002 China
[*]Corresponding author: Ying Guo, Lang Li

Received August 11, 2022; revised December 9, 2022; accepted January 7, 2023; published January 31, 2023

Abstract

The number of industrial Internet of Things (IoT) users is increasing rapidly. Lightweight block ciphers have started to be used to protect the privacy of users. Hardware-oriented security design should fully consider the use of fewer hardware devices when the function is fully realized. Thus, this paper designs a lightweight block cipher IIoTBC for industrial IoT security. IIoTBC system structure is variable and flexibly adapts to nodes with different security requirements. This paper proposes a 4×4 S-box that achieves a good balance between area overhead and cryptographic properties. In addition, this paper proposes a preprocessing method for 4×4 S-box logic gate expressions, which makes it easier to obtain better area, running time, and power data in ASIC implementation. Applying it to 14 classic lightweight block cipher S-boxes, the results show that is feasible. A series of performance tests and security evaluations were performed on the IIoTBC. As shown by experiments and data comparisons, IIoTBC is compact and secure in industrial IoT sensor nodes. Finally, IIoTBC has been implemented on a temperature state acquisition platform to simulate encrypted transmission of temperature in an industrial environment.

Keywords: Industrial IoT, Sensor nodes, Lightweight block cipher, S-box, Security design.

This research is supported by the Scientific Research Fund of Hunan Provincial Education Department with Grant No.21C0540, Open fund project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application for Hengyang Normal University(2021HSKFJJ041), Hunan Provincial Natural Science Foundation of China(2022JJ30103, 2022JJ50016), the Science and Technology Plan Project of Hunan Province (2016TP1020), "the 14th Five-Year Plan" Key Disciplines and Application-oriented Special Disciplines of Hunan Province(Xiangjiaotong [2022]351).

http://doi.org/10.3837/tiis.2023.01.006 **ISSN : 1976-7277**

1. Introduction

Industrial IoT is the largest and most important component of the IoT. It integrates various mobile intelligent terminals and communication technologies to upgrade traditional industrial models [1]. The enterprise centralizes the data collected in the industrial production process, equipment operation status information, etc. Then these data are exchanged, analyzed, and processed through various communication networks. However, the massive collection and processing of sensor data also raises some privacy concerns [2].

Xie et al. noted that due to the nature of wireless sensor networks (WSNs), they are vulnerable to many types of attacks [3]. Various security measures for the network layer have been proposed. Popoola et al. proposed a federated deep learning (FDL) approach for unknown (zero-day) botnet attack detection without any data privacy concerns [4]. This is important for improving network security. Zhao et al. proposed a novel network intrusion detection (NID) method for IoT based on the lightweight deep neural network (LNN) [5]. This not only has excellent performance in intrusion detection, but can also significantly reduce computational cost and model size. Xenofontos et al. proposed an attack taxonomy that considers the unique structure and operational constraints of IoT ecosystems [6]. According to the classification results, security countermeasures and best practices can be used to mitigate threats and vulnerabilities before they evolve into catastrophic attacks. Additionally, Xenofontos et al. reported nine IoT-based attacks, three of which belonged to the industrial IoT. Sengupta et al. reported the industrial IoT architecture, security issues, and types of attacks encountered in detail [7]. A variety of security protection schemes for the industrial IoT have been proposed [8-11]. Protecting data in transit is necessary, and many attacks more often target terminal devices. After contacting the hardware, attackers can use tools to extract data directly from nodes, or even install malicious devices, bypassing various security measures in the software and destroying data security. It is worth noting that nodes cannot provide enough hardware resources to support traditional encryption methods solidified into hardware circuits.

One of the methods of industrial IoT defense is to build a defense mechanism directly on the sensor node device. Dalmasso et al. noted that there is an inevitable trend to using secure and efficient lightweight block ciphers to protect the privacy of any resource-constrained device [12].

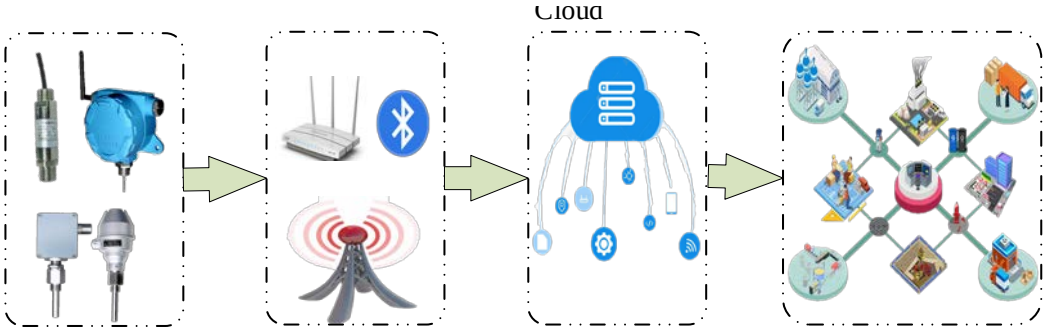

Fig. 1. Industrial IoT encryption model

As shown in **Fig. 1**, a lightweight block cipher is deployed at the perception layer, which allows sensor data to circulate as ciphertext. Then build the first line of defense for the entire

industrial IoT. Therefore, this paper mainly studies a secure lightweight block cipher with low hardware overhead and a simple structure. Specifically, it addresses the security requirements of the industrial IoT perception layer.

2. Related Work

Lightweight block ciphers can be divided into two categories based on their nonlinear components. The nonlinear components in the first category are mainly 4×4 S-boxes. Representative lightweight block ciphers include ULC [13], PRESENT [14], GIFT [15], Midori [16], Piccolo [17], and RECTANGLE [18]. The nonlinear components in the second category are mainly modular addition operations (⊞), AND operations (&), etc., such as HIGHT [19], SIMON and SPECK [20], Simeck [21], CHAM [22], Shadow [23], and SAND [24].

S-box is an important component of a lightweight block cipher. It has a direct impact on the overall security of the design. S-box is usually implemented in two ways: lookup table (LUT) and logic gates. The memory overhead required by the LUT method is high, which is not ideal for some devices with limited memory (such as smart cards and embedded devices) [25]. Logic gates are the basic structure of a digital system, which are usually combined to realize more complex logic operations. Easy logic implementation means that its S-box is only realized by a simple combination of a few logic gates.

At present, the main idea of S-box lightweight is to reduce the number of logic gates used. Some ciphers use only a few logic gates and then generate S-boxes based on iterative architecture, such as Piccolo, SKINNY [26]. For S-box designers, it is not easy to strike a good balance between hardware overhead and safety. For example, the two S-boxes used in the Midori trade security for low area. Piccolo and SKINNY trade latency for low area. Their special iterative architecture can result in relatively high S-box latency, which is undesirable in low-latency devices. At the same time, the SKINNY cipher S-box has a fixed point. The lightweight of the S-box should not be limited to sacrificing other S-box indicators. Therefore, this paper intends to explore a lightweight method for the 4×4 S-box without changing its cryptographic nature.

Modular addition and AND operations are simple and easy to implement, but they cannot be proven to meet the security requirements of the industrial IoT. In addition, as the most important part of the IoT, the industrial IoT also has higher requirements for the economy and efficiency of industrial production [27].

Feistel and SPN are the main structures used in lightweight block ciphers. SPN structure-based decryption generally consumes too much extra resources. For example, the decryption process needs to allocate resources for the inverse S-box. The Feistel structure uses the feature that XOR twice equals itself, and the decryption process can share the encryption structure highly. This does not require resource allocation for the inverse S-box, and the decryption hardware implementation does not consume too many additional resources. In resource-constrained embedded devices, area consumption is a major consideration.

It is worth noting that designers need to deal with the trade-off between security, lightweight, and performance of lightweight block ciphers [28]. As shown in **Fig. 2**, key size, rounds, and hardware architecture are the main balancing factors. The key size offers a trade-off between security and lightweight. It ranges from 56-bit to 256-bit. In terms of long-term protection requirements, the key size is set to 128-bit, which can achieve a better balance. The number of encryption rounds provides a balance between security and operational performance. In general, the higher the number of rounds, the higher the security achieved. But it inevitably

increases the running time and energy. The hardware architecture provides a balance between lightweight and performance. The main cryptographic hardware architectures include serial and parallel architectures. Serial architectures consume less area but have lower throughput. However, parallel architectures have high throughput but a large area. The choice of architecture depends on the actual needs of the application scenario.

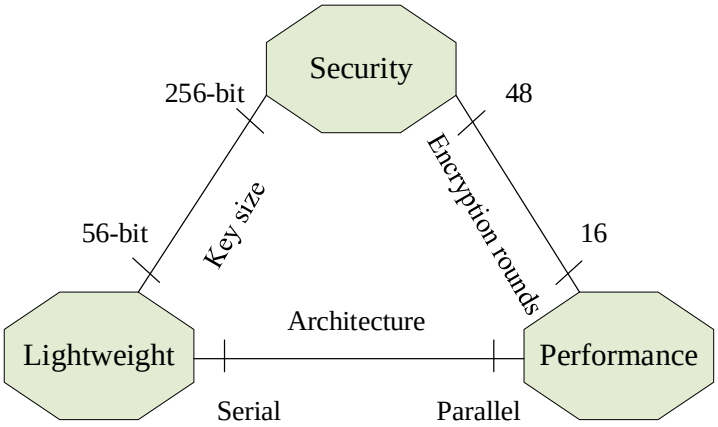

Fig. 2. The balance between security, lightweight, and performance

Our contribution This paper focuses on a lightweight block cipher IIoTBC for the security requirements of the industrial IoT sensor layer. IIoTBC accepts 128-bit keys with a block size of 64-bit and consists of 10 rounds. Secondly, the IIoTBC has a variable structure. Compared with a fixed cryptographic system, it can be better adapted to sensor nodes with different security requirements. In addition, a 4×4 S-box can be regarded as a non-repeating arrangement of elements on GF(2^4). Only part of the arrangement does not have the better cryptographic properties of the S-box. This paper randomly generates such permutations and uses the cryptographic properties of the S-box as a filtering condition. Randomly selected from eligible permutations, the area overhead is then used to assess their suitability for use as S-boxes. The S-box, which achieves a compromise between area overhead and cryptographic properties, will be selected as the nonlinear layer of IIoTBC. Also, this paper finds a preprocessing method for 4×4 S-box logic expressions. This can achieve better area, runtime, and energy in ASIC implementations without changing its cryptographic properties. Applying it to IIoTBC S-box and 14 classic lightweight block cipher S-boxes. The experimental results show that it is feasible. ASIC hardware implementation, avalanche test, impossible differential cryptanalysis, side channel cryptanalysis, and algebraic cryptanalysis are done for IIoTBC. As experiments and data comparisons show, IIoTBC is compact and secure in industrial IoT sensor nodes. Finally, this paper constructs an end-to-end temperature state acquisition platform. IIoTBC has been successfully applied to simulate encrypted transmission of temperature in an industrial environment.

3. Specification of IIoTBC

As shown in **Fig. 3**, the IIoTBC system structure is variable. The MCU size is used as a control signal to select one of them. Also, the widely used controller area network (CAN) bus size is generally 64-bit. Therefore, IIoTBC accepts 128-bit keys with a block size of 64-bit. Meanwhile, the number of encryption rounds r=32. This paper denotes the 64-bit plaintext as

$M_0, M_1, \cdots, M_6, M_7$. Each round subkey is represented by $key_j^i (1 \le i \le 4,\ 1 \le j \le r)$.

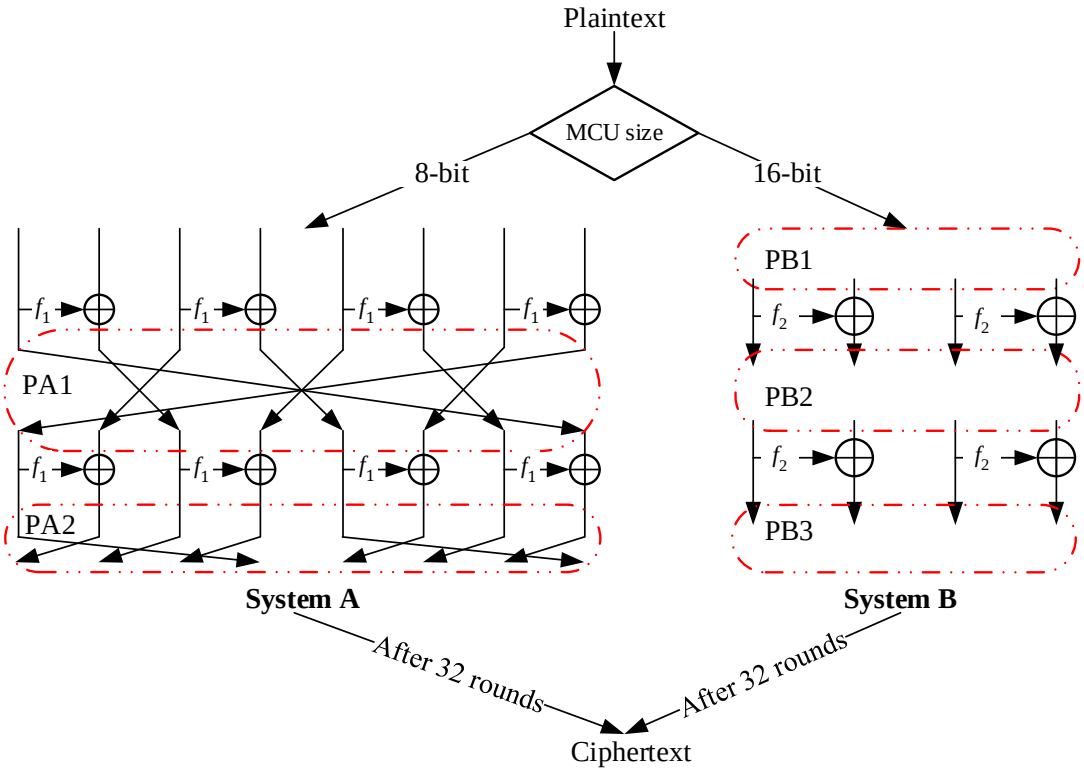

Fig. 3. The encryption process of IIoTBC

3.1 System Structure A

The embedded systems with 8-bit MCU are more common in the industrial IoT. Therefore, this paper focuses on the research based on System A. It is based on an 8-branch generalized Feistel structure. The approximate encryption process of it can be described by the pseudocode in Algorithm 1.

Algorithm1. IIoTBC System Structure A Encryption Routine
Input: Plaintext, key
Output: Ciphertext
1: $(M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7) \leftarrow Plaintext$
2: for $j = 1$ to r do
3: $M_0^{'} = M_0,\ M_1^{'} = f_1(M_0, key_j^1) \oplus M_1$
4: $M_2^{'} = M_2,\ M_3^{'} = f_1(M_2, key_j^2) \oplus M_3$
5: $M_4^{'} = M_4,\ M_5^{'} = f_1(M_4, key_j^3) \oplus M_5$
6: $M_6^{'} = M_6,\ M_7^{'} = f_1(M_6, key_j^4) \oplus M_7$
7: if ($j\%2 != 0$)
8: $M_{state} = \mathrm{PA1}(M_0^{'}, M_1^{'}, M_2^{'}, M_3^{'}, M_4^{'}, M_5^{'}, M_6^{'}, M_7^{'})$
9: else

10: $M_{state} = \mathrm{PA2}(M_0^{'}, M_1^{'}, M_2^{'}, M_3^{'}, M_4^{'}, M_5^{'}, M_6^{'}, M_7^{'})$
11: else if j==r
12: $Ciphertext \leftarrow (M_0^{'}, M_1^{'}, M_2^{'}, M_3^{'}, M_4^{'}, M_5^{'}, M_6^{'}, M_7^{'})$
13: end for
Return Ciphertext

System structure A mainly includes f_1 function, PA1, and PA2. Among them, PA1 and PA2 mainly realize the exchange of branch data positions. The specific exchange process has been given in **Fig. 3**. The operation process of the f_1 function includes AddRoundkey, S-box permutation, and 1-bit left roation ($\lll 1$), as shown in **Fig. 4**. The values of S-box elements are shown in **Table 1**.

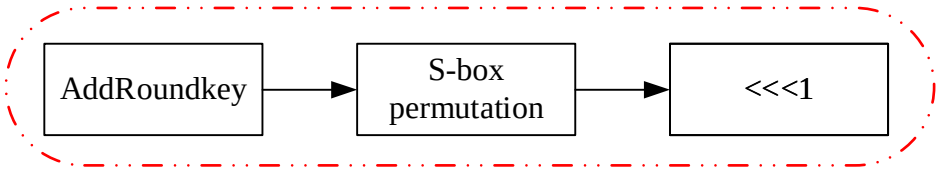

Fig. 4. The operation process of f_1

Table 1. The 4×4 bijection S-box of IIoTBC

i	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(i)$	5	D	9	4	6	3	F	1	B	8	E	0	7	2	C	A

3.2 System Structure B

Currently, 8-bit MCUs still dominate the mainstream low-end market. Compared with this, 16-bit MCUs have better data processing capability and more memory. If advanced security features are needed, it may be wise to use 16-bit. Thus, this paper provides a more complex system B for 16-bit MCUs. System B is based on a 4-branch generalized Feistel structure. System B is designed based on bit-slice technology, and each branch is a 4×4 matrix. PB1 is to convert the 64-bit data to be processed into 4 slices, as shown in **Fig. 5**. Taking the first branch as an example, the illustrated f_2 function operation process is shown in **Fig. 6**. PB2 and PB3 are used for odd and even rounds, respectively. Their operation process is shown in **Fig. 7** and **Fig. 8**.

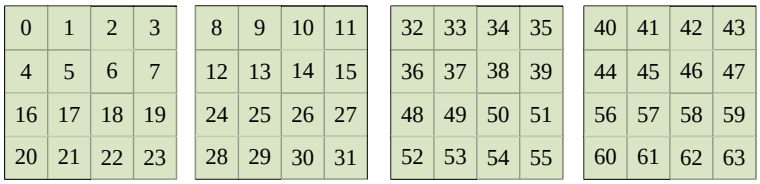

Fig. 5. The operation process of PB1

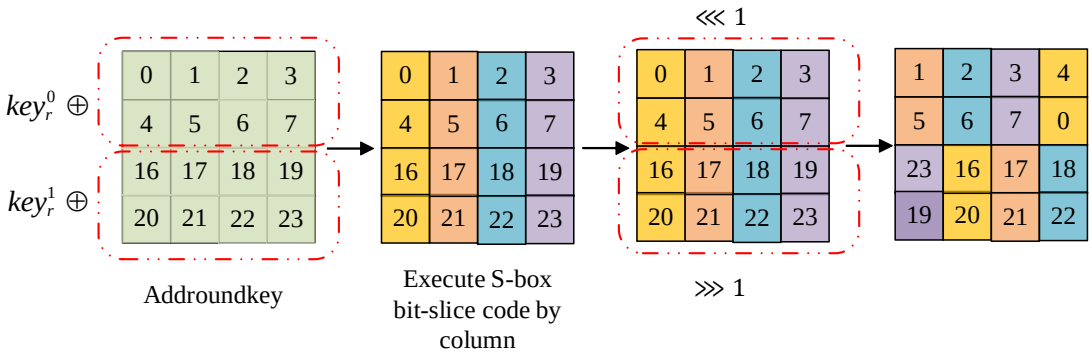

Fig. 6. The operation process of f_2

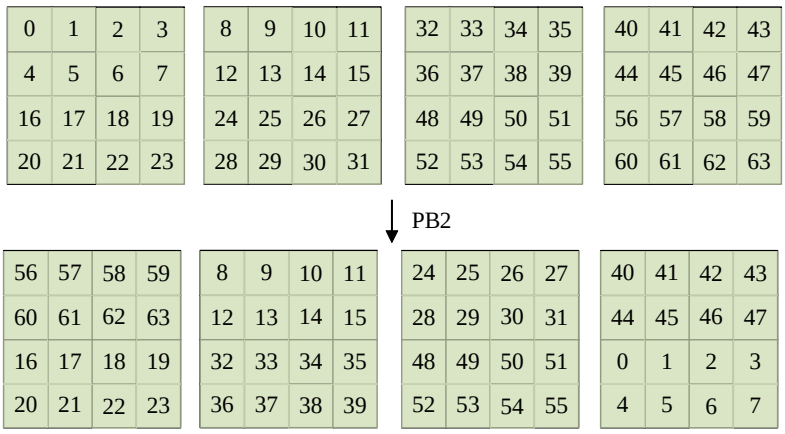

Fig. 7. The operation process of PB2

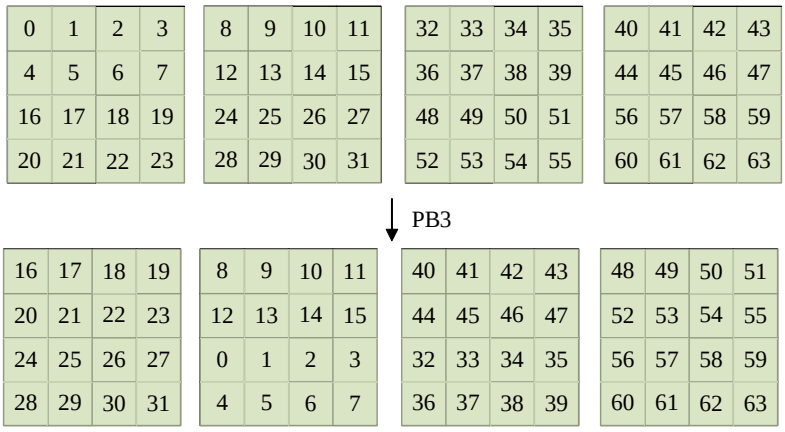

Fig. 8. The operation process of PB3

3.3 Subkey Generation

In this paper, the key size supported by IIoTBC is 128-bit. Firstly, for the convenience of description, $k_0 \| k_1 \| k_2 \cdots k_{125} \| k_{126} \| k_{127}$ is used to represent the primary 128-bit keys. Secondly, the relationship between $reg_1, reg_2, reg_3, reg_4$, and 128-bit keys can be expressed by (1).

$$reg_1 = k_0 \| k_1 \cdots k_{30} \| k_{31}, reg_2 = k_{32} \| k_{33} \cdots k_{62} \| k_{63}$$
$$reg_3 = k_{64} \| k_{65} \cdots k_{94} \| k_{95}, reg_4 = k_{96} \| k_{97} \cdots k_{126} \| k_{127} \quad (1)$$

reg_1, reg_2, reg_3, and reg_4 are used sequentially for rounds 1 to 4 of the AddRoundkey operation. For rounds 5 to r, the subkey generation method satisfies (2). In the first round of AddRoundkey, the 8-bit $k_0 \| k_1 \cdots k_6 \| k_7$ serves as key_1^1, the 8-bit $k_8 \| k_9 \cdots k_{14} \| k_{15}$ serves as key_1^2, the 8-bit $k_{16} \| k_{17} \cdots k_{22} \| k_{23}$ serves as key_1^3, and the lowest 8-bit $k_{24} \| k_{25} \cdots k_{30} \| k_{31}$ serves as key_1^4.

$$reg_u = f_3(reg_{u-4}) \oplus reg_{u-1}, 5 \le u \le r \quad (2)$$

Among them, the f_3 function includes P_3 permutation and 4×4 S-box permutation. The operation process of f_3 is shown in **Fig. 9.**

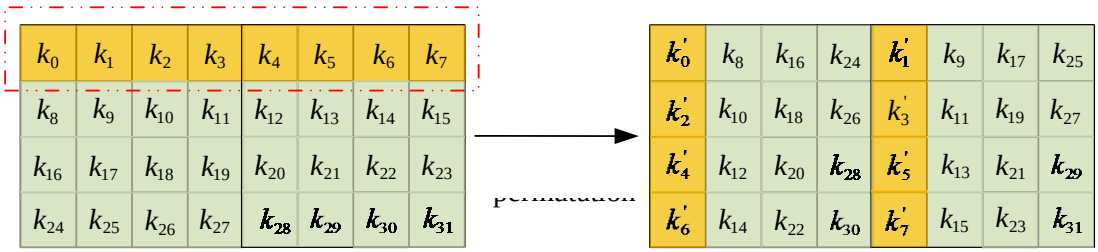

Fig. 9. The operation process of f_3

The 4×4 S-box used by the f_3 function is the same as the S-box used by the encryption round function. The position transformation corresponding to the specific P_3 permutation is shown in **Table 2**.

Table 2. The values of P_3 permutation

k_i	k_i'	k_i	k_i'	k_i	k_i'	k_i	k_i'
0	0	8	2	16	4	24	6
1	8	9	10	17	12	25	14
2	16	10	18	18	20	26	22
3	24	11	26	19	28	27	30
4	1	12	3	20	5	28	7
5	9	13	11	21	13	29	15
6	17	14	19	22	21	30	23
7	25	15	27	23	29	31	31

3.4 Decryption Algorithm

IIoTBC is based on the generalized Feistel structure, which is highly consistent in encryption and decryption. There is no difference between the decryption round function and the encryption round function. It should be noted that the permutation operation connects the upper and lower rounds. In addition to controlling logical resources, it will not consume too many additional resources. The reverse order of the round key in the encryption process is

used in the decryption process.

4. Design Rationale of IIoTBC

4.1. The structure of IIoTBC

IIoTBC system structure is variable, including A and B. In view of the wide application of 8-bit MCU, A is our main structure. Its round function only involves AddRoundkey, S-box permutation, and 1-bit left rotation ($\lll 1$). The AddRoundkey operation prevents attackers from using potential cryptanalysis methods to bypass the S-box operation and bring security risks to the cipher. Nonlinear S-box is an important part of lightweight block cipher, and its cryptographic properties directly affect the security of the overall algorithm. The 1-bit left rotation makes the influence of a 1 bit on the branch propagate to the remaining bits. IIoTBC cipher uses a 4×4 lightweight S-box. 1-bit left rotation can be realized through wire during hardware implementation. Therefore, IIoTBC round function has the characteristics of simple operation and less resource consumption.

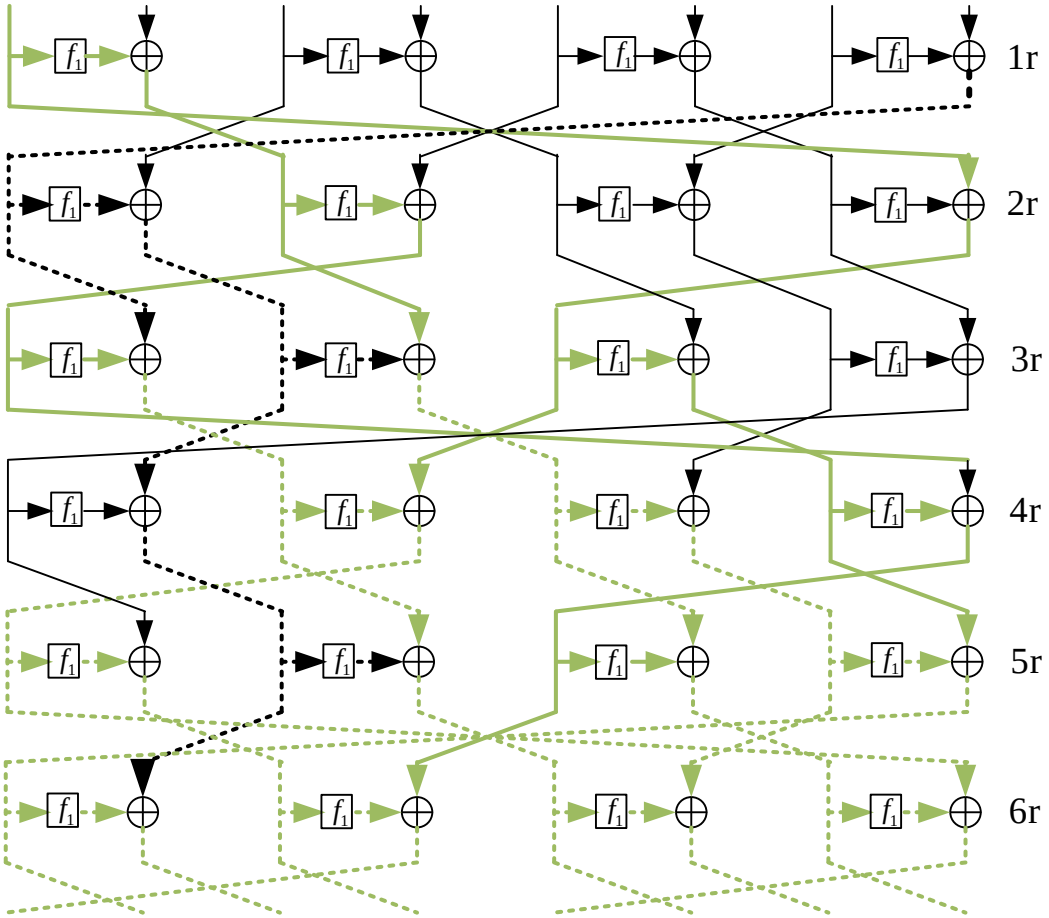

Fig. 10. IIoTBC cipher system A P permutation effect

For Feistel structure, P permutation is often used to connect the upper and lower rounds of encryption operations. A good P permutation can make each branch data in the cryptographic structure affect all the remaining branch data. In particular, this process does not involve the AddRoundkey operation. An involutive P permutation can usually make the encryption and decryption of the cipher with Feistel structure consistent. Similarly, for an 8-branch generalized Feistel structure, one branch data can affect at most four branches if only involutive permutations are used in successive rounds of encryption. Although encryption and decryption are consistent, confusion and diffusion are less effective. Of course, in the 8-branch generalized Feistel structure, it is possible to use only non-involutive P permutation, such as HIGHT. However, without the effect of AddRoundkey operation, HIGHT usually requires 6 to 8 rounds to achieve full diffusion. IIoTBC cipher system A alternately uses involute PA1 permutation and non-involute PA2 permutation. As shown in **Fig. 10**, full diffusion can be achieved in round 6. In addition, the IIoTBC cipher system A dynamically selects one of the P permutation methods for each round encryption. After 32 rounds of iteration, the attack difficulty coefficient increased to a certain extent.

4.2. Nonlinear Layer

Most lightweight block ciphers use S-boxes for their nonlinear layers. The security of ciphers also mainly depends on this. An $n \times m$ S-box can be seen as the mapping $S: F_2^n \to F_2^m$ or as an n-element Boolean function $S(X) = (f_1(X), f_2(X), \cdots, f_m(X))$. The rapid growth of resource-constrained devices in the context of industrial IoT applications has spawned the need for lightweight implementation of S-box. Therefore, this paper mainly considers the situation of the bijective S-box when $n = m = 4$.

Our goal is to get a safe and lightweight 4×4 S-box. In fact, the 4×4 S-box can be seen as a possible arrangement of 0 to 15. It is just that some of the permutations do not have the S-box security attributes. S-box security attributes are usually evaluated by indicators such as nonlinearity, differential uniformity, fixed-point numbers, and algebraic degree. This paper randomly generates 0 to 15 possible permutations, using the security properties of the S-box as a filter condition. Within the eligible permutations, one is randomly selected to test its area overhead. The S-box, which achieves a compromise between area overhead and cryptographic properties, will be selected as the nonlinear layer of IIoTBC. These security attributes of the S-box are described below.

Differential uniformity (DU) of n-bit S-box f is defined as follows [29].

$$DU(F) = \max_{\Delta I, \Delta Y \in GF(2^n)} \left| \left\{ x \in GF(2^n) : f(x + \Delta I) + f(x) = \Delta Y \right\} \right| \tag{3}$$

x denotes all possible input values, ΔI denotes input difference, and ΔY denotes output difference. The maximum value in the differential distribution table (not considering the case of $\Delta I = \Delta Y = 0$) is the DU value of an S-box. The smaller the DU, the stronger the ability of S-box to resist differential cryptanalysis.

Nonlinearity of f_x can be seen by Walsh spectrum [30].

$$N(f)=2^{n-1}(1-2^{-n}\max_{\omega\in GF(2^n)}\mid S_f(\omega))$$

$$S_f(\omega)=\sum_{\omega\in GF(2^n)}-1^{f(x)\oplus x*\omega} \tag{4}$$

$$\omega\in GF(2^n)\Rightarrow x*\omega=x_1*\omega_1\oplus\cdots\oplus x_n*\omega_n$$

A *n*-bit Boolean function f can be expressed as a multivariate polynomial on the field $GF(2)$, which is the algebraic normal form (ANF) as shown below [29].

$$f(x_1,\cdots,x_n)=a_0+a_1\cdot x_1+\cdots+a_{1,2}\cdot x_1\cdot x_2+a_{1,2,\cdots,n}\cdot x_1\cdot x_2\cdots x_n \tag{5}$$

Where $a_0,a_1,\cdots,a_n,a_{1,2,\cdots},a_{1,\cdots,n}\in GF(2)$. The number of variables in the largest monomial of ANF is called algebraic degree (AD), as shown in the (6). The optimal number of algebras for each component function of an excellent S-box is *n*-1. The number of algebraic terms is determined by the sum of the algebraic terms between all the component functions.

$$AD(f)=max\{deg(f_1),deg(f_2),\cdots,deg(f_n)\} \tag{6}$$

Let x_0 , x_1 , x_2 , and x_3 represent the input of the S-box. Let y_0 , y_1 , y_2 , and y_3 represent the output of S-box. The algebraic norms of the components y_0 , y_1 , y_2 , y_3 of IIoTBC S-box are shown as (7).

$$\begin{aligned}
y_0&=x_0{*}x_1{*}x_2+x_0{*}x_1{*}x_3+x_0{*}x_1+x_0{*}x_2{*}x_3+x_0{*}x_2+x_0{*}x_3+x_0+x_1{*}x_2{*}x_3+x_1{*}x_3+x_2+x_3;\\
y_1&=x_0{*}x_1+x_0+x_1{*}x_2{*}x_3+x_1{*}x_2+x_1{*}x_3+x_2{*}x_3+x_2+1;\\
y_2&=x_0{*}x_1{*}x_2+x_0{*}x_1{*}x_3+x_0{*}x_1+x_0{*}x_3+x_0+x_1{*}x_2{*}x_3+x_1;\\
y_3&=x_0{*}x_1{*}x_2+x_0{*}x_1{*}x_3+x_0{*}x_1+x_0{*}x_2+x_0{*}x_3+x_1{*}x_2+x_1{*}x_3+x_1+x_2{*}x_3+1;
\end{aligned} \tag{7}$$

In addition, this paper compares the security properties of 14 classical cipher S-boxes with those of IIoTBC. The test results are shown in **Table 3**.

Table 3. Comparison of cryptographic properties between IIoTBC S-box and other cipher S-boxes

S-box	DU	NL	AD	AT	FP
IIoTBC	4	4	3,3,3,3	11,8,7,10	0
PRESENT	4	4	3,3,3,2	8,8,7,4	0
Piccolo	4	4	2,2,3,3	5,5,8,9	0
RECTANGLE	4	4	3,3,2,2	6,8,5,4	0
GIFT	6	4	3,3,2,2	3,5,5,6	0
Midori-Sb0	4	4	3,3,2,3	6,7,5,6	4
Midori-Sb1	4	4	3,3,3,3	6,9,7,8	4
mCryption-S0	4	4	3,3,2,3	7,8,8,9	0
mCryption-S1	4	4	3,3,3,3	7,9,7,8	0

mCryption-S2	4	4	3,3,3,3	6,10,8,9	0
mCryption-S3	4	4	3,3,3,3	9,6,10,8	0
KLEIN	4	4	3,3,3,3	6,9,8,9	0
PRINCE	4	4	3,3,3,3	8,7,6,8	0
SKINNY	4	4	2,2,3,3	5,5,7,9	1
TWINE	4	4	3,3,3,3	8,9,6,6	1

Satisfy that the differential uniformity is 4. The maximum differential probability and the maximum linear probability are 2^{-2}. There is no fixed point. The algebraic degree of y_0 , y_1 , y_2 , y_3 are 3. And the algebraic terms, the higher the better. A 4×4 S-box satisfying the above conditions can be considered to have good cryptographic properties. As shown in **Table 3**, most S-boxes achieve ideal values for differential uniformity and nonlinearity. The algebraic degree of each component function of IIoTBC, Midori-Sb1, mCryption, KLEIN, PRINCE, and TWINE has reached the ideal value of 3. At the same time, IIoTBC has the largest number of algebraic terms and no fixed point. Therefore, it can be considered that the IIoTBC S-box has good cryptographic properties.

5. Hardware Implementation

It is particularly important to study nonlinear and linear components that have a small hardware overhead and achieve optimal cryptographic properties for lightweight block ciphers. 1-bit left rotation and P permutation can be realized by wire in the hardware implementation, so that no much additional hardware resources are consumed. Therefore, the 4×4 S-box is the main resource consuming component in the round function.

The physical area required by the cipher is usually expressed in μm^2. It can be represented by GE in ASIC, and a 2-input NAND gate is equal to 1 GE. IIoTBC has been synthesized using Synopsys Design Compiler version L-2016.03-SP1, which is based on the SMIC 0.13μm technology library. In this subsection, an S-box logic expression preprocessing approach proposed in this paper will be presented, followed by the overall hardware overhead of the IIoTBC.

5.1 An S-box Logic Expression Preprocessing Approach

The "set_max_area 0" command can be set during ASIC synthesis, so that the design area after synthesis is as small as possible. Although the designer may specify the S-box logic gate expression, the actual circuit after ASIC synthesis does not necessarily match that specified by the designer. It can be found that the logical expression of the S-box is simplified to a minimum by the Karnaugh map method. Then through the "set_max_area 0" instruction, the synthesized result is not necessarily optimal. The ASIC synthesis cannot be guaranteed to match the logic gates specified by the designer. But it can be used as a basis for synthesis optimization. Therefore, ASIC synthesis results depend on the form of logic gate expressions.

In hardware circuits, AND gates, OR gates, and NOT gates are the basic logic gates. It is worth noting that in most process libraries, the GE number of NAND gates and NOR gates is smaller than that of AND gates and OR gates, such as UMCL18G212T3 technology library. This paper combines these two phenomena and proposes a method for preprocessing S-box logic gate expressions. That is, try to use NAND gates, NOR gates, NOT gates, etc. to describe the logic gate expression of the S-box. Then send it to the ASIC platform for synthesis. This paper tested 14 classical lightweight block cipher S-boxes and the IIoTBC cipher S-box to

confirm the effectiveness of this method. The test results are shown in **Table 4**. Where "✓" indicates that our method is feasible, and "〇" indicates that the data are consistent before and after processing.

Table 4. Comparison of area, time and power of each S-box before and after processing

S-box	Before			After			Effect
	Area (μm^2)	Time (ns)	Power (μW)	Area (μm^2)	Time (ns)	Power (μW)	
IIoTBC	87.14	0.49	0.001899	82.60	0.41	0.001800	✓
PRESENT	95.32	0.46	0.002077	83.51	0.38	0.001819	✓
Piccolo	59.00	0.53	0.005900	59.91	0.42	0.001305	✓
RECTANGLE	107.11	0.48	0.002334	80.79	0.39	0.001760	✓
GIFT	84.42	0.44	0.001839	84.42	0.44	0.001839	〇
Midori-Sb0	46.30	0.2	0.001009	46.30	0.2	0.001009	〇
Midori-Sb1	57.77	0.23	0.001259	57.77	0.23	0.001259	〇
mCryption-S0	105.3	0.54	0.002294	94.40	0.55	0.002057	✓
mCryption-S1	96.22	0.51	0.002096	95.94	0.41	0.002090	✓
mCryption-S2	101.66	0.51	0.002215	98.95	0.50	0.002156	✓
mCryption-S3	100.76	0.68	0.002195	93.50	0.44	0.002037	✓
KLEIN	96.22	0.51	0.002096	96.22	0.51	0.002096	〇
PRINCE	60.82	0.25	0.001325	54.46	0.25	0.001187	✓
SKINNY	74.44	0.47	0.001622	74.44	0.47	0.001622	〇
TWINE	100.76	0.65	0.002195	94.40	0.41	0.002057	✓

According to **Table 4**, 10 of the 15 S-boxes obtained better power, area, and running time data. In addition, **Table 4** shows that more than half of S-boxes have larger area, running time, and power data than the IIoTBC S-box. The Midori-S0 hardware test data is the best, and the RECTANGLE, SKINNY data is close to that of the IIoTBC S-box. However, combined with the test of the cryptographic properties of the S-box in the previous chapter, it can be seen that Midori-S0 has four fixed points. Therefore, it can be said that the IIoTBC S-box has achieved a good balance between hardware overhead and encryption performance.

5.2 The Hardware Overhead of the IIoTBC Cipher

Similarly, this paper synthesizes IIoTBC system A on the ASIC platform. **Fig. 11** is a system diagram of it. The result of system A area implementation is 8123.38 μm^2, which is equal to 1769.80GEs. Also, latency is 32. At 100MHz, the throughput is $(100MHz \times 64)/32 = 200Kbps$. The amount of power required by the circuit to process system A is 1.77 μW. The result of system B area implementation is 2338.12GEs. Latency is 32. At 100MHz, the throughput is $200Kbps$. The amount of power required by the circuit to process system B is 2.34 μW.

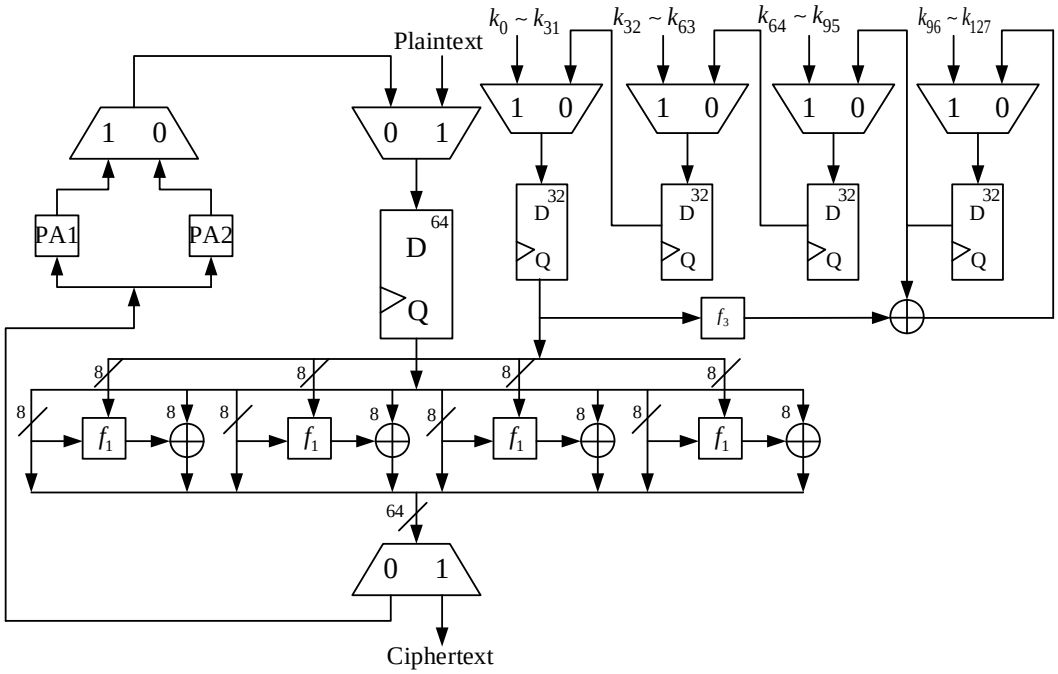

Fig. 10. The system diagram of system A

In this paper, appropriate ciphers are selected for comparison to give a fair indication of the overall hardware resource consumption of the IIoTBC. First, select ciphers with 64-bit block size and 128-bit key size. Then select the ciphers for simulation under the SIMC 0.13μm technology library. Ciphers meeting the above conditions are listed in **Table 5**. It can be seen from **Table 5** that the hardware resources consumed by IIoTBC have certain advantages in the same area.

Table 5. Comparison between the hardware resources of the IIoTBC and other ciphers

Ciphers	Block size	Key size	Latency	Thr. (*Kbps*)	Area (*GEs*)	Power (μW)	Tech. (μm)
LED [31]	64	128	48	133.33	3194	3.19	0.13
RECTANGLE [31]	64	128	26	246	1787	1.79	0.13
RECTANGLE [32]	64	128	26	246	2063.5	2.06	0.13
mCryption [31]	64	128	13	492.3	2949	2.95	0.13
mCryption [31]	64	128	13	492.3	4108	4.11	0.13
mCryption [31]	64	128	191	33.5	2709	2.71	0.13
XTEA [31]	64	128	32	200	2521	2.52	0.13
System A	64	128	32	200	1769.80	1.77	0.13
System B	64	128	32	200	2338.12	2.34	0.13

6. Security Analysis

6.1 Avalanche test

In cryptography, the avalanche effect means that when a binary bit of the input is inverted, 1/2 of the binary in the output is inverted. A block cipher is considered to have poor randomization properties if it does not exhibit a certain degree of avalanche characteristics. This can lead to partial or even complete cracking of the algorithm. For designers, satisfying the avalanche effect is an essential criterion.

This paper randomly selects 10,000 64-bit samples to test the change in output bits when the IIoTBC cipher system A input changes by 1 bit. When the input bits change 1-bit, the number of output bits change ω is expected to be 1/4~3/4 of the output size, and the mean value ω_a is approximately 1/2. The Strict Avalanche Criterion (SAC) is a formalization of avalanche effects. It states that when any input bit is inverted, every output bit has a 1/2 probability of changing. When the input bits change by 1 bit, the mean value of the output change ω_a is close to 32, and the mean probability of output change Pr_a is close to 0.5. It can be considered that the encryption algorithm has a good avalanche effect and can resist statistical analysis.

Table 6. IIoTBC cipher avalanche effect test results

Round	$\boldsymbol{\omega}_{\mathbf{max}}$	$\boldsymbol{\omega}_{\mathbf{min}}$	$\boldsymbol{\omega}_{\mathbf{a}}$	Pr_{max}	Pr_{min}	Pr_a
9	50	12	31.964827	0.501139	0.498245	0.499450
10	51	15	31.9992275	0.501394	0.498514	0.499989
11	50	14	32.004723	0.501411	0.498683	0.500074
32	50	13	31.999498	0.501569	0.497775	0.499992

The test results are shown in **Table 6**. IIoTBC cipher system A reaches full dependency after 10 rounds. For the 32-round IIoTBC cipher system A, the input changes 1-bit, and the average value of the number of digits changed by the output ω_a is close to 32. The value range of the probability Pr of each bit output change is between 0.497775~0.501569. The average probability Pr_a of each bit output change is 0.499992, which is very close to 0.5.

In addition, this paper uses the 10,000 test samples to test the changes in the output bits of algorithms such as CHAM, HIGHT, Piccolo when the input bits change 1 bit in each round as shown in **Fig. 11**. The experimental test results are compared with IIoTBC system A, as shown in **Table 7**.

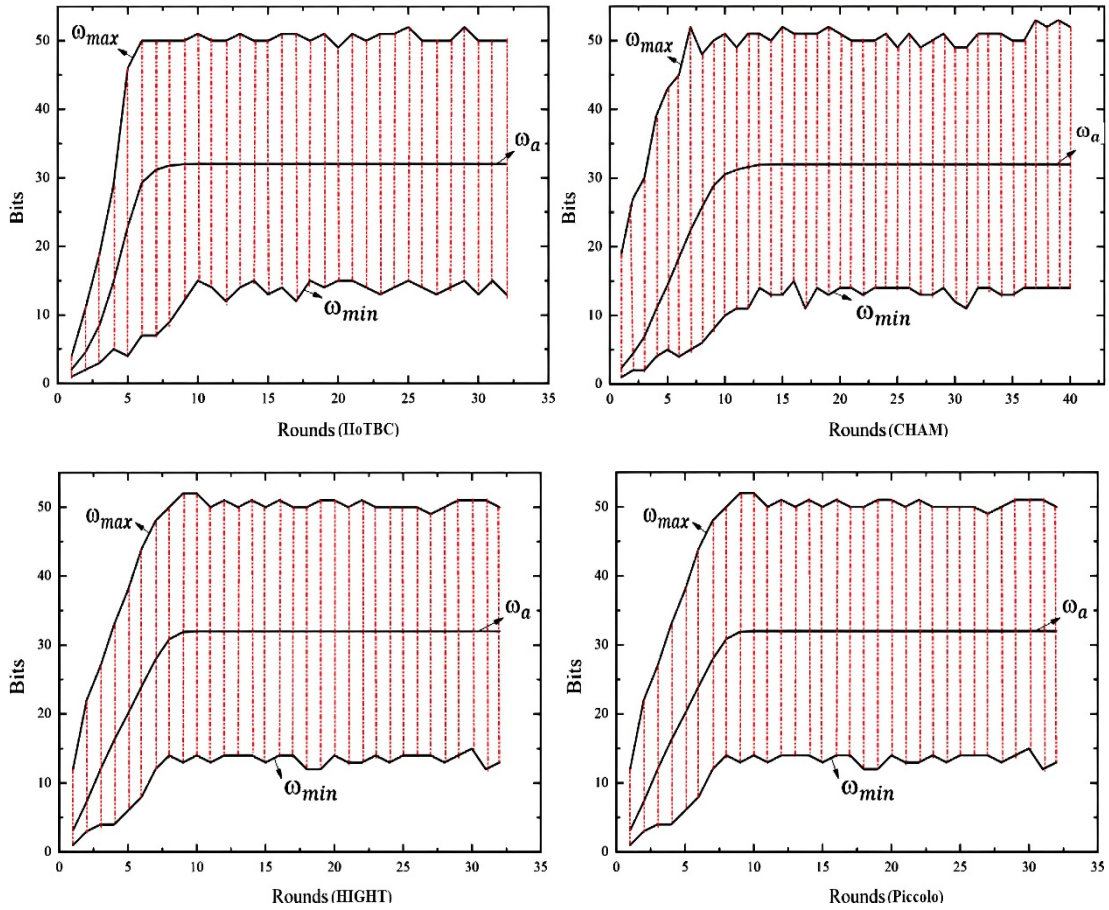

Fig. 11. Changes of output bits of IIoTBC, CHAM, HIGHT and Piccolo in each round

Table 7. Comparison of avalanche effect test results of IIoTBC with CHAM, HIGHT and Piccolo

Cipher	ω_{max}	ω_{min}	ω_a	Pr_{max}	Pr_{min}	Pr_a
IIoTBC	50	13	31.999498	0.501569	0.497775	0.499992
CHAM	52	14	32.005102	0.502036	0.498948	0.500080
HIGHT	50	13	32.004072	0.501934	0.498756	0.500064
Piccolo	50	12	31.994864	0.501097	0.498627	0.499920

As can be seen from **Table 7**, the input of system A changes by 1 bit, and the mean value and mean probability of the output change are closest to the ideal value. Therefore, it can be considered that the avalanche effect of IIoTBC cipher system A has certain advantages in the algorithm of the same type of GFN structure, and can resist statistical analysis.

6.2 Impossible differential cryptanalysis

The main idea of impossible differential analysis is to reduce the number of key searches using differential features with probability 0. That is, if under a certain guessing key, a plaintext ciphertext pair of the cipher can make the differential feature with a probability of 0 appear. Then this guessing key must be wrong and will be eliminated. Assuming that E is a lightweight block cipher encryption algorithm, $\Delta\alpha$ and $\Delta\beta$ are its input difference and output difference, respectively. Perform r_1 rounds of encryption operation on $\Delta\alpha$, and find a differential feature E_0: $\Delta\alpha \to \Delta\gamma_1$ with a probability of 1. Perform r_2 rounds of decryption on $\Delta\beta$, and at the same time find a differential feature E_1: $\Delta\gamma_2 \leftarrow \Delta\beta$ with a probability of 1. Using the meet-in-the-middle technique, if there is always a contradiction between γ_1 and γ_2, then $\Delta\alpha \nrightarrow \Delta\beta$ is called an impossible differential feature of $r_1 + r_2$ rounds of encryption algorithm E.

Table 8 is the differential distribution table of the IIoTBC S-box. According to **Table 8**, when the input difference or output difference of the S-box is a fixed value, the output difference or input difference ****, where * means the difference is unknown. A 6-round impossible differential feature of the IIoTBC cipher system A is found, as shown in (8).

$$\begin{aligned}&(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(a,0),(0,0) \nrightarrow \\ &(0,0),(0,0),(0,c),(0,0),(0,0),(0,0),(0,0),(0,0)\end{aligned} \tag{8}$$

Table 8. Differential distribution table for IIoTBC S-box

Δx	Δy															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	4	2	0	2	0	0	0	0	2	4	0
2	0	0	2	0	0	2	0	0	4	4	0	2	2	0	0	0
3	0	2	0	0	2	0	2	2	0	0	0	2	2	2	2	0
4	0	0	2	2	0	2	2	0	0	0	4	0	2	0	2	0
5	0	0	0	0	2	0	2	0	2	2	0	4	2	0	0	2
6	0	0	4	0	0	0	0	4	0	2	2	0	2	0	0	2
7	0	2	4	0	4	0	0	2	0	0	2	0	2	0	0	0
8	0	4	0	2	2	2	0	2	0	0	0	2	0	0	2	0
9	0	0	0	0	4	2	2	0	0	2	2	0	0	4	0	0
A	0	0	0	2	2	0	0	0	2	2	2	2	2	2	0	0
B	0	2	0	2	0	2	2	0	0	0	0	0	2	2	0	4
C	0	4	2	0	0	2	0	0	0	0	0	2	0	2	2	2
D	0	0	0	2	0	0	0	2	4	0	2	0	0	0	2	4
E	0	0	0	2	2	0	2	2	2	4	0	0	0	0	2	0
F	0	2	0	2	0	0	2	2	0	0	2	2	0	2	0	2

According to the 6-round impossible difference, the difference propagation characteristics are studied from the direction of encryption 3-round and the direction of decryption 4-round. Then the 13-round impossible difference is obtained as shown in **Table 9**.

Table 9. 13-round impossible differential path of IIoTBC

	ΔM_0	ΔM_1	ΔM_2	ΔM_3	ΔM_4	ΔM_5	ΔM_6	ΔM_7
Δr_0	$(0,0)$	(b_1,b_2)	$(a,0)$	(d_5,d_6)	(d_7,d_8)	(d_9,d_{10})	$(0,0)$	$(0,0)$
Δr_1	$(0,0)$	$(a,0)$	(b_1,b_2)	(d_3,d_4)	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
Δr_2	$(a,0)$	(b_1,b_2)	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
Δr_3	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(a,0)$
Δr_4	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(a,0)$	$(0,0)$
6-round impossible difference								
Δr_{10}	$(0,0)$	$(0,0)$	$(0,c)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
Δr_{11}	$(0,0)$	$(0,c)$	$(0,0)$	$(0,0)$	(d_1,d_2)	$(0,0)$	$(0,0)$	$(0,0)$
Δr_{12}	$(0,c)$	$(0,0)$	$(0,0)$	$(0,0)$	(d_3,d_4)	$(0,0)$	$(0,0)$	(d_1,d_2)
Δr_{13}	(d_1,d_2)	$(0,0)$	(d_5,d_6)	(d_3,d_4)	$(0,0)$	$(0,0)$	(d_7,d_8)	$(0,c)$

Choose the 2^N plaintext structures shown in (9).

$$\begin{aligned} &P^0_{M_0}=(\alpha_1,\alpha_2), P^0_{M_1}=(x_1,x_2), P^0_{M_2}=(x_3,\alpha_3), P^0_{M_3}=(x_4,x_5),\\ &P^0_{M_4}=(x_6,\alpha_7), P^0_{M_5}=(x_8,x_9), P^0_{M_6}=(\alpha_4,\alpha_5), P^0_{M_7}=(\alpha_6,\alpha_7) \end{aligned} \tag{9}$$

Among them, $\alpha_i(1\le i\le 7)$ takes a fixed value, and $x_i(1\le j\le 9)$ takes all possible values. Obviously, each plaintext structure has 2^{36} plaintexts, and the number of plaintext pairs that can be formed is 2^{71}. Then 2^N plaintext structures undergo 13 rounds of encryption, correspondingly, 2^{N+71} ciphertext pairs can be obtained. After 13 rounds of encryption, select the ciphertext pair that satisfies the difference form shown in (10).

$$\begin{aligned} &\Delta C^{13}_{M_0}=(d_1,d_2), \Delta C^{13}_{M_1}=(0,0), \Delta C^{13}_{M_2}=(d_5,d_6), \Delta C^{13}_{M_3}=(d_3,d_4),\\ &\Delta C^{13}_{M_4}=(0,0), \Delta C^{13}_{M_5}=(0,0), \Delta C^{13}_{M_6}=(d_7,d_8), \Delta C^{13}_{M_7}=(0,c) \end{aligned} \tag{10}$$

Among them, c and $d_i(1\le i\le 8)$ are non-zero values. Under this 13-round impossible differential feature path, the data complexity of the attack is 2^{N+36} selected plaintexts.

6.3 Side channel cryptanalysis

Many lightweight block ciphers are less likely to be designed with side channel immunity in mind. However, sensor nodes will inevitably leak side channel information in actual deployment such as heat, electromagnetic, and energy. Attackers can use these to analyze and attack industrial IoT sensor nodes. Rabie A et al. noted that the bit-slice implementation supports immunity against side channel analysis [33]. Therefore, this paper extends system A to obtain system B and implements it with bit-slicing technology. This applies to nodes with

higher level security requirements and relatively relaxed device resources. Nodes that are extremely resource sensitive can simply use system A for encryption.

Let x_0, x_1, x_2, and x_3 represent the four *n*-bit inputs of the S-box. Let y_0, y_1, y_2, and y_3 represent the four *n*-bit outputs of S-box. Let t_i denote the 4-bit temporary variables, i =0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 20, 21. The bit-slice implementation codes of the IIoTBC cipher S-box are shown in Algorithm2. Where "~" is bitwise NOT, "&" is bitwise AND, " | " is bitwise OR. Of course, the immunity of resource-constrained nodes to side channel analysis can also be improved by the first-order mask technique, which will not be discussed here.

Algorithm2. IIoTBC S-box bit-slice implements code		
Input: x_0, x_1, x_2, x_3		
Output: y_0, y_1, y_2, y_3		
$1: t_0 = \sim x_1;$	$2: t_1 = t_0 \oplus x_2;$	$3: t_2 = t_1 \mid x_2;$
$4: t_3 = x_2 \oplus x_3;$	$5: t_4 = t_3 \& t_2;$	$6: t_5 = t_1 \& x_0;$
$7: y_0 = t_5 \mid t_4;$	$8: t_7 = x_1 \mid x_2;$	$9: t_8 = \sim t_7;$
$10: t_9 = x_0 \mid x_1;$	$11: t_{10} = t_8 \mid x_3;$	$12: y_1 = t_{10} \oplus t_9;$
$13: t_{12} = x_0 \oplus x_3;$	$14: t_{13} = t_{12} \& x_0;$	$15: t_{14} = t_{12} \& x_2;$
$16: t_{15} = t_{14} \oplus t_{13};$	$17: t_{16} = t_{15} \mid x_1;$	$18: y_2 = t_{16} \oplus t_{14};$
$19: t_{18} = x_2 \mid x_3;$	$20: t_{19} = t_3 \mid x_1;$	$21: t_{20} = \sim t_{19};$
$22: t_{21} = t_{20} \mid x_0;$	$23: y_3 = t_{21} \oplus t_{18};$	

6.4. Algebraic cryptanalysis

Algebraic attack transforms the analytical problem of block ciphers into solving multivariate higher order algebraic equations in finite fields. The attacker can obtain the unknown key by solving the equation system. Finding the polynomial of the entire cipher takes a lot of time. Therefore, this paper mainly understands the complex multivariate algebraic equations of nonlinear components to analyze whether the entire cryptosystem can resist algebraic attacks. The nonlinear component of IIoTBC is S-box. In general, a 4×4 S-box is described by 21 quadratic equations with 8 input/output bit variables. IIoTBC requires 10 S-boxes for one round of encryption. After 32 rounds, there will be 320 S-boxes. In this case, IIoTBC can be described as 6720 quadratic equations of 2560 variables. According to this, IIoTBC is able to resist algebraic attacks.

7. Temperature condition monitoring application

Industrial companies can use IoT to implement predictive maintenance and condition-based monitoring. Sensors such as temperature, humidity, speed, and position play a vital role in this. This paper constructs an end-to-end temperature status acquisition platform to simulate temperature monitoring in industrial environments. At the same time, IIoTBC is deployed to the platform. Initially, the temperature transducer collects temperature data, which is transmitted to the receiver via the can bus. It worth noticed that temperature data is transmitted in plaintext. After deploying the IIoTBC, the temperature data collected by the transducer will be encrypted first. The receiver receives the temperature data in ciphertext via the can bus, and

performs the IIoTBC cipher decryption process. Then display the actual temperature data. The specific process is shown in **Fig. 13**.

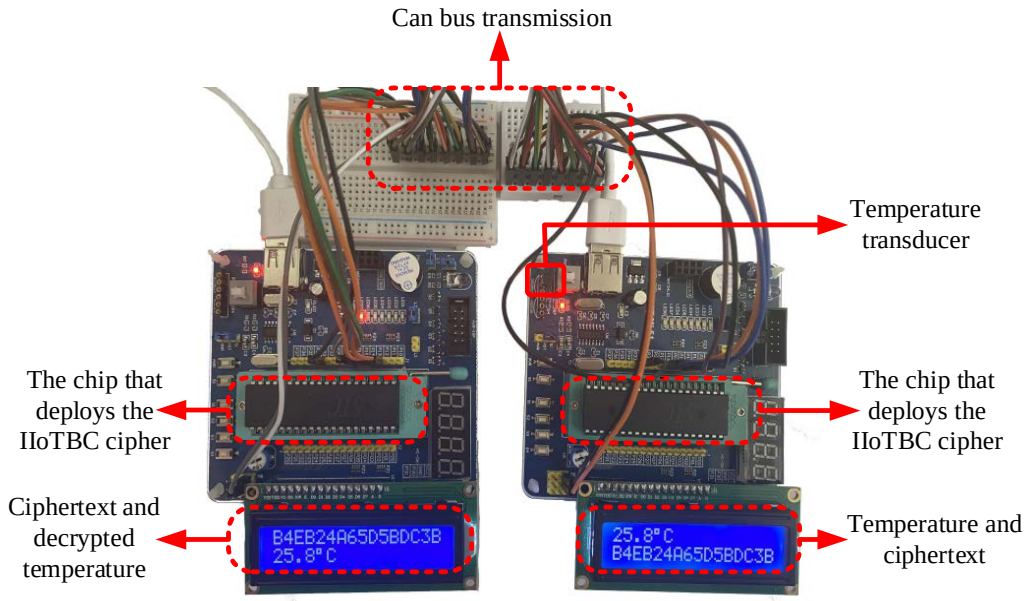

Fig. 13. Temperature status collection and encryption platform

The successful implementation of the real-time temperature state acquisition encryption platform shows that IIoTBC encryption is feasible. In general, we have deployed a lightweight encryption method at the endpoint perception layer, so that the sensing data can be transmitted and stored in ciphertext form. It provides a solution for building the first line of defense for industrial IoT systems.

8. Conclusion

This paper designs a lightweight block cipher IIoTBC for industrial IoT sensor nodes. IIoTBC adopts a variable system structure. Flexible support for MCUs of different sizes and levels of security requirements. Secondly, this paper proposes a 4×4 S-box, which strikes a good balance between hardware overhead and cryptographic properties. At the same time, this paper proposes a preprocessing method for 4×4 S-box logic gate expressions. In addition, ASIC hardware implementation, avalanche effect test, impossible differential cryptanalysis, side channel cryptanalysis, and algebraic cryptanalysis are performed on IIoTBC. As our experiments and data comparison show, IIoTBC is compact and safe in industrial IoT sensor nodes. Finally, this paper has built an end-to-end temperature acquisition platform to simulate temperature monitoring in industrial environments, and IIoTBC is successfully applied to it. At present, it is meaningful to use machine learning technology to solve the cryptanalysis problem. For example, the minimum number of active S-boxes is predicted based on deep learning [34]. A given block cipher output is classified as secure or insecure using machine learning classification and the number of active S-boxes [35]. In future work, this paper will study the security of lightweight block cipher under the background of machine learning technology.

Reference

[1] S. Vitturi, C. Zunino and T. Sauter, "Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 944-961, 2019. Article (CrossRef Link)

[2] H. Khalid, S. J. Hashim, S. M. S. Ahmad, et al., "SELAMAT: a new secure and lightweight multi-factor authentication scheme for cross-platform industrial IoT systems," *Sensors*, vol. 21, no. 4, pp. 1428, 2021. Article (CrossRef Link)

[3] H. Xie, Z. Yan, Z. Yao and M. Atiquzzaman, "Data Collection for Security Measurement in Wireless Sensor Networks: A Survey," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205-2224, 2019. Article (CrossRef Link)

[4] S. I. Popoola, R. Ande, B. Adebisi, et al., "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930-3944, 2022. Article (CrossRef Link)

[5] R. Zhao, G. Gui, Z. Xue, et al., "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9960-9972, 2022. Article (CrossRef Link)

[6] C. Xenofontos, I. Zografopoulos, C. Konstantinou, et al., "Consumer, commercial, and industrial iot (in) security: Attack taxonomy and case studies," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 199-221, 2022. Article (CrossRef Link)

[7] J. Sengupta, S. Ruj, S. D. Bit., "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, pp. 102481, 2020. Article (CrossRef Link)

[8] K. Yang, H. Wang, L. Sun, "An effective intrusion-resilient mechanism for programmable logic controllers against data tampering attacks," *Computers in Industry*, vol. 138, pp. 103613, 2022. Article (CrossRef Link)

[9] O. Gungor, T. Rosing, B. Aksanli, "STEWART: STacking Ensemble for White-Box AdversaRial Attacks Towards more resilient data-driven predictive maintenance," *Computers in Industry*, vol. 140, pp. 103660, 2022. Article (CrossRef Link)

[10] M. Jbair, B. Ahmad, C. Maple, et al., "Threat modelling for industrial cyber physical systems in the era of smart manufacturing," *Computers in Industry*, vol. 137, pp. 103611, 2022. Article (CrossRef Link)

[11] B. Liu, J. Chen, Y. Hu, "Mode division-based anomaly detection against integrity and availability attacks in industrial cyber-physical systems," *Computers in Industry*, vol. 137, pp. 103609, 2022. Article (CrossRef Link)

[12] L. Dalmasso, F. Bruguier, P. Benoit and L. Torres, "Evaluation of SPN-Based Lightweight Crypto-Ciphers," *IEEE Access*, vol. 7, pp. 10559-10567, 2019. Article (CrossRef Link)

[13] L. Sliman, T. Omrani, Z. Tari, et al., "Towards an ultra lightweight block ciphers for Internet of Things," *Journal of information security and applications*, vol. 61, pp. 102897, 2021. Article (CrossRef Link)

[14] A. Bogdanov, L. R. Knudsen, G. Leander, et al., "PRESENT: An ultra-lightweight block cipher," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 450–466, 2007. Article (CrossRef Link)

[15] S. Banik, S. K. Pandey, T. Peyrin, et al., "GIFT: a small present," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 321-345, 2017. Article (CrossRef Link)

[16] S. Banik, A. Bogdanov, T. Isobe, et al., "Midori: A block cipher for low energy," in *Proc. of International Conference on the Theory and Application of Cryptology and Information Security*, pp. 411-436, 2015. Article (CrossRef Link)

[17] K. Shibutani, T. Isobe, H. Hiwatari, et al., "Piccolo: an ultra-lightweight blockcipher," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 342-357, 2011. Article (CrossRef Link)

[18] W. Zhang, Z. Bao, D. Lin, et al., "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, vol. 58, no. 12, pp. 1-15, 2015. Article (CrossRef Link)
[19] D. Hong, J. Sung, S. Hong, et al., "HIGHT: A new block cipher suitable for low-resource device," in *Proc. of International workshop on cryptographic hardware and embedded systems*, pp. 46-59, 2006. Article (CrossRef Link)
[20] R. Beaulieu, D. Shors, J. Smith, et al., "The SIMON and SPECK lightweight block ciphers," in *Proc. of the 52nd annual design automation conference*, pp. 1-6, 2015. Article (CrossRef Link)
[21] G. Yang, B. Zhu, V. Suder, et al., "The simeck family of lightweight block ciphers," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 307-329, 2015. Article (CrossRef Link)
[22] B. Koo, D. Roh, H. Kim, et al., "CHAM: A family of lightweight block ciphers for resource-constrained devices," in *Proc. of International conference on information security and cryptology*, pp. 3-25, 2017. Article (CrossRef Link)
[23] Y. Guo, L. Li and B. Liu, "Shadow: A Lightweight Block Cipher for IoT Nodes," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 13014-13023, 2021. Article (CrossRef Link)
[24] S. Chen, Y. Fan, L. Sun, et al., "SAND: an AND-RX Feistel lightweight block cipher supporting S-box-based security evaluations," *Designs, Codes and Cryptography*, vol. 90, no. 1, pp. 155-198, 2022. Article (CrossRef Link)
[25] R. S. Jenny, R. Sudhakar, M. Karthikpriya, "Design of compact s box for resource constrained applications," *Journal of Physics: Conference Series*, vol. 1767, no. 1, pp. 012059, 2021. Article (CrossRef Link)
[26] C. Beierle, J. Jean, S. Kölbl, et al., "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Proc. of Annual International Cryptology Conference*, pp. 123-153, 2016. Article (CrossRef Link)
[27] M. Ma, D. He, N. Kumar, et al., "Certificateless searchable public key encryption scheme for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 759-767, 2018. Article (CrossRef Link)
[28] H. N. Noura, M. Noura, A. Chehab, et al., "Efficient and secure cipher scheme for multimedia contents," *Multimedia tools and applications*, vol. 78, no. 11, pp. 14837-14866, 2019. Article (CrossRef Link)
[29] B. Rashidi, "Compact and efficient structure of 8-bit S-box for lightweight cryptography" *Integration*, vol. 76, pp. 172-182, 2021. Article (CrossRef Link)
[30] L. Li, J. Liu, Y. Guo, et al., "A new S-box construction method meeting strict avalanche criterion," *Journal of Information Security and Applications*, vol. 66, pp. 103135, 2022. Article (CrossRef Link)
[31] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, et al., "A review of lightweight block ciphers," *Journal of cryptographic Engineering*, vol. 8, no. 2, pp. 141-184, 2018. Article (CrossRef Link)
[32] P. Singh, B. Acharya, R. K. Chaurasiya, "A comparative survey on lightweight block ciphers for resource constrained applications," *International Journal of High Performance Systems Architecture*, vol. 8, no. 4, pp. 250-270, 2019. Article (CrossRef Link)
[33] R. A. Ramadan, B. W. Aboshosha, K. Yadav, et al., "LBC-IOT: lightweight block cipher for iot constraint devices," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 67, no. 3, pp. 3563-3579, 2021. Article (CrossRef Link)
[34] M. F. Idris, J. S. Teh, J. L. S. Yan, et al., "A Deep Learning Approach for Active S-Box Prediction of Lightweight Generalized Feistel Block Ciphers," *IEEE Access*, vol. 9, pp. 104205-104216, 2021. Article (CrossRef Link)
[35] T. R. Lee, J. S. Teh, N. Jamil, et al., "Lightweight Block Cipher Security Evaluation Based on Machine Learning Classifiers and Active S-Boxes," *IEEE Access*, vol. 9, pp. 134052-134064, 2021. Article (CrossRef Link)

Juanli Kuang received the B.S. degree from Hengyang Normal University, Hengyang,China,in 2014 and received a master's degree from Hunan University, Changsha, China, in 2017 .She is currently working toward her Ph.D in Macau University of Science and Technology, Macau,China. Since 2017, her current research interests include embedded systems and information security.

Ying Guo received the B.S. degree from Hengyang Normal University, Hengyang, China, in 2019 and she is currently working toward a Master's degree in Hengyang Normal University, Hengyang, China. Since 2019, her current research interests include embedded systems and information security.

Lang Li received his Ph.D. and Master’s degrees in computer science from Hunan University, Changsha, China, in 2010 and 2006, respectively, and earned his B.S. degree in circuits and systems from Hunan Normal University, Changsha, China, in 1996. Since 2011, he has been working as a professor in the College of Computer Science and Technology at the Hengyang Normal University, Hengyang, China. His research interests include embedded computing and information security.