Loong: A Family of Involutional Lightweight Block Cipher

Anmol Sagar[1], Ashutosh Garg[2] and Prince Kumar Pansari[3]

[1] IIT Bhilai, Raipur, India, anmols@iitbhilai.ac.in
[2] IIT Bhilai, Raipur, India, ashutoshg@iitbhilai.ac.in
[3] IIT Bhilai, Raipur, India, princep@iitbhilai.ac.in

Abstract.
In the past few years, Cryptography scientists are mainly focusing on lightweight ciphers because the field of Internet of Things (IoT) has grown fast from the last few years and uses of them will grow as time passes. The IoTs cannot use ciphers like AES that although very secure because it needs more power and resources that is an issue with them. So, the main concern behind lightweight is to use less power and resources, and provide as much security as ciphers like AES provide.
Block ciphers mainly have two structures, SPN based and Feistel Network based. The SPN structure has been widely used in the design of block cipher. However, the algorithm of encryption and decryption in the SPN network is different. Loong is a block cipher that is based on SPN structure. The design of Loong is such that it can use the same implementation for both encryption and decryption. Thus it is very easy to implement on software as well as hardware. The direction of input and output is the same for both encryption and decryption. Rigorous analysis indicates that Loong is of high security against cryptanalysis, especially the differential attack and linear attack. Those facts are based on proper experiments and comparisons. As Loong is easy to implement that makes it more suitable for hardware implementation or IoT, so it is a tough competitor with other lightweight ciphers like PRESENT, TEA etc.

Keywords: Block Cipher · Lightweight Cryptography · Involution · Substitution Permutation Network · Similar Encryption Decryption Process · Save Resources

1 Introduction

Block Cipher is the effective method for resisting adversarial attempts to read data, it is an efficient method to store the data as well as transit on the network. It is one the two common modern symmetric cipher types. It is different from the stream cipher because it deals with the data in chunks. One of the most efficient block ciphers that we are using right now is Advanced Encryption Standard(AES). It is the one of the most famous cipher approved by U.S. Government to protect electronic data, and certified in 2001.
Although AES is one of the most secure ciphers and no-one breaks it till now, research is still going on in the field of cryptography because in this new era we need security in small devices that are called IoT devices. These devices are very useful but in terms of resources and power they have their own limitations. It is said that Security comes at the cost of computation and it's easy to understand. In today's time IoT devices are everywhere, Institutes and militaries are using RFID tags and smart cards. Mobile devices like smart watches, tablets have become a part of daily life. Armies are using drones as their secret weapon. So all these devices are fully content with highly confidential information that can affect some one or many ones life. But if we use a heavy-weight cipher then that consumes

power very early. So in today's time most of the research are going on in the lightweight ciphers.
In last few years many of the block ciphers had developed and still development is going on some of the famous Lightweight block ciphers PRESENT, TEA and so on.
Block cipher mainly has two structures, SPN based and Feistel Network. One of the famous examples of Feistel networks is DES(Data Encryption Standard), The advantage of Feistel Network is the process to encrypt and decrypt the data is the same. But Feistel Network makes diffusion of the half block in each round so to make the block secure we need too many rounds that make it inefficient. That's why people majorly uses SPN because it provides more efficiency then Feistel but hard to implement on hardware because it uses different encryption and decryption processes.
AES is one of the best SPN block ciphers but it has some limitations, (1) It's costly to make two different circuits on hardware for encryption and decryption. (2) It is not useful for smart cards because they have very less memory to store data. Therefore, It is much needed to make a cipher like SPN that have similar encryption and decryption processes.

2 Specification of Loong

Loong Cipher is SPN based cipher. There are three members of Loong Cipher each with a block size of 64 bits, but three different key lengths: 64, 80 and 128 bits. The round numbers are 16 for 64 bit key, 20 for 80 bits and 32 for 128 bits. The members of Loong are denoted as **Loong-64**, **Loong-80** and **Loong-128** on the basis of length of the keys.

3 Encryption

In encryption, the 64-bit block of plaintext and the primary key is regarded as input. The size of primary key can be 64-bit, 80-bit or 128-bit. The round function of Loong includes AddRoundKey operation, SubCell operation, MixRow operation and MixColumn operation. The encryption process of Loong Cipher is given in Figure 1.
The encryption function of Loong can be expressed as:

$$ENC_{RN} = ARK(RK, RC^0) \circ (\bigcirc_{r=1}^{RN} SC \circ MR \circ MC \circ SC \circ ARK(RK, RC^{RN})) \quad (1)$$

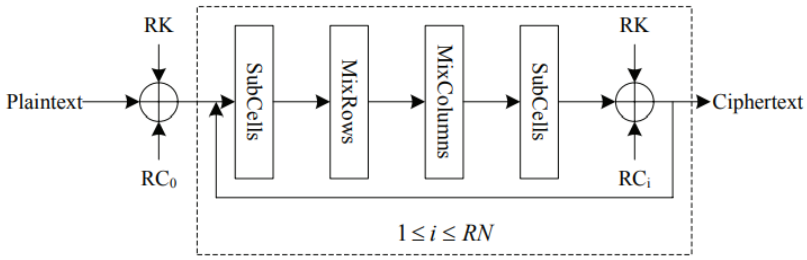


Figure 1: Encryption in Loong

The algorithm 1 illustrates the ENC_{RN} equation given earlier, where the input to the algorithm is plaintext and round key and the output is ciphertext.

Algorithm 1 Encryption Algorithm

ENC_{RN}
Input : Plaintext, RK(Round Key);
Output: Ciphertext;
01: state ← Plaintext;
02: AddRoundKey(state, RK, RC);
03: for i = 1 to RN do
04: SubCell(state);
05: MixRow(state);
06: MixColumn(state);
07: SubCell(state);
08: AddRoundKey(state, RK, RC);
09: endfor
10: Ciphertext ← state;
11: **Return** Ciphertext;

3.1 Subcell

The Subcell operation is used to provide confusion. It means this operation is used to obscure the relationship between the plaintext and the ciphertext. In this operation, we use a S-Box which performs substitution. Since, in loong we work on 4bit nibbles, so possible values for nibbles are 0 to E. So, the S-Box contains 16 elements. The S-Box is given in the Table 1.

Table 1: S-Box

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

There is a special property of **involutive** in the SBox of loong cipher. It means that the SBox is its own inverse. The same property is illustrated in the Table 2.

Table 2: x = S(S(x))

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

3.2 MixRow

The MixRow Operation is used to provide better diffusion. In MixRow Operation, the state is post-multiplied by a diffusion matrix M. The matrix multiplication is performed in finite field $GF(2^4)$ where the irreducible polynomial is $x^4 + x + 1$.

The matrix multiplication operation of the MixRow is as follows:

$$state \leftarrow \begin{pmatrix} state_0 & state_1 & state_2 & state_3 \\ state_4 & state_5 & state_6 & state_7 \\ state_8 & state_9 & state_{10} & state_{11} \\ state_{12} & state_{13} & state_{14} & state_{15} \end{pmatrix} \times \begin{pmatrix} 1 & 4 & 9 & 13 \\ 4 & 1 & 13 & 9 \\ 9 & 13 & 1 & 4 \\ 13 & 9 & 4 & 1 \end{pmatrix} \tag{2}$$

3.3 MixColumn

The MixColumn Operation is also used to provide better diffusion. In MixColumn Operation, the state is pre-multiplied by a diffusion matrix $M^{'}$. The matrix multiplication is performed in finite field $GF(2^4)$ where the irreducible polynomial is $x^4 + x + 1$.

The matrix multiplication operation of the MixColumn is as follows:

$$state \leftarrow \begin{pmatrix} 13 & 9 & 4 & 1 \\ 9 & 13 & 1 & 4 \\ 4 & 1 & 13 & 9 \\ 1 & 4 & 9 & 13 \end{pmatrix} \times \begin{pmatrix} state_0 & state_1 & state_2 & state_3 \\ state_4 & state_5 & state_6 & state_7 \\ state_8 & state_9 & state_{10} & state_{11} \\ state_{12} & state_{13} & state_{14} & state_{15} \end{pmatrix} \tag{3}$$

3.4 Round Constants

The round constants of this cipher is of 6-bit. It can be denoted as $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$
It is generated by the 6-bit affine linear-feedback shift register(LFSR).
The update function is defined as:

$$(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0) \leftarrow (rc_4, rc_3, rc_2, rc_1, rc_0, rc_5 \oplus rc_4 \oplus 1) \tag{4}$$

The round constants are initialized to zero, means all the 6-bits are 0. And it is updated before using in a given round with the update function defined earlier.
The Table 3 is the pre-computed element of round constants of all the rounds.

Table 3: Elements of Round Constants

Rounds	Constants										
0-10	01	03	07	0F	1F	3E	3D	3B	37	2F	1E
11-21	3C	39	33	27	0E	1D	3A	35	2B	16	2C
22-32	18	30	21	02	05	0B	17	2E	1C	38	31

Loong has three diiferent key blocks with 16, 20 and 32 rounds respectively. Round Constant are used in AddRoundKey operation of every round and there is also one extra AddRoundKey operation earlier. So, 17, 21 and 33 different round constants are required for three different key size respectively.
The adding(XOR) of the round constants in the AddRoundKey is arranged as follows:

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

3.5 AddRoundKey

Loong has three different key length which are 64-bit, 80-bit and 128-bit. The 64-bit key is written as $k_0, k_1, ..., k_{15}$, the 80-bit key is written as $k_0, k_1, ..., k_{19}$ and the 128-bit key is written as $k_0, k_1, ..., k_{31}$.
The 64-bit key of Loong-64 is arranged into a round key matrix as:

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

The 80-bit key of Loong-80 is arranged into two round key matrix as:

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

The 128-bit key of Loong-128 is arranged into two round key matrix as:

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

For 64-bit key, the AddRoundKey operation is as:

$$state \leftarrow state \oplus RK \oplus RC_i \quad (0 \leq i \leq RN) \tag{5}$$

For 80-bit and 128-bit key, the AddRoundKey Operation is as:

$$state \leftarrow state \oplus RK_{i \bmod 2} \oplus RC_i \quad (0 \leq i \leq RN) \tag{6}$$

4 Decryption

The decryption algorithm in loong cipher is same as the encryption. Means the direction of input and output data are same for both encryption and decryption. The only difference with the encryption is that here the round constants are used in reverse order.

The decryption function of Loong can be expressed as:

$$DEC_{RN} = ARK(RK, RC^{RN}) \circ (\bigcirc_{r=RN-1}^{0} SC \circ MR \circ MC \circ SC \circ ARK(RK, RC^0)) \tag{7}$$

Since, the algorithm of encryption and decryption is same, decryption can reuse the implementation of the encryption in both hardware and software. It uses less resource(memory) to implement, so it will be very easy to implement in hardware.

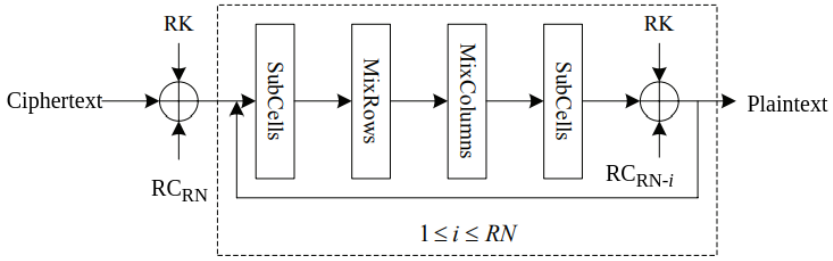


Figure 2: Decryption in Loong

5 Observations

5.1 DDT-Difference Distribution Table

As we can see from the DDT Table 4,the highest value (except 16, where input both input and output difference is 0) is 4.
Therefore, the maximum differential probability of the SBox is $\frac{4}{16} = \frac{1}{4}$

Table 4: DDT of S-Box

I/O	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	2	4	-	2	2	2	-	2	-	-	-	-	-	2	-
2	-	4	-	-	4	-	-	-	2	2	-	-	2	2	-	-
3	-	-	-	-	2	-	4	2	-	4	-	-	2	-	-	2
4	-	2	4	2	2	2	-	-	-	2	2	-	-	-	-	-
5	-	2	-	-	2	-	-	4	2	-	2	2	2	-	-	-
6	-	2	-	4	-	-	-	2	2	-	-	-	-	4	2	-
7	-	-	-	2	-	4	2	-	-	-	-	2	2	2	-	2
8	-	2	2	-	-	2	2	-	-	2	-	2	2	-	2	-
9	-	-	2	4	2	-	-	-	2	2	-	2	2	-	-	-
10	-	-	-	-	2	2	-	-	-	-	2	2	-	4	-	4
11	-	-	-	-	-	2	-	2	2	2	2	2	-	2	-	2
12	-	-	2	2	-	2	-	2	2	2	-	-	2	-	2	-
13	-	-	2	-	-	-	4	2	-	-	4	2	-	-	2	-
14	-	2	-	-	-	-	2	-	2	-	-	-	2	2	2	4
15	-	-	-	2	-	-	-	2	-	-	4	2	-	-	4	2

5.2 LAT-Linear Approximation Table

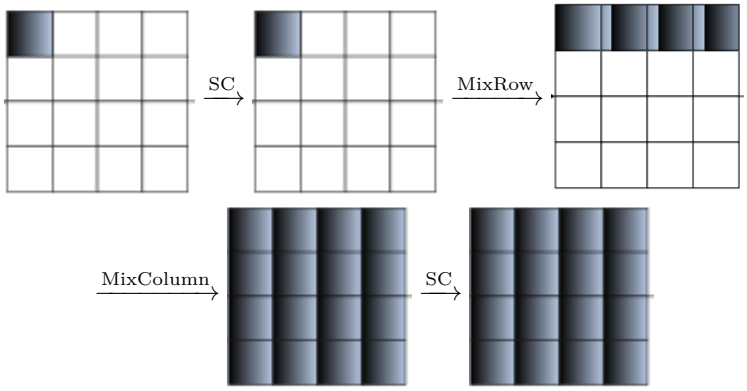
The highest absolute value observed in the LAT Table 5 is 4(ignoring the point where alpha and beta is zero) indicating it occurs with probability $\frac{12}{16} = \frac{3}{4}$.

Table 5: LAT of S-Box

I/O	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	4	-	-2	-2	2	-2	-2	2	2	2	4	-	-	-
2	-	4	-	-	4	-	-	-	-4	-	-	-	-	4	-	-
3	-	-	-	-	-2	-2	2	2	2	-2	-2	2	-	4	-	4
4	-	-2	4	-2	2	-	-2	-	-2	-4	-2	-	-	-2	-	2
5	-	-2	-	-2	-	-2	-	-2	-	2	-4	-2	-	2	4	-2
6	-	2	-	2	-2	-	2	-4	-2	-	-2	-	-4	-2	-	2
7	-	-2	-	2	-	-2	-4	-2	-	2	-	-2	-	2	-4	2
8	-	-2	-4	2	-2	-	-2	-	-4	-2	-	2	2	-	2	-
9	-	2	-	-2	-4	2	-	2	-2	-	-2	-4	2	-	-2	-
10	-	2	-	-2	-2	-4	-2	-	-	-2	4	-2	-2	-	2	-
11	-	2	-	2	-	-2	-	-2	2	-4	-2	-	2	-	-2	-4
12	-	4	-	-	-	-	-4	-	2	2	-2	2	2	-2	2	2
13	-	-	4	4	-2	2	-2	2	-	-	-	-	-2	2	2	-2
14	-	-	-	-	-	4	-	-4	2	-2	2	-2	2	2	2	2
15	-	-	-	4	2	-2	2	2	-	-	-	-4	2	-2	2	2

5.3 Diffusion Spread in Loong

Here you can observe that if one bit is differ in two different states, then how the difference or diffusion spread in the whole 64-bit block,



So after one round diffusion spread in whole 64-bit block.

6 SECURITY ANALYSIS

6.1 DIFFERENTIAL ATTACK AND LINEAR ATTACK

For differential attack maximum differential characteristic probability is small enough, the cipher can resist differential cryptanalysis and the differential characteristic probability upper bound often can be calculated by the number of active S-boxes, So if we find total number of active S-Box then we can comment Differential attack.

To calculate the minimum number of active S-Box we construct mixed integer linear programming that is also called as MILP. Round Function process can be represented as : SubCells → MixRows → MixColumns → SubCells. Changes of the first round of loong

can be represented as follows. Every variable x as a nibble of Loong state. If the difference is non-zero, the variable x is 1, otherwise, x is 0. all variable are represent input of the sub-cell so equation are summed in the objective function which basically represent active S-Box.

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.

$$\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \xrightarrow{\text{SC}} \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \xrightarrow{\text{Mix Row}} \begin{bmatrix} x_{16} & x_{17} & x_{18} & x_{19} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{24} & x_{25} & x_{26} & x_{27} \\ x_{28} & x_{29} & x_{30} & x_{31} \end{bmatrix}$$

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

The MixRows can be constrained by the following linear equations:

$$x_0 + x_1 + x_2 + x_3 + x_{16} + x_{20} + x_{24} + x_{28} - 5d_0 \geq 0 \tag{8}$$
$$x_4 + x_5 + x_6 + x_7 + x_{17} + x_{21} + x_{25} + x_{29} - 5d_1 \geq 0 \tag{9}$$
$$x_8 + x_9 + x_{10} + x_{11} + x_{18} + x_{22} + x_{26} + x_{30} - 5d_2 \geq 0 \tag{10}$$
$$x_{12} + x_{13} + x_{14} + x_{15} + x_{19} + x_{23} + x_{27} + x_{31} - 5d_3 \geq 0 \tag{11}$$

The MixColumns can be constrained by the following linear equations:

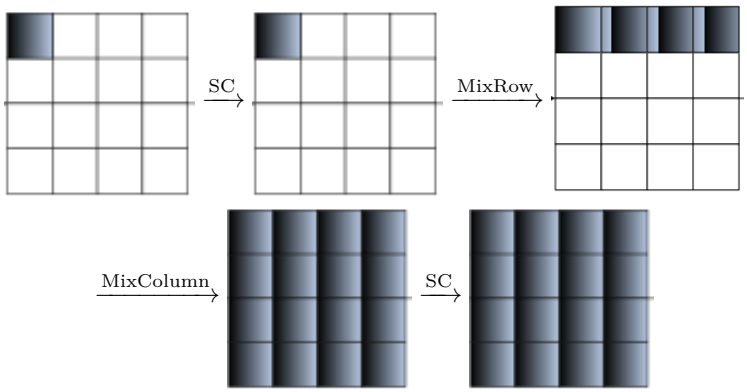
$$x_{16} + x_{17} + x_{18} + x_{19} + x_{32} + x_{33} + x_{34} + x_{35} - 5d_4 \geq 0 \tag{12}$$
$$x_{20} + x_{21} + x_{22} + x_{23} + x_{36} + x_{37} + x_{38} + x_{39} - 5d_5 \geq 0 \tag{13}$$
$$x_{24} + x_{25} + x_{26} + x_{27} + x_{40} + x_{41} + x_{42} + x_{43} - 5d_6 \geq 0 \tag{14}$$
$$x_{28} + x_{29} + x_{30} + x_{31} + x_{44} + x_{45} + x_{46} + x_{47} - 5d_7 \geq 0 \tag{15}$$

From above Equation we get minimum number of differential or linear active S-boxes for n rounds of Loong is 8n, after first round minimum 8 S-Box are active, after first round minimum 32 S-Box are active.
The S-box of Loong have maximal probability of a differential is 2^2 and the maximal absolute bias of a linear approximation is 2^2 . For Loong, it's 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.

6.2 RELATED-KEY ATTACK

In this analyze the probability of related-key differential characteristics. For 64-bit key version. The 64-bit primary key is round key. When the differences are inserted in the

round key input,the round key will be active in Loong-64.



After obtaining related-key differential path, they are ensured that at least one round is active in every two rounds by this method. So Loong-64 has at least (RN /2) × 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}
Hence it's secure under related-key attacks.

6.3 ALGEBRAIC ATTACK

The Algebraic attacks are based on the deducing the secret key by solving the nonlinear equations that involves key bits, cipher-text and plain-text. This is based on the major work of **Shannon**.
In the Algebraic attack we deal with two phases:

1. Analyzed sufficiently enough low-degree or multivariate nonlinear equation.
2. Recover the key by solving the equation.

For the *Loong* non-linear component is Sub-Cell or you can say s-box. for each round of *Loong* you are using sub-cell two times and each sub-cell uses 16 s-boxes that leads to 32 s-boxs in each round. Here, A intresting result that we have is '*A 4 × 4 S-box can be described by 21 quadratic equations of 8 input/output-bit variables over*'[1].
For, Loong-64 has 512 sbox after 16 rounds of encryption. similarly, Loong-80 has 640 s-boxes after 20 round and Loong-128 has 1024 s-boxes after 32 round of encryption. Therefore Loong-64 can be described as 10752 quadratic equa- tions 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. Here is the comparison of Loong with other lightweight cipher.
From the comparison table 6 you can conclude that Loong is getting much better results compare to other lightweight ciphers. You can see that if you can break Loong-80 from algebric attack then you can break PRESENT-80 That is still one of the best lightweight cipher.

6.4 MEET-IN-THE-MIDDLE ATTACK

The Meet in the middle is one of the known-plaintext attack. It is based on the trade-off between time and memory that also rely on performing multiple encryption oprations in

[1]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

Table 6: Algebraic Comparison in Differnet Ciphers

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

sequence.
For the Loong Diffusion is faster, and after 2 round it spread in each cell, The block of 64 bit is fully defused. So the number of round used in Partial matching is 3(2*Number of round to defused - 1 = 2*2 -1 = 3). It is part of the intermediate state as it uses alot more then 3 rounds. you can not get the primary key by this process too. With the splice and cut technique, meet in the middle attack can alalyze upto 7 rounds That it worst case for now. Therefore, Loong is safe for this attack for now.

7 Conclusion

In this paper, we **YAnonymous** studied Loong which is a Lightweight Block cipher based on SPN structure. The S-Box of the Loong is involutional in nature, which means that the S-Box is its own inverse. The round transformation of the Loong are also involutional, which makes both the encryption and decryption process same. The decryption process of the Loong can simply use the codes and circuitries of the encryption process to save resources in both the software and hardware implementation. Since, the Round Function of Loong has two subcell operations which makes the more active S-Box compared to its competitors Lightweight block ciphers. Since, it has more active S-Box, so it will also provide better confusion which means there is a obscure relationship between the plaintext and the ciphertext.
From the cryptoanalysis perspective Loong is Secure too. Loong has high security and we believe Loong is enough to resist against differential and linear attacks. It has two operations for diffusion, one MixRow and the other one MixColumn. So, It provides better diffusion than other Lightweight block ciphers. It is not vulnerable for modern attack like algebraic or Meet-in-the-middle attack. The two times of sub-cell in each round give it high confusion and makes it very hard to break.

8 Reference

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.
A Journals on Loong: A Family of Involutional Lightweight Block Cipher Based on SPN Structure
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.
L. Dalmasso, F. Bruguier, P. Benoit, and L. Torres, “Evaluation of SPN- based lightweight crypto-ciphers,” IEEE Access, vol. 7, pp. 10559–10567, 2019.