Received August 13, 2019, accepted August 26, 2019, date of publication September 10, 2019, date of current version October 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940330

Loong: A Family of Involutional Lightweight Block Cipher Based on SPN Structure

BO-TAO LIU[1,2], LANG LI[1,2], RUI-XUE WU[3], MING-MING XIE[3], AND QIU PING LI[1,2]

[1]Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University, Hengyang 421002, China
[2]College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China
[3]College of Computer Science and Technology, Guizhou University, Guiyang 550025, China

Corresponding author: Lang Li (lilang911@126.com)

This work was supported in part by the Open Fund Project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University under Grant IIPA18K02, in part by the Science and Technology Plan Project of Hunan under Grant 2016TP1020, in part by the National Natural Science Foundation of China under Grant 61572174, in part by the Double First-Class University Project of Hunan under Grant Xiangjiaotong [2018]469, in part by the Science Foundation Project of Hengyang Normal University under Grant 18D23, in part by the Hunan Province Special Funds of Central Government for Guiding Local Science and Technology Development under Grant 2018CT5001, in part by the Subject Group Construction Project of Hengyang Normal University under Grant 18XKQ02, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2019JJ60004.

ABSTRACT In past few years, as security ciphers in the Internet of Things (IoT), the research of lightweight block cipher has attracted tremendous attention in cryptography. The SPN structure has been widely used in the design of block cipher. However, the encryption and decryption processes of ciphers based on the SPN structure are different. We design a new SPN structure, which is perfect for lightweight block cipher. The new SPN structure makes that the encryption process is the same as decryption. Moreover, input and output data directions are the same for encryption and decryption processes. Thus, the same process can absolutely be shared in decryption and encryption both for software and hardware implementation. Further, we propose a family of involutional lightweight block cipher, called Loong, based on the proposed SPN structure and components. Rigorous analysis indicates that Loong is of high security against cryptanalysis, especially the differential attack and linear attack. As shown by our experiments and comparisons, Loong is compact in hardware environment and is suitable for the IoT.

INDEX TERMS Block cipher, SPN structure, involution, lightweight cryptography, Internet of Things.

I. INTRODUCTION

All along, the embedded and mobile devices of Internet of Things (IoT) have been sought-after by countless people. Embedded devices such as RFID tags and Smart cards have been widely used in the military industry, government agencies, financial institutions, medical institutions and so on. And mobile devices like smart watches and phones have already become an indispensable part in our daily life. Moreover, IoT is even considered by enthusiasts as the world's third wave of the information industry evolution after the inventions of computer and Internet [1]. However, embedded and mobile devices are resource-limited in terms of communication, computation and storage. The openness of wireless links can easily cause these devices data be stolen, and applications require the protection of sensitive data. As the lightweight block ciphers have the advantages of fast speed and little resource in the hardware and software implementation. We need adopt the lightweight block ciphers to protect sensitive data [2].

In recent years, lightweight block ciphers have developed rapidly, and a variety of lightweight block ciphers have been designed. There are many well designed lightweight block ciphers such as LILLIPUT [3], QTL [4], PRESENT [5], LBlock [6], GIFT [7], SFN [8], Midori [9], SKINNY and MANTIS [10], PRINCE [11], Piccolo [12], Rectangle [13], TWINE [14], SIMON and SPECK [15], LED [16], KLEIN [17], SPARX [18], ITUbee [19], SIMECK [20], DESL [21], PRIDE [22], GOST revisited [23], and so on. Since the appearance of the concept of lightweight block cipher, it is attracting tremendous attention from the academia, industry and government [24].

There are mainly two types of foundation structures, the Feistel-type networks structure and the Substitution-Permutation Networks (SPN) structure, for these lightweight block ciphers. The Feistel-type networks structure is

The associate editor coordinating the review of this manuscript and approving it for publication was Qing Yang.

widely known from the classical Data Encryption Standard (DES) [25]. The encryption and decryption processes of cryptographic algorithm based on the Feistel-tpye networks structure are the same. This is an extremely good superiority of the Feistel-type networks structure. However, the round function of the Feistel-type networks structure has weak diffusion capability, and only the half data is converted into the round function at each iterated operation. In order to ensure security, these algorithms require a lot of iterations, while their efficiency is reduced. In addition, when a cryptographic algorithm based on the Feistel-type networks structure carries out encryption and decryption, the input data direction of the decryption is the output data direction of the encryption.

The other one is the SPN structure which is the result of the seminal work of Shannon [26]. And the best example of the SPN structure is the Advanced Encryption Standard (AES) [27] which has a great influence over the design of block ciphers. Affected by AES, 13 out of the 17 New European Schemes for Signatures, Integrity, and Encryption (NESSIE) candidate algorithms have adopted the SPN structure [28]. AES has been researched by cryptographic community for more than 20 years and is still safe. The SPN structure has great confusion and diffusion effects. But the encryption and decryption processes of a SPN structure based cryptographic algorithm are different. This limitation makes this kind of algorithms that: (1) the inverse cipher is less suitable to be implemented on a smart card than the cipher itself: it needs more codes and cycles; (2) in software, the cipher and its inverse make use of different codes and/or tables; (3) in hardware, the inverse cipher can only partially re-use the circuitry that implements the cipher. Therefore, it is important to design a similar encryption and decryption process for the lightweight block cipher. The cryptographic community has been studying the encryption and decryption processes of modern lightweight SPN algorithms being identical.

PRESENT, a lightweight block cipher proposed in 2007, has been one of the ISO-29192. It applies SPN and its round function is simple and efficient. But encryption of this cipher is different with its decryption. I-PRESENT^{TM} [29] is an involutional block cipher based on PRESENT. PRINCE is a low-latency involutional block cipher proposed in 2012. Its PRINCE$_{core}$ is a typical SPN structure, and its design has a significant advantage that the overhead for decryption on top of encryption is negligible. However, PRINCE is an unrolled architecture, which needs lots of Gate Equivalents (GE), and GE is logic area in ASIC. Midori is a very competitive lightweight block cipher proposed in 2015. It is also SPN structure based. To reduce hardware and software resources, this cipher optimizes the S-box and the diffusion Matrix. Moreover, the S-box and the diffusion Matrix are involutive. Then, in software and hardware implementations, the inverse cipher of Midori can partially re-use the codes and circuitries. But encryption of this cipher is also different with its decryption. In summary, it becomes an important design aspect that the encryption and decryption processes of modern lightweight SPN algorithms being identical.

A. OUR CONTRIBUTIONS

We propose an involutional lightweight block cipher, called Loong, that is based on a new SPN structure, which makes the encryption and decryption process of a cipher be the same. It is important that the input and output data directions are consistent in encryption and decryption process. Loong has 64-bit block with 64-bit, 80-bit, and 128-bit key blocks, respectively. The round transformation components are AddRoundKey, SubCells, MixRows, and MixColumns. These components are involutional in the round function. Therefore, the encryption and decryption process of Loong are the same. The inverse cipher of Loong can absolutely re-use the codes and the circuitries in software and hardware implementation, respectively. And the round transformation components are lightweight and efficient. So the structure and components of Loong are extremely suited for lightweight block ciphers.

The round function process of Loong is expressed as: SubCells $\rightarrow$ MixRows $\rightarrow$ MixColumns $\rightarrow$ SubCells $\rightarrow$ AddRoundKey. The round function has two SubCells operations, this makes Loong more secure, and the number of active S-boxes is far more than those of PRESENT, LED, Midori and GIFT in the round function. The round function has the MixRows and MixColumns operations, and the diffusion matrix of the MixRows and MixColumns is Maximal Distance Separable (MDS). Then the branch numbers of the matrix of the MDS is 5. Respectively, the branch numbers of the MixRows and the MixColumns are 5. Therefore, Loong has better diffusion effect than AES and Midori. Due to good confusion and diffusion in the round function, the round function can sufficiently resist differential and linear cryptanalysis. Meanwhile, in order to reduce hardware and software resources, the SubCells operation adopts a small-delay and lightweight 4-bit S-box, and the MixColumns can be replaced by MixRows. Thus, the Loong structure is highly efficient and secure.

Loong is highly symmetrical lightweight block cipher. Although the round function includes AddRoundKey, SubCells, MixRows and MixColumns, we simplified the round function in a round-based implementation. Thus, double use of resources can be avoided in the round function and the area cost requires 11.33625 GE per bit of internal state, which is lower than in the cases of PRESENT and Piccolo and so on.

B. STRUCTURE OF THE PAPER

In Section II, we outline the algorithm specification of Loong. In Section III, we explain the design rationale of Loong. In Section IV, we present the security analysis of Loong. The performance evaluation of hardware and software implementations is presented in Section V. And Section VI concludes the paper.

II. SPECIFICATION OF LOONG

Loong is based on a SPN structure and has 64-bit block with 64-bit, 80-bit, and 128-bit key block and corresponding

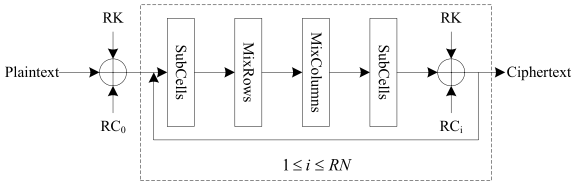


FIGURE 1. The encryption process of Loong.

Round Number (RN) of 16, 20 and 32 rounds, respectively. According to the length of the three keys, algorithms are denoted as Loong-64, Loong-80 and Loong-128.

A. ENCRYPTION ALGORITHM

The encryption and decryption processes of Loong are the same, but the only difference is using round constants in reverse order. The round function of Loong includes AddRoundKey, SubCells, MixRows and MixColumns. The Loong encryption process is shown in Fig. 1.

The Loong encryption process can be expressed as ENC_{RN}. In ENC_{RN}, a plaintext which could be divided into several plaintext blocks with 64-bit and a primary key are regarded as input. The encryption process of Loong-64, Loong-80 and Loong-128 is denoted by ENC_{16}, ENC_{20} and ENC_{32}, respectively, as described in (1):

$$ENC_{RN} : \left\{ \begin{array}{c} \{0,1\}^{64} \times \{0,1\}^{K_{RN}} \rightarrow \{0,1\}^{64} \\ (plaintext, key) \rightarrow ciphertext \end{array} \right\} \quad (1)$$

where, $RN = \{16, 20, 32\}$, $K_{16} = 64$, $K_{20} = 80$, $K_{32} = 128$.

The Algorithm 1 illustrates ENC_{RN} in details, where Round Key (RK) is a 64-bit , which is described in Key schedule.

Algorithm 1 Loong Encryption Routine

ENC_{RN}
Input : Plaintext , RK , RC;
Output : Ciphertext;
1: $state \leftarrow$ Plaintext;
2: AddRoundKey($state$, RK, RC);
3: for i=1 to RN do
4: SubCells($state$);
5: MixRows($state$);
6: MixColumns($state$);
7: SubCells($state$);
8: AddRoundKey($state$, RK, RC);
9: endfor
10: Ciphertext $\leftarrow$ $state$;
11: **Return** Ciphertext;

The round function process of Loong is expressed as: SubCells $\rightarrow$ MixRows $\rightarrow$ MixColumns $\rightarrow$ SubCells $\rightarrow$ AddRoundKey (See Fig. 2).

AddRoundkey: The data of 64-bit $state$ performs XOR operation with the data of 64-bit RK and the round constants RC, then $\mathbb{Z}_2^{64} \rightarrow \mathbb{Z}_2^{64}$:

The 64-bit key form:

$$state \leftarrow state \oplus RK \oplus RC_i \quad (2)$$

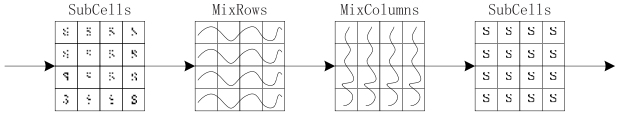


FIGURE 2. An overview of a round of Loong.

TABLE 1. 4-Bit bijective S-box in hexadecimal form.

X	0	1	2	3	4	5	6	7
S[x]	C	A	D	3	E	B	F	7
X	8	9	A	B	C	D	E	F
S[x]	9	8	1	5	0	2	4	6

The 80-bit and 128-bit key forms:

$$state \leftarrow state \oplus RK_{imod2} \oplus RC_i, \quad (0 \le i \le RN) \quad (3)$$

SubCells: The S-box with 4 bit is involutive. The elements of S-box are shown in Table 1. The data of 64-bit $state$ need to be divided into 16 4-bit nibbles $state_j (0 \le j \le 15)$,then SubCells operation is as follows:

$$\mathbb{Z}_{2^4}^{16} \rightarrow \mathbb{Z}_{2^4}^{16} : state_{j[0:3]} \leftarrow S - box(state_{j[0:3]}) \quad (4)$$

In order to ensure better diffusion performance after round function, we design the matrix multiplication operation. We use two different involutional diffusion matrices M and M' whose matrix multiplication is an identity matrix as follows:

$$\begin{aligned} &M \times M' \\ &\rightarrow \begin{pmatrix} 1 & 4 & 9 & 13 \\ 4 & 1 & 13 & 9 \\ 9 & 13 & 1 & 4 \\ 13 & 9 & 4 & 1 \end{pmatrix} \times \begin{pmatrix} 13 & 9 & 4 & 1 \\ 9 & 13 & 1 & 4 \\ 4 & 1 & 13 & 9 \\ 1 & 4 & 9 & 13 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (5)$$

MixRows: The matrix multiplication operation is that the matrix M multiplies with the data of 64-bit $state$ in finite field $GF(2^4)$. The irreducible polynomial is called $x^4 + x + 1$ in $GF(2^4)$. The matrix multiplication operation updates the data of 64-bit $state$ in the MixRows as follows.

$$\begin{aligned} &\mathbb{Z}_{2^{16}}^4 \rightarrow \mathbb{Z}_{2^{16}}^4 : \\ &state \leftarrow \begin{pmatrix} state_0 & state_1 & state_2 & state_3 \\ state_4 & state_5 & state_6 & state_7 \\ state_8 & state_9 & state_{10} & state_{11} \\ state_{12} & state_{13} & state_{14} & state_{15} \end{pmatrix} \\ &\times \begin{pmatrix} 1 & 4 & 9 & 13 \\ 4 & 1 & 13 & 9 \\ 9 & 13 & 1 & 4 \\ 13 & 9 & 4 & 1 \end{pmatrix} \end{aligned} \quad (6)$$

MixColumns: The MixColumns is also a matrix multiplication operation, which is similar to MixRows. It is the $state$

multiplied by the matrix M', and the matrix multiplication operation updates the data of 64-bit *state* as follow.

$$\mathbb{Z}_{2^{16}}^4 \rightarrow \mathbb{Z}_{2^{16}}^4 :$$
$$state \leftarrow \begin{pmatrix} 13 & 9 & 4 & 1 \\ 9 & 13 & 1 & 4 \\ 4 & 1 & 13 & 9 \\ 1 & 4 & 9 & 13 \end{pmatrix} \times \begin{pmatrix} state_0 & state_1 & state_2 & state_3 \\ state_4 & state_5 & state_6 & state_7 \\ state_8 & state_9 & state_{10} & state_{11} \\ state_{12} & state_{13} & state_{14} & state_{15} \end{pmatrix} \quad (7)$$

B. KEY SCHEDULE AND ROUND CONSTANTS

Key schedule: Loong has 64-bit, 80-bit and 128-bit key block respectively. The round key RK is the primary key K. The 64-bit K is denoted as $k_0, k_1, \ldots k_{15}$, the 80-bit K is denoted as $k_0, k_1, \ldots k_{19}$, and the 128-bit K is denoted as $k_0, k_1, \ldots k_{31}$.

The 64-bit key is arranged into a round key square array.

$$RK \leftarrow \begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix}$$

The 80-bit key is arranged into two round key square arrays.

$$RK_0 \leftarrow \begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix}$$
$$RK_1 \leftarrow \begin{bmatrix} k_{16} & k_{17} & k_{18} & k_{19} \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \end{bmatrix}$$

The 128-bit key is arranged into two round key square arrays.

$$RK_0 \leftarrow \begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix}$$
$$RK_1 \leftarrow \begin{bmatrix} k_{16} & k_{17} & k_{18} & k_{19} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{24} & k_{25} & k_{26} & k_{27} \\ k_{28} & k_{29} & k_{30} & k_{31} \end{bmatrix}$$

Round Constants: The round constants are generated by the 6-bit affine LFSR. The 6-bit round constants are denoted as $(rc_5, rc_4, rc_3, rc_2, rc_1)$. Its update function is defined as:

$$(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0) \leftarrow (rc_4, rc_3, rc_2, rc_1, rc_0, rc_5 \oplus rc_4 \oplus 1) \quad (8)$$

The 6-bit round constants are initialized to zero, and updated before being used in a given round. Table 2 shows the list of $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$ encoded to byte values for each round, where rc_0 is the least significant bit.

TABLE 2. The elements of round constants in hexadecimal form.

Rounds	Constants										
1-11	01	03	07	0F	1F	3E	3D	3B	37	2F	1E
12-22	3C	39	33	27	0E	1D	3A	35	2B	16	2C
23-33	18	30	21	02	05	0B	17	2E	1c	38	31

Loong has three different key blocks with three 17, 21 and 33 different round constants respectively. The adding of round constants is arranged into an array as follows:

$$\begin{bmatrix} 0 & 0 & 0 & (rc_5||rc_4||rc_3) \\ 0 & 0 & 1 & (rc_2||rc_1||rc_0) \\ 0 & 0 & 2 & (rc_5||rc_4||rc_3) \\ 0 & 0 & 4 & (rc_2||rc_1||rc_0) \end{bmatrix}$$

C. DECRYPTION ALGORITHM

The cipher of Loong and its inverse is the same algorithm. And the input and output data directions are the same for encryption and decryption processes. Loong decryption can be accomplished by reading round constants in reverse order. So this Loong decryption is highly simple and efficient.

III. DESIGN RATIONALE

A. ALGORITHM STRUCTURE

It is well known that AES is a SPN structure based algorithm. The SPN structure can provide better confusion and diffusion than the Feistel-type networks structure for round function. The SPN-based algorithms are more efficient and reliable. However, the encryption and decryption process of SPN-based algorithms are different. To deal with this problem, we propose a new SPN structure. This new SPN structure makes the encryption and decryption process of a cipher be the same. Thus, the Loong utilizes a new SPN structure. The round transformation components of Loong are SubCells (SC), MixRows (MR), MixColumns (MC) and AddRoundKey (ARK). These components are involutional. The encryption round function of Loong is expressed as:

$$ENC_{RN}\left[RC^0, \ldots, RC^{RN}\right] = ARK\left(RK, RC^0\right) \circ \left(\bigcirc_{r=1}^{RN} SC \circ MR \circ MC \circ SC \circ ARK\left(RK, RC^{RN}\right)\right) \quad (9)$$

Meanwhile, the decryption round function of Loong is expressed as:

$$DEC_{RN}\left[RC^{RN}, \ldots, RC^0\right] = ARK\left(RK, RC^{RN}\right) \circ \left(\bigcirc_{r=RN-1}^{0} SC \circ MR \circ MC \circ SC \circ ARK\left(RK, RC^0\right)\right) \quad (10)$$

So the encryption process of Loong is the same as the decryption. But the only difference between encryption and decryption is using round constants in reverse order. We have to prove that the encryption process is the same as the decryption process in Appendix A.

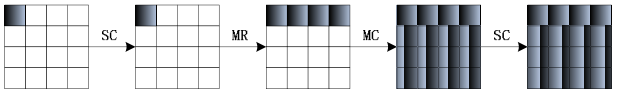


FIGURE 3. The diffusion effect of the round function of Loong.

However, there are already some algorithms in the field of cryptography, where the encryption and decryption processes of these algorithms based on the Feistel-type networks or the SPN structure are the same. But the input data direction of the decryption process is the output data direction of the encryption process. We propose the new SPN structure which makes the encryption and decryption processes of the cryptographic algorithm to be the same. And the input and output data directions and the using order of key are the same for encryption and decryption processes. Thus, the decryption can absolutely re-use the encryption in software and hardware implementation. The new SPN structure is well suited for lightweight block cipher.

Considering most lightweight block ciphers with lower security coefficients, we use twice SubCells operations in the round function. The design of this kind of round function has appeared in some lightweight block ciphers, such as Piccolo and QTL. Piccolo and QTL are Feistel-type networks structure. So, the number of active S-boxes is more than other lightweight block ciphers based on SPN structure in round function. The round function has two distinct linear operations which are MixRows and MixColumns. The MixRows and MixColumns are used between two SubCells. The diffusion matrices M and M' are 4×4 MDS, and the branch numbers of the MixRows and the MixColumns are 5, respectively. Therefore, Loong has better diffusion effect than AES and Midori. Fig. 3 shows the diffusion effect of the round function. Considering differential and linear cryptanalysis, Loong 16-round differential probability and linear probability are 2^{-256} respectively. Meanwhile, in order to reduce hardware and software resources, the SubCells operation adopts a small-delay and lightweight 4-bit S-box and the MixColumns can be replaced by MixRows. Thus, the Loong structure is highly efficient and secure.

B. SUBCELLS

Loong is an involutive lightweight block cipher, its S-box has involution, low latency, and small gate area. And its S-box is modified based on the Sb0 of Midori. As we known, compared with other 4-bit S-boxes, the greatest advantage of Sb0 is a small-delay and lightweight 4-bit S-box, which only needs 0.24 ns and 13.3 GE with using NanGate 45 nm open cell library [30]. But the fixed-point number of Sb0 is too large. The fixed-point number of S-boxes is defined as δ, $\delta = \#\{x|S-box(x) = x, x \in \mathbb{F}_{2^n}\}$, and $\delta = \{0 \leq \delta \leq 2^n - 1\}$. Then, δ of Sb0 is 4. Considering the fixed-point number of Sb0, we have modified Sb0. After modified we find the δ of S-box in this paper is 2 and S-box still has involution, low latency, and small gate area. The study in [9] presents a metric depth to estimate path delay and gate area of S-boxes. We calculate the depth of S-box, where outputs and inputs are defined as $\{a', b', c', d'\}$ and $\{a, b, c, d\}$.

$$\begin{aligned} a' &= \left(\bar{c}\ NAND\ (a\ NAND\ b)\right)\ NAND\ (a\ OR\ d) \\ b' &= \left((a\ NOR\ d)\ NAND\ (b\ AND\ c)\right) \\ &\quad \times NAND\left((a\ AND\ c)\ NAND\ d\right) \\ c' &= (b\ NAND\ d)\ NAND\left((b\ NOR\ d)\ OR\ a\right) \\ d' &= \left((a\ NAND\ b)\right. \\ &\quad \times NAND\left((b\ NOR\ c)\ NAND\ d\right)\Big) \\ &\quad \times NOR\left((a\ OR\ c)\ NOR\ d\right) \end{aligned} \tag{11}$$

Compared with Sb0, the d' is different, and the remaining three are same. The depth of d' is estimated as $4 = (1 + 1 + 1 + 1)$. The depth of d' is 3.5 in Sb0. For Sb0, in this paper the depth of S-box only adds to 0.5, but the fixed-point number reduces 2. Moreover, the algebraic degree and avalanche effect of S-box are the same as Sb0, and this S-box meets the performance that the maximal differential probability of is 2^{-2}, and the maximal absolute bias of a linear approximation is 2^{-2}.

C. MIXROWS, MIXCOLUMNS

The MixRows and MixColumns are linear components. In order to ensure the high security of Loong, we need to use a strong linear confusion matrix in the MixRows and MixColumns. The strong linear diffusion matrices are MDS matrices. Meanwhile, Loong is an involutive lightweight block cipher, the MDS matrix is involutive and can be quite light to implement. We apply the MDS matrix M which comes from the work [31]. And the involution MDS matrix M' forms by columns of M shifting. The M is used in the MixRows and the M' is used in the MixColumns. The MixRows can be expressed for the first 4-bit element of the first row of matrix multiplication $state_0 \cdot$ 0x1 $\oplus state_1 \cdot$ 0x4 $\oplus state_2 \cdot$ 0x9 $\oplus state_3 \cdot$ 0xd. And The MixColumns can be expressed for the first 4-bit element of the first column of matrix multiplication 0xd $\cdot state_0 \oplus$ 0x9 $\cdot state_4 \oplus$ 0x4 $\cdot state_8 \oplus$ 0x1 $\cdot state_{12}$. It takes 18 XORs to implement one row or column of the matrix. The M is one of the matrices belonging to fewer resources. Therefore, the implementation of MixRows and MixColumns does not require too many resources.

D. ROUND CONSTANTS

The round constants are generated by the 6-bit affine LFSR which is used in LED, SKINNY and GIFT and they are different with each other. The different round constant performs XOR operation with the internal state in the AddRoundkey of Loong. Moreover, the fixed constants 1, 2, 4 only require a bit XOR operation. Thus, the goals of round constant differentiate the rounds and the columns. The adding of round constants is arranged into two columns of an array, and the round constant is spread to a whole array after the MixRows. So the round constants make the rounds and columns different. The design of round constants is not only to save resources, but also to resist self-similarity attacks.

TABLE 3. Result of diffusion effect.

Ciphers	Plaintexts	Results(first round)
Loong-64	0000000000000001	2EC9225BFF9DF970
PRESENT-80	0000000000000001	FFFFFFFF00000001
KLEIN-64	0000000000000001	7777767772060374
LBlock-80	0000000000000001	BED21F7400000000

E. DIFFUSION EFFECT

This paper analyzes the security of the Loong on diffusion effect. In the subsection III-A, we describe that the round function of Loong has strongly diffusion effect. Now we make the experiment of avalanche effect to prove above conclusion, and in the experiment we compare the Loong-64 with other lightweight block ciphers about diffusion effect under the $key = 0$. Table 3 shows the comparison results of these block ciphers after one round of encryption. We found that the Loong-64 has good diffusion effect.

IV. SECURITY ANALYSIS

A. DIFFERENTIAL AND LINEAR ATTACK

As we all know, Differential Cryptanalysis (DC) and Linear Cryptanalysis (LC) are powerful tools to attack block ciphers, and it is necessary to consider the block ciphers to resist these two attacks. The work in [32] pointed out that if the maximum differential characteristic probability (maximum linear approximation probability) is small enough, the cipher can resist differential cryptanalysis (linear cryptanalysis), and the differential characteristic probability (linear approximation probability) upper bound often can be calculated by the number of active S-boxes, thus, estimation of block cipher against differential cryptanalysis (linear cryptanalysis) ability is the key to calculate the number of active S-boxes.

We construct a mixed integer linear programming (MILP) to calculate the lower bounds for the minimum number of active S-box. The round function process of Loong is expressed as: SubCells → MixRows → MixColumns → SubCells. The update of the first Loong round function shows as follows. Each variable x is regarded as a nibble of Loong state. If the difference is non-zero, the variable x is 1, otherwise, x is 0. All variables corresponding to the inputs of the SubCells operations are summed in the objective function, so this corresponds to the number of active S-boxes [33]. The MixRows and MixColumns are linear functions and have differentials. The linear branch number of the MixRows and MixColumns is 5. We use linear equations to describe the input and output difference and linear mask vectors MixRows and MixColumns operations. The MixRows can be constrained by the following linear equations:

$$\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \xrightarrow{SC} \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}$$

$$\xrightarrow{MR} \begin{bmatrix} x_{16} & x_{17} & x_{18} & x_{19} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{24} & x_{25} & x_{26} & x_{27} \\ x_{28} & x_{29} & x_{30} & x_{31} \end{bmatrix}$$

$$\xrightarrow{MC} \begin{bmatrix} x_{32} & x_{36} & x_{40} & x_{44} \\ x_{33} & x_{37} & x_{41} & x_{45} \\ x_{34} & x_{38} & x_{42} & x_{46} \\ x_{35} & x_{39} & x_{43} & x_{47} \end{bmatrix}$$

$$\xrightarrow{SC} \begin{bmatrix} x_{32} & x_{36} & x_{40} & x_{44} \\ x_{33} & x_{37} & x_{41} & x_{45} \\ x_{34} & x_{38} & x_{42} & x_{46} \\ x_{35} & x_{39} & x_{43} & x_{47} \end{bmatrix}$$

$$\begin{cases} x_0+x_1+x_2+x_3+x_{16}+x_{20}+x_{24}+x_{28}-5d_0 \geq 0 \\ d_0 - x_i \geq 0 \ (i = 0, 1, 2, 3, 16, 20, 24, 28) \end{cases}$$

$$\begin{cases} x_4+x_5+x_6+x_7+x_{17}+x_{21}+x_{25}+x_{29}-5d_1 \geq 0 \\ d_1 - x_i \geq 0 \ (i = 4, 5, 6, 7, 17, 21, 25, 29) \end{cases}$$

$$\begin{cases} x_8+x_9+x_{10}+x_{11}+x_{18}+x_{22}+x_{26}+x_{30}-5d_2 \geq 0 \\ d_2 - x_i \geq 0 \ (i = 8, 9, 10, 11, 18, 22, 26, 30) \end{cases}$$

$$\begin{cases} x_{12}+x_{13}+x_{14}+x_{15}+x_{19}+x_{23}+x_{27}+x_{31}-5d_3 \geq 0 \\ d_3 - x_i \geq 0 \ (i = 12, 13, 14, 15, 19, 23, 27, 31) \end{cases}$$

The MixColumns can be constrained by the following linear equations:

$$\begin{cases} x_{16}+x_{17}+x_{18}+x_{19}+x_{32}+x_{33}+x_{34}+x_{35}-5d_4 \geq 0 \\ d_4 - x_i \geq 0 \ (i = 16, 17, 18, 19, 32, 33, 34, 35) \end{cases}$$

$$\begin{cases} x_{20}+x_{21}+x_{22}+x_{23}+x_{36}+x_{37}+x_{38}+x_{39}-5d_5 \geq 0 \\ d_5 - x_i \geq 0 \ (i = 20, 21, 22, 23, 36, 37, 38, 39) \end{cases}$$

$$\begin{cases} x_{24}+x_{25}+x_{26}+x_{27}+x_{40}+x_{41}+x_{42}+x_{43}-5d_6 \geq 0 \\ d_6 - x_i \geq 0 \ (i = 24, 25, 26, 27, 40, 41, 42, 43) \end{cases}$$

$$\begin{cases} x_{28}+x_{29}+x_{30}+x_{31}+x_{44}+x_{45}+x_{46}+x_{47}-5d_7 \geq 0 \\ d_7 - x_i \geq 0 \ (i = 28, 29, 30, 31, 44, 45, 46, 47) \end{cases}$$

Table 4 shows minimum number of differential or linear active S-boxes for n rounds of Loong, AES, PRESENT, GIFT and RECTANGLE. As shown in table 4, Loong has an average of 32 active S-boxes every 4 rounds, but AES has only an average of 25 active S-boxes every 4 rounds. Meanwhile, Loong is more active S-boxes than GIFT, PRESENT and RECTANGLE in 9-round. The S-box of Loong meet the performance: the maximal probability of a differential is 2^{-2} and the maximal absolute bias of a linear approximation is 2^{-2}. For Loong, its 9-round differential probability is 2^{-144} and its 9-round bias of linear probability is 2^{-73}. Loong-64 has 16 rounds, and its differential probability is 2^{-256} and its bias of linear probability is 2^{-129}. Loong-80 has 20 rounds, and its differential probability is 2^{-320} and its bias of linear probability is 2^{-161}. And Loong-128 has 32 rounds, and its differential probability is 2^{-512} and its bias of linear probability is 2^{-257}. Therefore, Loong has high security and we believe Loong is enough to resist against differential and linear attacks.

TABLE 4. Lower bounds for the number of active S-boxes.

Ciphers	DC/LC	Rounds									Ref.
		1	2	3	4	5	6	7	8	9	
Loong-64	DC	8	16	24	32	40	48	56	64	**72**	This work
	LC	8	16	24	32	40	48	56	64	**72**	
AES-128	DC	1	5	9	25	26	30	34	50	51	[33]
	LC	1	5	9	25	26	30	34	50	51	
PRESENT-80	DC	1	2	4	6	10	12	14	16	18	[7]
	LC	1	2	3	4	5	6	7	8	9	
GIFT-64	DC	1	2	3	5	7	10	13	16	18	[7]
	LC	1	2	3	5	7	9	12	15	18	
RECTANGLE	DC	1	2	3	4	6	8	11	13	14	[7]
	LC	1	2	3	4	6	8	10	12	14	

B. RELATED-KEY ATTACK

Related-Key Attack mainly uses some properties of the encryption and KeyExpantion to obtain the key information. Among the related-key attacks, the related-key differential attack is the most widely used. Thus, we need to analyze the probability of related-key differential characteristics. In the subsection IV-A, we computed the number of active S-boxes to evaluate the resistance differential attack under the single-key setting. The key schedule of Loong is very efficient like the schedule of LED.

64-bit key version. The 64-bit primary key is round key RK. When the differences are inserted in the RK input, the RK will be active in Loong-64. The work [16] presents a method for obtaining related-key differential path, we are ensured that at least one round is active in every two rounds by this method. So Loong-64 has at least $(RN/2) \times 8$ active S-boxes in related-key differential path, and its 8-round differential probability is 2^{-64}. Overall, the differential probability of Loong-64 is 2^{-128} and it is secure against related-key attacks.

80-bit and 128-bit key version. The 80-bit and 128-bit primary keys are round keys RK_0 and RK_1. When the differences are inserted in the RK_0 or RK_1 input, the RK_0 or RK_1 will be active in Loong-80 or Loong-128. We are ensured that at least two round are active in every four rounds. So Loong-80 and Loong-128 have at least $(RN/4) \times 2 \times 8$ active S-boxes in related-key differential path. For Loong-80, its 10-round differential probability is 2^{-80}. And for Loong-128, its 16-round differential probability is 2^{-128}. Overall, the differential probability of Loong-80 and Loong-128 is 2^{-160} and 2^{-256} respectively, they are doubtful to resist the related-key attacks.

C. ALGEBRAIC ATTACK

Algebraic attack is also an important cryptanalysis method developed in recent years, which have been used against block ciphers and stream ciphers. When we analyze whether a block cipher resist algebraic attack, we mainly understand the complex multivariate algebraic equations of the non-linear components. The non-linear component of Loong is SubCell. The round function of Loong has 32 S-boxes. A 4×4 S-box can be described by 21 quadratic equations of 8 input/output-bit variables over [34]. When the Loong makes a round of encryption operation, the Loong has 32 S-boxes in round function. Therefore, the Loong-64 has 512 S-boxes after 16 rounds of encryption, the Loong-80 has 640 S-boxes after 20 rounds of encryption, the Loong-128 has 1024 S-boxes after 32 rounds of encryption. Meanwhile, the Loong-64 can be described as 10752 quadratic equations of 4096 variables, the Loong-80 can be described as 13440 quadratic equations of 5120 variables and the Loong-128 can be describe as 21504 quadratic equations of 8192 variables. For these experimental results, we need to compare Loong with some of the typical lightweight block cipher, such as PRESENT, MIBS and KLEIN. And the result of algebraic comparison information is shown in Table 5. According to these data, the Loong achieve a good result. So, the Loong is able to resist algebraic attack.

TABLE 5. The algebraic comparison information in different ciphers.

Ciphers	Rounds	S-boxes	Quadratic equations	Variables
Loong-64	16	512	10752	4096
Loong-80	20	**640**	**13440**	**5120**
Loong-128	32	1024	21504	8192
KLEIN-64	12	240	5040	1920
MIBS-80	32	320	6720	2560
PRESENT-80	31	527	11067	4216

D. XSL ATTACK

The eXtended Sparse Linearization (XSL) attack is a method of cryptanalysis for block ciphers published in 2002 [34]. This attack is the same as algebraic attack to derive a system of quadratic simultaneous equations for block ciphers. WF is defined as the complexity of the XSL attack in [34].

$$WF = \Gamma^{\omega} \cdot (Block\ size)^{\omega\lceil\frac{t-r}{s}\rceil} \cdot (Number\ of\ rounds)^{2\omega\lceil\frac{t-r}{s}\rceil} \quad (12)$$

where $\Gamma = 2^{8.0}$, $\omega = 2.376$ and $\lceil\frac{t-r}{s}\rceil$ is 4 for Loong. So, the WF of Loong-64 is about $2^{152.064}$, the WF of Loong-80 is about $2^{158.223}$, and the WF of Loong-128 is about $2^{171.072}$. WF is large enough in Loong, thus, Loong is able to resist XSL attack.

E. MEET-IN-THE-MIDDLE ATTACKS

Meet-in-the middle attack is an analytical method of balancing time and memory and has been widely used in block ciphers now. The round function of Loong has a faster diffusion effect, and after two rounds of operation, 64-bit encrypted data can be fully diffused. So, the number of rounds used for the partial matching (PM) [35] is upper bounded by 3 $((2-1)+(2-1)+1=3)$. Part of the intermediate state of the block cipher can be computed from the plaintext in the backward direction or from the ciphertext in the forward direction. In addition, the above computation process cannot obtain the primary key. By combining with the splice-and-cut technique [36], in the worst case, the meet-in-the-middle attack can analyze up to 7 $(2+2+3=7)$ rounds. Therefore, Loong is able to resist the meet-in-the-middle attack.

V. PERFORMANCE EVALUATION

A. HARDWARE IMPLEMENTATION

We have a hardware experiment on the Loong. At first, we used the Verilog-HDL to program the Loong. Secondly, we utilized the ModelSim SE PLUS 6.1f Evaluation to simulate the Loong. Finally, Loong has been synthesized using Synopsys Design Compiler version A-2007.12-SP1 with UMCL18G212T3 [37] standard cell library, which is based on the UMC L180 0.18 μ m 1P6M logic process with a typical voltage of 1.8V and a clock frequency of 100KHz. The bit operation of combinational logic requires area cost: an OR/AND gate is equivalent to 1.33 GE, a NOR/NAND gate is equivalent to 1 GE, a NOT gate is equivalent to 0.67 GE, and a XOR/XNOR gate costs 2.67 GE.

1) THEORETICAL PERFORMANCE

Loong is a highly symmetrical lightweight block cipher, its round function includes AddRoundKey, SubCells, MixRows and MixColumns. In order to save resources, we simplified the round function of Loong as depicted in Fig. 4, where we only implement AddRoundKey, SubCells, MixRows and P in a round-based implementation. P is a permutation operation as depicted in Fig. 5 and does not require resources in hardware implementation. The P permutation operation is as follows:

$$\begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix} \xrightarrow{P} \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}$$

The P permutation is a 4-bit block shift operation to implement matrix transposition. The MixRows is by the M multiplied the *state* ($state \times M$), and the MixColumns is the *state* multiplied by the matrix M' ($M' \times state$). The result of $M \times state$ is not equal to the result of $state \times M$, but the result of $M \times state$ equals the result of $\left(state^T \times M\right)^T$. Because the function of P permutation operation is a matrix transposition, and the columns of M make shifting operation, the result of $M' \times state$ equals the result of $P\left(P\left(state\right) \times M'\right)$. The M' can be acted as the M by a permutation operation.

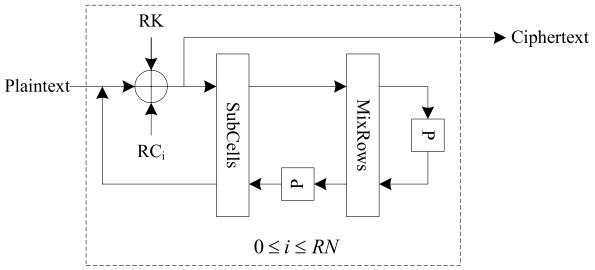


FIGURE 4. The simplified round structure of Loong.

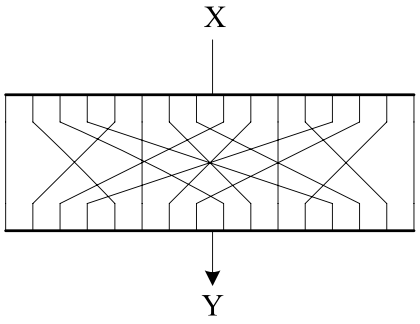


FIGURE 5. The P permutation operation.

So a permutation operation needs to be added after the second P permutation in the round function. The MixColumns can be replaced by the MixRows. The simplified round function can save resources of SubCells and MixColumns in hardware implementation. This way makes an encrypted clock cycle for Loong longer, but compared with other lightweight block ciphers, Loong has fewer iterative rounds.

Loong-64. Firstly, the 64-bit internal state XOR the 64-bit round key in round function, it requires 64 XOR gates, thus costing 1 XOR per bit of internal state. Then, The SubCells uses 16 same 4-bit S-boxes, and each S-box requires 11 NAND, 5 NOR, 2 AND and 3 OR gates (11 NAND and 5 NOR gates are equivalent to 16 NAND gates, 2 AND and 3 OR gates are equivalent to 5 AND gates). This eventually amounts to 4 NAND and 1.25 AND per bit of internal state. The MixRows layer applies a diffusion matrix which requires 72 XOR gates, where 24 XOR gates use for the matrix coefficients and 48 XOR gates use for the elements sums, and thus it updates 64-bit leading to 1.125 XOR per bit of internal state. In total, the round function of Loong-64 uses 4 NAND, 1.25 AND and 2.125 XOR gates per bit of internal state. And the area cost requires 11.33625 $(4\times1+1.25\times1.33+2.125\times2.67=11.33625)$ GE per bit of internal state. As remarked in [10], considering simplify the computations, we often omit the round constant which requires very little XOR gate, like other lightweight block ciphers. Moreover, the key schedule of Loong does not cost gates in Table 6. Table 6 shows total number of operations and theoretical performance of Loong and various lightweight block ciphers.

2) EXPERIMENTAL IMPLEMENTATION

We have carried out two different architectures of Loong implementations which are the round-based architecture and the serialized architecture.

TABLE 6. Total number of operations and theoretical performance of Loong and various lightweight block ciphers. *N* denotes a NAND/NOR Gate, *A* denotes a AND/OR gate, and *X* denotes a XOR/XNOR gate.

Ciphers	nb.of rds	gate cost (per bit per round)			nb.of op.	nb.of op.	round-based
		int.cipher	key sch.	total	w/o key sch.	w/ key shc.	impl.area
Loong-64	16	$4N + 1.25A + 2.215X$		$4N + 1.25A + 2.215X$	7.465×16 =119.44	7.465×16 = **119.44**	$4 \times 1 + 1.25 \times 1.33 + 2.125 \times 2.67$=11.34
SKINNY-64-128	36	$1N + 2.25X$	$0.625X$	$1N + 2.875X$	3.25×36 =117	3.875×36 = 139.5	$1 \times 1 + 2.875 \times 2.67$=8.68
SIMON-64-128	44	$0.5A + 1.5X$	$1.5X$	$0.5A + 3.0X$	2×44 =88	3.5×44 = 154	$1 \times 0.67 + 3 \times 2.67$=8.68
PRESENT-128	31	$1A + 3.75X$	$0.125A + 0.344X$	$1.125A + 4.094X$	4.75×31 =147.2	5.22×31 = 161.8	$1.125 \times 1.33 + 4.094 \times 2.67$=12.43
Piccolo-128	31	$1N + 4.25X$		$1N + 4.25X$	5.25×31 =162.75	5.25×31 = 162.75	$1 \times 1 + 4.25 \times 2.67$=12.35
KATAN-64-80	254	$0.047N + 0.094X$	$3X$	$0.047N + 3.094X$	0.141×254 **=35.81**	3.141×254 = 797.8	$0.047 \times 1.33 + 3.094 \times 2.67$=**8.45**
NOEKEON-128	16	$0.5A + 0.5N + 5.25X$	$0.5A + 0.5N + 5.25X$	$1.0A + 1.0N + 10.5X$	6.25×16 =100	12.5×16 = 200	$1 \times 1 + 1 \times 1.33 + 10.5 \times 2.67$=30.36
AES-128	10	$4.25A + 16X$	$1.06A + 3.5X$	$5.31A + 19.5X$	20.25×10 =202.5	24.81×10 = 248.1	$5.31 \times 1.33 + 19.5 \times 2.67$=59.12

TABLE 7. Comparison of lightweight block ciphers for round based architecture (Encryption + Decryption).

Algorithms	Structure	Rounds	Data(bits)	Key(bits)	Area(GE)	Speed(kbps@100KHz)	Logic Process	Ref.
PRESENT	SPN	31	64	80	2018	200	0.18µm	[3]
PRESENT	SPN	31	64	128	2783	—	0.18µm	[29]
TWINE	GFN	36	64	80	1799	178	90nm	[3]
Piccolo	GFN	25	64	80	1634	237	0.13µm	[3]
LILLIPUT	EGFN	30	64	80	1581	213	LP 65nm	[3]
PRINCE	SPN	12	64	128	3491	533.3	130nm	[11]
Loong	SPN	16	64	64	**1467**	200	0.18µm	This work
Loong	SPN	20	64	80	1542	200	0.18µm	This work
Loong	SPN	32	64	128	1766	200	0.18µm	This work

Round-based Architecture: The round-based architecture is optimized in terms of speed, energy and area. In Fig. 6 shows the round-based architecture of Loong-64. The round-based architecture is the reuse of the round structure and several components. A round of Loong is a clock cycle. This architecture of Loong-64 is run by the encryption module. The datapath of encryption module consists of 64-bit XOR, one 16 S-boxes, Constants, MixRows, Control logic and other counters. In addition, our design needs a data and a key register to store the internal state and key, respectively.

The Loong-64 encryption process needs to take 16 clock cycles, and the hardware implementation of Loong-64 needs 1467.4 GE in the round-based architecture. Since Loong is an involutional structure, the decryption mode reuses the encryption without any addition. Table 7 shows the comparison between the hardware resources of the Loong and the others, which implement both encryption and decryption modes. The experimental results are under the 100 KHz clock frequency. Where Generalized Feistel Networks (GFN) and Extended Generalized Feistel Networks (EGFN) belong to Feistel-type networks.

We describe the hardware resources occupied by each component of the Loong in detail. The 64-bit data register requires 384 GE. And the 64-bit key register of Loong-64 requires 298.88 GE, the 80-bit key register of Loong-80 requires 373.6 GE, the 128-bit key register of Loong-128 requires 597.76 GE. One S-box requires 22.65 GE. The SubCells of Loong has 16 S-boxes, which require 362.4 GE. The AddRoundKey includes a 64-bit XOR operation and a round constant XOR operation, the 64-bit XOR requires 170.88 GE and the round constant XOR requires 42 GE. The MixRows uses the matrix M, which requires 192.24 GE. The logic and other counters of Loong require 17 GE. The P permutation and the shifting operation of columns of not require any hardware resources. Eventually, the area of Loong-64 is 1467.4 GE, and the detailed list is given in table 8. The area of Loong-80 is 1542.12 GE and the area of Loong-128 is 1766.28 GE.

Serialized Architecture: The serialized architecture is to save more chip area by reducing the reuse of components, and the data path of encryption is 4-bit. Due to the repeated use of S-box and matrix M in the round function, Loong is

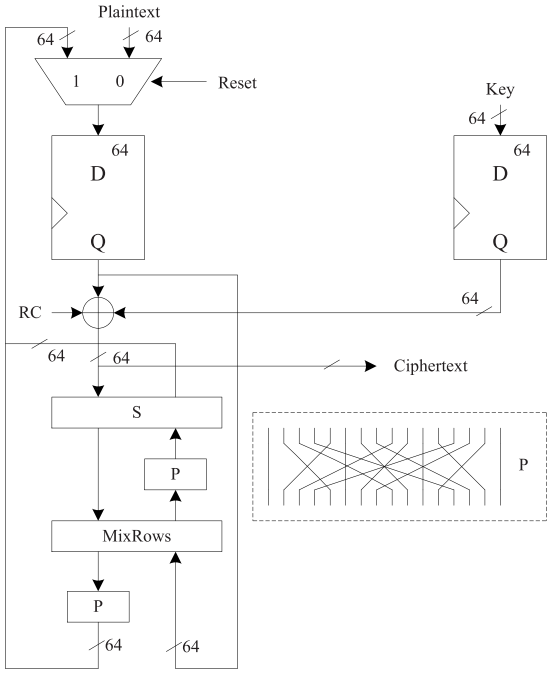


FIGURE 6. The round-based architecture of Loong-64.

TABLE 8. Area requirement of the Loong-64 round based architecture.

Modules	GE	%
Data Register	384	26.17
Key Register	298.88	20.37
Key XOR	170.88	11.65
Constants	42	2.86
SubCells	362.4	24.7
MixRows	192.24	13.1
Control logic and other counters	17	1.16
Total	1467.4	100

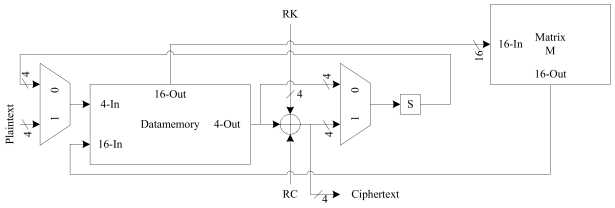


FIGURE 7. The serialized architecture of Loong-64.

very suitable for the serialized architecture to implement in hardware. The datapath of encryption module consists of two 4-bit XORs, one S-box and one matrix M. In addition, our design needs two 4-bit 2-to-1 multiplexers, one data memory and one key memory.

In Fig. 7 shows the serialized architecture of Loong-64. We describe a serialized version of Loong as follows. We mainly describe how the serialized version saves resources compared with the round-based version. The SubCells just need one S-box, which implements 22.65 GE. The implementations of MixRows and the MixColumns only require 48.06 GE instead of 192.24 GE. And the AddRoundKey only need two 4-bit XOR operations for RK and RC, which require 21.36 GE. In addition, the control logical unit of Loong reduces some hardware resources, and the Loong-64, the Loong-80 and the Loong-128 decreased by 61 GE, 73 GE and 73 GE respectively. But the serialized version increases a lot of clock cycles. Eventually, the size of the serialized version of Loong-64, the Loong-80 and the Loong-128 are 835.87 GE, 922.67 and 1146.83 GE. Loong has been compared with LILLIPUT and Piccolo for serial implementation. Table 9 shows the comparison of Piccolo, LILLIPUT, and Loong.

TABLE 9. Comparison of lightweight block ciphers for serialized architecture (Encryption + Decryption).

Algorithms	Data(bits)	Key(bits)	Area(GE)	Logic Process
Piccolo	64	80	1103	0.13μm
LILLIPUT	64	80	1055	LP 65nm
Loong	64	80	**923**	0.18μm

B. SOFTWARE IMPLEMENTATION

The Loong is a fully involutional structure, which is very suitable for hardware and software platforms. In the software implementation, we do not need to consider preserving memory resources for the decryption. In addition, the two SubCells of round function use the same S-box, and the MixRows and the MixColumns use the same diffusion matrix. The design can save the Flash ROM, and require less code size. Loong is not based on a single bit unit, but based on a 4-bit block unit, and has a faster speed to encrypt and decrypt. Moreover, the number of encrypted iterative rounds is a few in Loong. Consequently, Loong can achieve at good results in the software implementation.

The Loong has been implemented by C-code in a software experiment without optimization. The measurements were taken on GCC 4.2.1 with an Intel(R) Core(TM) i5 CPU clocked at 2.3 GHz. The encryption speed of loong-64 is 184544 cycles per block, the encryption speed of loong-80 is 239055 cycles per block, and the encryption speed of loong-128 is 384409 cycles per block.

VI. CONCLUSION

In this paper, we propose a family of involutional lightweight block cipher, called Loong, based on a new SPN structure. Round transformation components are involutional in the new SPN structure, so the encryption and decryption of Loong is the same. The inverse cipher of Loong can absolutely reuse the codes and circuitries to reduce resources in software and hardware implementation. Round function has two SubCells operations, this way makes Loong a more active S-boxes than other lightweight block ciphers in round function. Round function has the MixRows and MixColumns operations, in this way Loong has better diffusion and diffusion effect than other lightweight block ciphers. Because the new SPN structure is highly symmetrical, double use

TABLE 10. Test vectors for Loong-64.

Plaintexts	Keys	Ciphertexts
0000000000000000	0000000000000000	BEB075F4FFE49241
FFFFFFFFFFFFFFFF	0000000000000000	223D4B1CB72E5DA8
80CDE558893D0D88	6DA7D7CA434762B1	4A9B216ADE5D93C0

of resources can be avoided in the round function, the area cost requires 11.33625 GE per bit of internal state, which is lower than per bit of internal state of PRESENT and Piccolo and so on. Moreover, Loong is exceptionally well resistant to differential and linear attacks and also resists related-key attack, algebraic attack, meet-in-the-middle attack and so on. In addition, the round transformation components are lightweight and efficient. On hardware implementation, the resource areas of Loong-64, Loong-80 and Looong-128 are 1467.4 GE, 1542.12 GE and 1766.28 GE respectively in the round-based architecture, and the resource areas of Loong-64, Loong-80 and Loong-128 are 835.87 GE, 922.67 GE and 1146.83 GE respectively in the serialized version. Consequently, Loong achieves not only notably compact implementation but also high security. Furthermore, we welcome all the cryptographic community to comment and analyze.

APPENDIX A
PROOF OF THE ENCRYPTION OF LOONG IS THE SAME AS THE DECRYPTION

Proof: We have to prove that the round function process: Plaintext/Ciphertext → AddRoundKey → (SubCells → MixRows → MixColumns → SubCells → AddRoundKey) → . . . → Ciphertext/Plaintext.

Fig. 8 shows a round operation of the round function and the alphabets represent input and output of each component. We explain the relations between them as following. First, we encrypt a with RK, RC^0 and RC^1: (S represents SubCells)

$$
\begin{aligned}
b &= a \oplus RK \oplus RC^0 \\
c &= S(b),\ c = S(a \oplus RK \oplus RC^0) \\
d &= c \times M,\ d = S(a \oplus RK \oplus RC^0) \times M \\
e &= M' \times d,\ e = M' \times (S(a \oplus RK \oplus RC^0) \times M) \\
f &= S(e),\ f = S(M' \times (S(a \oplus RK \oplus RC^0) \times M)) \\
g &= f \oplus RK \oplus RC^1, \\
g &= S(M' \times (S(a \oplus RK \oplus RC^0) \times M)) \oplus RK \oplus RC^1
\end{aligned}
$$

Outputs the g.

Then, when we decrypt g with RC^0 and RC^1 in reverse order.

$$
\begin{aligned}
g &= S(M' \times (S(a \oplus RK \oplus RC^0) \times M)) \oplus RK \oplus RC^1 \\
h &= g \oplus RK \oplus RC^1 \\
&= S(M' \times (S(a \oplus RK \oplus RC^0) \times M)) = f \\
i &= S(h) \\
&= M' \times (S(a \oplus RK \oplus RC^0) \times M) = e
\end{aligned}
$$

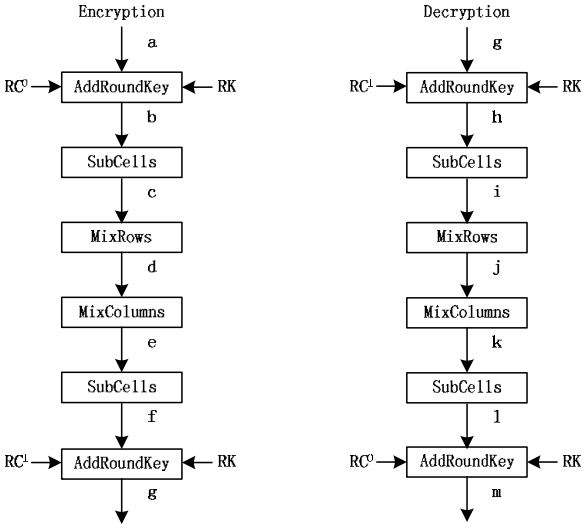


FIGURE 8. The encryption/decryption of a round Loong.

TABLE 11. Test vectors for Loong-80.

Plaintexts	Keys	Ciphertexts
0000000000000000	00000000000000000000	9C4091C4D294DD94
FFFFFFFFFFFFFFFF	00000000000000000000	25B3A5DECADC49BD
80CDE558893D0D88	6DA7D7CA434762B1F85A	2A984CA15E518F23

TABLE 12. Test Vectors for Loong-128.

Plaintexts	Keys	Ciphertexts
0000000000000000	00000000000000000000000000000000	A0BAB18618ADD363
FFFFFFFFFFFFFFFF	00000000000000000000000000000000	D1A878A85DB1CD86
80CDE558893D0D88	6DA7D7CA434762B1F85A666868A18C23	DEFA24C05CBC166F

$$
\begin{aligned}
j &= i \times M \\
&= S(a \oplus RK \oplus RC^0) \times M = d \\
k &= M' \times j \\
&= S(a \oplus RK \oplus RC^0) = c \\
l &= S(k) \\
&= a \oplus RK \oplus RC^0 = b \\
m &= l \oplus RK \oplus RC^0 = a
\end{aligned}
$$

Outputs the a.

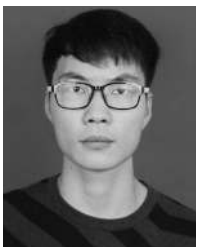
The results of the encryption input and decryption output are both a, so our proof is correct. ■

APPENDIX B
TEST VECTORS OF LOONG-64, LOONG-80 AND LOONG-128

Test vectors of Loong-64, Loong-80 and Loong-128 are given here. The data are expressed in hexadecimal form.

REFERENCES

[1] J. Xiong, J. Ren, L. Chen, Z. Q. Yao, M. Lin, D. Wu, and B. Niu, "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1530–1540, Apr. 2019.
[2] L. Dalmasso, F. Bruguier, P. Benoit, and L. Torres, "Evaluation of SPN-based lightweight crypto-ciphers," *IEEE Access*, vol. 7, pp. 10559–10567, 2019.
[3] T. P. Berger, J. Francq, M. Minier, and G. Thomas, "Extended generalized Feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput," *IEEE Trans. Comput.*, vol. 65, no. 7, pp. 2074–2089, Jul. 2016.
[4] L. Li, B. Liu, and H. Wang, "QTL: A new ultra-lightweight block cipher," *Microprocessors Microsyst.*, vol. 45, pp. 45–55, Aug. 2016.
[5] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *Proc. 9th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 4727, 2007, pp. 450–466.
[6] W. Wu and L. Zhang, "LBlock: A lightweight block cipher," in *Proc. Appl. Cryptography Netw. Secur.*, vol. 6715, 2011, pp. 327–344.
[7] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A small present," in *Proc. 19th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 10529, 2017, pp. 321–345.
[8] L. Li, B. Liu, Y. Zhou, and Y. Zou, "SFN: A new lightweight block cipher," *Microprocessors Microsyst.*, vol. 60, pp. 138–150, Jul. 2018.
[9] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, "Midori: A block cipher for low energy," in *Proc. 21st Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 9453, 2015, pp. 411–436.
[10] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Proc. 36th Annu. Int. Cryptol. Conf.*, vol. 9815, 2016, pp. 123–153.
[11] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, "PRINCE–a low-latency block cipher for pervasive computing applications," in *Proc. 18th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 7658, 2012, pp. 208–225.
[12] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "*Piccolo*: An ultra-lightweight blockcipher," in *Proc. 13th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 6917, 2011, pp. 342–357.
[13] W. T. Zhang, Z. Z. Bao, D. D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, no. 12, pp. 1–15, Dec. 2015.
[14] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: A lightweight block cipher for multiple platforms," in *Proc. Int. Conf. Sel. Areas Cryptogr.*, vol. 7707, 2012, pp. 339–354.
[15] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. (2013). *The Simon And Speck Families of Lightweight Block Ciphers*. [Online]. Available: https://eprint.iacr.org/2013/404.pdf
[16] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Proc. 13th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 6917, 2011, pp. 326–341.
[17] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *Proc. 7th Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*, vol. 7055, 2011, pp. 1–18.
[18] D. Dinu, L. Perrin, A. Udovenko, V. Velichkov, J. Großschädl, and A. Biryukov, "Design strategies for ARX with provable bounds: Sparx and LAX," in *Proc. 22nd Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 10031, 2016, pp. 484–513.
[19] F. Karakoç, H. Demirci, and A. E. Harmancı, "ITUbee: A software oriented lightweight block cipher," in *Proc. Int. Workshop Lightweight Cryptogr. Secur. Privacy*, vol. 8162, 2013, pp. 16–27.
[20] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The Simeck family of lightweight block ciphers," in *Proc. 17th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 9293, 2015, pp. 307–329.
[21] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New lightweight DES variants," in *Proc. 14th Int. Workshop Fast Softw. Encryption*, vol. 4593, 2007, pp. 196–210.
[22] M. R. Albrecht, B. Driessen, E. B. Kavun, G. C. Leander Paar, and T. N. Yalçın, "Block ciphers–focus on the linear layer (feat. PRIDE)," in *Proc. 34th Annu. Cryptol. Conf.*, vol. 8616, 2014, pp. 57–76.
[23] A. Poschmann, S. Ling, and H. Wang, "256 bit standardized crypto for 650 GE–GOST revisited," in *Proc. 12th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, vol. 6225, 2010, pp. 219–233.
[24] B. J. Mohd and T. Hayajneh, "Lightweight block ciphers for IoT: Energy optimization and survivability techniques," *IEEE Access*, vol. 6, pp. 35966–35978, 2018.
[25] W. Kou, "Data encryption standards," in *Networking Security and Standards* (The Springer International Series in Engineering and Computer Science), vol. 394. Boston, MA, USA: Springer, 1997, pp. 49–67.
[26] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.
[27] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. Heidelberg, Germany: Springer, 2002.
[28] *NESSIE*. Accessed: Sep. 20, 2019. [Online]. Available: https://www.cosic.esat.kuleuven.be/nessie/
[29] M. R. Z'aba, N. Jamil, M. E. Rusli, M. Z. Jamaludin, and A. A. M. Yasir, "I-PRESENT: An involutive lightweight block cipher," *J. Inf. Secur.*, vol. 5, no. 3, pp. 114–122, 2014.
[30] NANGATE. *The NanGate 45 nm Opencell Library*. Accessed: Sep. 20, 2019. [Online]. Available: http://www.nangate.com
[31] S. M. Sim, K. Khoo, F. Oggier, and T. Peyrin, "Lightweight MDS involution matrices," in *Proc. 22nd Int. Workshop Fast Softw. Encryption*, vol. 9054, 2015, pp. 471–493.
[32] L. R. Knudsen, "Practically secure Feistel ciphers," in *Proc. Int. Workshop Fast Softw. Encryption*, vol. 809, 1993, pp. 211–221.
[33] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Proc. 7th Int. Conf. Inf. Secur. Cryptol.*, vol. 7537, 2011, pp. 57–76.
[34] N. T. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations," in *Proc. 8th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 2501, 2002, pp. 267–287.
[35] K. Aoki and Y. Sasaki, "Preimage attacks on one-block MD4, 63-step MD5 and more," in *Proc. 15th Int Workshop Sel. Areas Cryptogr.*, vol. 5381, 2008, pp. 267–287.
[36] Y. Igarashi, R. Sueyoshi, T. Kaneko, and T. Fuchida, "Meet-in-the-middle attack with splice-and-cut technique on the 19-round variant of block cipher HIGHT," *Inf. Sci. Appl.*, vol. 339, pp. 423–429, Feb. 2015.
[37] *0.18μm VIP Standard Cell Library Tape Out Ready*, Virtual Silicon Inc., Taiwan, Jul. 2004.



BO-TAO LIU received the B.S. degree from Hengyang Normal University, Hengyang, China, in 2016, the master's degree in computer science from Guizhou University, Guizhou, China, in 2019. Since 2019, he has been a Teacher with the College of Computer Science and Technology, Hengyang Normal University. His current research interests include embedded systems and information security.

LANG LI received the B.S. degree in circuits and systems from Hunan Normal University, in 1996, and the master's and Ph.D. degrees in computer science from Hunan University, Changsha, China, in 2006 and 2010, respectively. Since 2011, he has been a Professor with the College of Computer Science and Technology, Hengyang Normal University, Hengyang, China. His research interests include embedded systems and information security.

RUI-XUE WU received the B.S. degree from the Chengdu University of Information Technology, Chengdu, China, in 2016, and the master's degree in computer science from Guizhou University, Guizhou, China, in 2019. Her current research interests include privacy and data security.

MING-MING XIE received the B.S. degree from Qiannan Normal University for Nationalities, Duyun, China, in 2015, and the master's degree in computer science from Guizhou University, Guizhou, China, in 2019. His current research interests include privacy and data security.

QIU PING LI received the B.S. degree from Jilin University, in 2013, and the Ph.D. degree from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2018. Since 2018, she has been a Teacher with the College of Computer Science and Technology, Hengyang Normal University, Hengyang, China. Her research interest includes information security.