



## Research Article

# Related-Key Multiple Impossible Differential Cryptanalysis on Full-Round LiCi-2 Designed for IoT

Kai Zhang <sup>1,2</sup>, Xuejia Lai <sup>1</sup>, Lei Wang,<sup>1</sup> Jie Guan,<sup>2</sup> Bin Hu,<sup>2</sup> Senpeng Wang,<sup>2</sup> and Tairong Shi<sup>2</sup>

<sup>1</sup>Shanghai Jiao Tong University, Shanghai 201100, China

<sup>2</sup>PLA SSF Information Engineering University, Zhengzhou 450000, China

Correspondence should be addressed to Kai Zhang; zhkai2010@139.com and Xuejia Lai; laix@sjtu.edu.cn

Received 15 November 2021; Revised 12 March 2022; Accepted 22 March 2022; Published 25 May 2022

Academic Editor: Marimuthu Karuppiah

Copyright © 2022 Kai Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

LiCi-2 is an ultralightweight block cipher designed for constrained IoT devices. It is a successor of LiCi and has even better performance in both software and hardware implementation. In this paper, based on the idea of related-key multiple impossible differential cryptanalysis, a key recovery attack on full-round LiCi-2 is proposed. First, an interesting property is revealed that, with a single bit difference in the related key, a 10-round differential character with probability of 1 exists on LiCi-2. With an automatic approach, the boundaries of impossible differential distinguishers in terms of single-key setting and related-key setting are explored. Under our construction method, the longest length is 8 rounds for single-key setting and 18 rounds for related-key setting. Finally, based on these 18-round distinguishers, a 25-round key recovery attack is proposed with adding 3 rounds before and 4 rounds after the distinguisher. Our attack needs one related key. The time complexity for our attack is  $O(2^{123.44})$ , the memory complexity is  $O(2^{94})$ , and the data complexity is  $O(2^{60.68})$ . As far as we know, no full-round attack has previously been reported on LiCi-2.

## 1. Introduction

As a representative for the new information age, the Internet of things (IoT) has shown its strong vitality in many fields. It can be viewed as an expanded network based on the Internet and it combines various sensing devices to form a huge network, which enables a wider interconnection of people, machines, and things. However, the sensing devices in IoT usually have very limited resources such as computing resource, power resource, and hardware implementation resource. So, it is very essential to propose lightweight primitives and ensure the information security in IoT devices. In recent years, many good lightweight block ciphers are proposed such as PRESENT [1], GIFT [2], Midori [3], LBlock [4], RECTANGLE [5], SIMON, and SPECK [6]. In 2019, NIST (National Institute of Standards and Technology) proposed a standardization project LWC (LightWeight Cryptography) to enhance the development for lightweight ciphers. Among these lightweight block ciphers, LiCi

(denoted as LiCi-1 in this paper to avoid confusion) is a lightweight block cipher proposed in 2017 [7]. The block size for LiCi-1 is 64-bit and the key size is 128-bit. It totally has 31 rounds. Up to now, the security margin on LiCi-1 is enough. To reach better performance, as a successor, LiCi-2 was proposed in 2018 [8]. For LiCi-2, it reduces the total rounds from 31 to 25, and half of the subkeys and a circular shift are omitted for better performance. LiCi-2 is designed for extremely constrained devices, such as 6LoWPAN. It shows good performance in memory requirement, power consumption, and hardware implementation.

Impossible differential cryptanalysis was originally proposed by Knudsen [9] and Biham et al. [10], respectively. It is one of the most effective cryptanalytic methods so far. The basic idea of impossible differential cryptanalysis is establishing an impossible differential distinguisher and filtering the wrong key candidates with this distinguisher until the correct key is recovered. Related-key impossible differential cryptanalysis is a variant of impossible

TABLE 1: Summary of previous results on LiCi-1 and LiCi-2.

Cipher	Type of attack	Attacked/total rounds	Length of the distinguisher	Time complexity	Memory complexity	Data complexity	Reference
LiCi-1	Linear	16/32	—	—	—	$2^{106}$	[7]
	Differential	16/32	—	—	—	$2^{96}$	
	Integral	13/32	12	$2^{83}$	$2^{41}$	$2^{63}$	[16]
		16/32	10	$2^{173}$	$2^{119}$	$2^{63.6}$	
LiCi-2	Impossible differential	16/32	10	$2^{83.08}$	$2^{76.76}$	$2^{59.76}$	[17]
	Linear	20/25	—	—	—	$2^{106}$	[8]
	Differential	20/25	—	—	—	$2^{80}$	
	Impossible differential	25/25	18	$2^{123.44}$	$2^{94}$	$2^{60.68}$	<b>This paper</b>

differential cryptanalysis. This attack is based on the assumption that the attacker has access to the encryption oracle and can use related keys as input. This kind of attack has been applied to many block ciphers successfully, such as AES and LBlock [11–15].

**Our Contributions:** The main purpose of this paper is to evaluate the security level on LiCi-2 against related-key multiple impossible differential cryptanalysis. There are three main contributions which are listed as follows:

- (i) A 10-round differential distinguisher with probability of 1 is constructed. With this property, many differential-type distinguishers can be extended for better attack.
- (ii) The length and number of impossible differential distinguishers on LiCi-2 in single-key setting and related-key setting are explored in an automatic approach. In the single-key setting, the longest length of the distinguisher under our construction method is 9-round and there are altogether 240 such distinguishers discovered. In the related-key setting, the longest length of the distinguishers discovered is 18-round and 65 such distinguishers are presented.

- (iii) Based on twenty of these 18-round related-key distinguishers, a key recovery attack on full-round LiCi-2 is proposed with adding 3 rounds before and 4 rounds after the distinguisher. The summary of our result and prior results on LiCi-1 and LiCi-2 is presented in Table 1.

This paper is organized as follows. Section 2 introduces the notations used throughout this paper. Section 3 gives a brief description on LiCi-2. Section 4 presents some observations and properties on LiCi-2. In Section 5, a key recovery attack on LiCi-2 is proposed, and Section 6 concludes the paper.

## 2. Notations

Suppose that  $E$  is an  $n$ -bit block cipher of  $R$  rounds; the following notations are used throughout this paper:

- (i)  $K$ : 128-bit master key,  $K^t$  represents the value of the 128-bit key register at the  $t$ -th round and  $K_{[i-j]}^t$  represents the  $i$ -th bit to the  $j$ th bit of  $K$
- (ii)  $RK^i$ : 32-bit round key at the  $i$ -th round

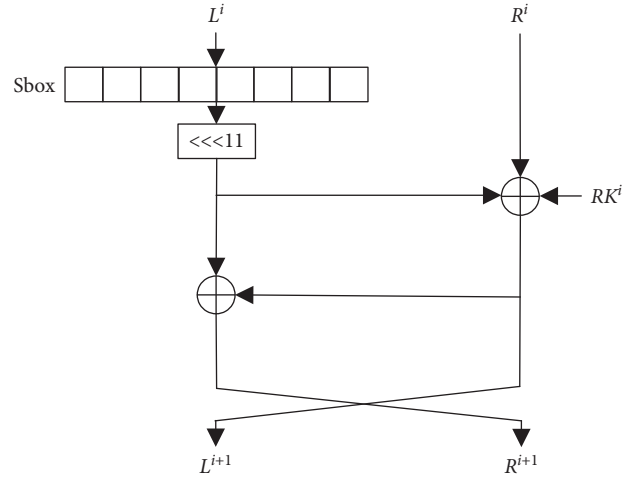


FIGURE 1: Round function of LiCi-2.

TABLE 2: Sbox in LiCi-2.

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

- (iii)  $(L^i, R^i)$ : input for the  $i$ -th round,  $L^i$  represents the left branch,  $R^i$  represents the right branch, and  $L_{[t_1-t_2]}^i$  ( $R_{[t_1-t_2]}^i$ ) represents the  $t_1$ -th bit to the  $t_2$ -th bit of  $L^i$  ( $R^i$ )
- (iv)  $S^i$ : output of the Sbox layer for the  $i$ -th round
- (v) “ $<<< i$ ”: left rotation with  $i$  bits
- (vi) “ $\oplus$ ”: XOR operation

## 3. Brief Description on LiCi-2

LiCi-2 is a lightweight block cipher that was proposed in 2018 [8]. It adopts a Feistel-type structure. The block size for LiCi-2 is 64-bit and the key size is 128-bit. It totally has 25 rounds. Each round consists of Sbox constitution, key addition, and circular shift. The round function is depicted in Figure 1 and the 4-bit Sbox used in LiCi-2 is depicted in Table 2.

TABLE 3: Related position of key registers with  $\Delta RK_{[19]}^0$  at rounds 0 to 9.

Round	0	1	2	3	4	5	6	7	8	9
Position	21	34	47	60	73	86	99	112	125	10

TABLE 4: Differential transmission property for the Sbox of LiCi-2.

Input difference	0001	0100	1000	1100	1101	**11	**1*	**0*	***1	***0
Output difference	1***	***1	***1	1**0	0***	0001	0010	0011	0100	0101

Key Schedule: The scale of master key  $K$  is 128-bit. It can be denoted as  $K = k_{127}, k_{126}, k_{125}, \dots, k_2, k_1, k_0$ . First, the 128-bit key register is initialized with the master key. The register is rotated by 13 bits on the left. Every four of the eight least significant bits of the key register are substituted with the Sbox. XOR the binary form of the round number to the bits  $[k_{63}k_{62}k_{61}k_{60}k_{59}]$ . The 32 least significant bits are extracted as the round key  $RK^i$ . The mathematical form of this updating progress can be illustrated as follows:

- (1)  $[k_{127}k_{126} \dots k_1k_0] \lll 13$
- (2)  $[k_3k_2k_1k_0] = S[k_3k_2k_1k_0]$
- (3)  $[k_7k_6k_5k_4] = S[k_7k_6k_5k_4]$
- (4)  $[k_{63}k_{62}k_{61}k_{60}k_{59}] = [k_{63}k_{62}k_{61}k_{60}k_{59}] \oplus RC^i$
- (5)  $RK^i = [k_{31}k_{30} \dots k_1k_0]$

#### 4. Observations on LiCi-2

In this section, some new observations and properties on LiCi-2 are introduced. First, a 10-round differential character with probability one is constructed in Property 1. The boundaries of impossible differential distinguishers on LiCi-2 in single-key setting and related-key setting are discussed in Observation 1 and Observation 2. These distinguishers will be a foundation for our later attack.

*Property 1 (a 10-Round Differential Character).* Given two plaintext pairs whose difference is  $(\Delta L^0, \Delta R^0)$  and a related key whose difference is  $\Delta K$ , if  $\Delta R_{[21]}^0 = 1$ ,  $\Delta K_{[21]} = 1$ , and other differences are all zeros, the difference of the internal state at the beginning of the 11th round  $(\Delta L^{10}, \Delta R^{10})$  will have only two-bit difference  $\Delta L_{[10]}^{10} = 1$ ,  $\Delta R_{[10]}^{10} = 1$  with probability of 1.

*Proof.* The difference of the first subkey  $\Delta RK^0$  can be derived from  $\Delta K$ , where  $\Delta RK_{[21]}^0 = 1$  and other bits are all zeros. According to the round function, the difference for the input of the second round can be calculated as follows:

$$\begin{cases} \Delta L^1 = \Delta R^0 \oplus \Delta RK^0 \oplus (\Delta S(\Delta L^0) \lll 11), \\ \Delta R^1 = (\Delta S(\Delta L^0) \lll 11) \oplus \Delta L^1. \end{cases} \quad (1)$$

As  $\Delta L^0 = 0$  and  $\Delta R^0 = \Delta RK^0$ , we can derive  $(\Delta L^1, \Delta R^1) = (0, 0)$ . So, after several consecutive rounds, the difference of the internal state will remain zero until some  $\Delta RK^i$  bring in a new nonzero difference.

Let us consider the differential transfer property of the key schedule. According to the key schedule, the single-bit difference of the master key will go through the following positions of the key register (Table 3) at the first 10 rounds.

After analyzing these positions, the nonzero difference will not appear in the subkeys until the 10th round, that is,  $\Delta RK_{[10]}^9$ . So the differences of  $(\Delta L^1, \Delta R^1)$ ,  $(\Delta L^2, \Delta R^2)$ ,  $\dots$ ,  $(\Delta L^9, \Delta R^9)$  will all be zeros and, for  $(\Delta L^{10}, \Delta R^{10})$ ,  $\Delta L_{[10]}^{10} = 1$ ,  $\Delta R_{[10]}^{10} = 1$  due to the difference of  $\Delta RK_{[10]}^9$ .

*Former Observation 1* [17]. For the Sbox of LiCi-2, there are some differential characters for the input and output difference as indicated in Table 4.

This observation was proposed on LiCi-1. As the Sboxes for these two ciphers are identical, it can also be used on LiCi-2. It reveals the details of the differential transmission property for the Sbox. With this property, we can construct longer impossible differential distinguishers.

*Observation 1 (Boundary of Single-Key Impossible Differential Distinguishers).* For LiCi-2, in single-key setting, considering the details of the Sbox in Table 4, the maximum length of impossible differential distinguisher is 8 rounds and there are 544 such distinguishers in terms of one active bit input and output difference. If the numbers of active bits for the input and output difference are both limited to 2, the maximum length of single-key impossible differential distinguisher is 9 rounds and there are 240 such 9-round distinguishers. Summary of the statistics for these distinguishers is illustrated in Table 5.

As expected by the designers, this new version of LiCi has better diffusion property. The result in Observation 1 validates this improvement. For the original LiCi-1, for single-bit difference, the maximum length can reach 10 rounds and, for 1-bit input and 2-bit output difference, the maximum length can reach 11 rounds.

*Observation 2 (Boundary of Related-Key Impossible Differential Distinguishers).* For LiCi-2, in related-key setting, considering the details of the Sbox in Table 4, the maximum length of impossible differential distinguisher is 17 rounds and there are 64 such distinguishers in terms of one active bit in input, output, and key difference. If the number of active bits is 1 for input and key and 2 for output, the maximum length of related-key impossible differential distinguisher can reach 18 rounds and there are 65 such distinguishers (all these 65 distinguishers are presented in the Appendix). The summary of all these distinguishers is illustrated in Table 6.





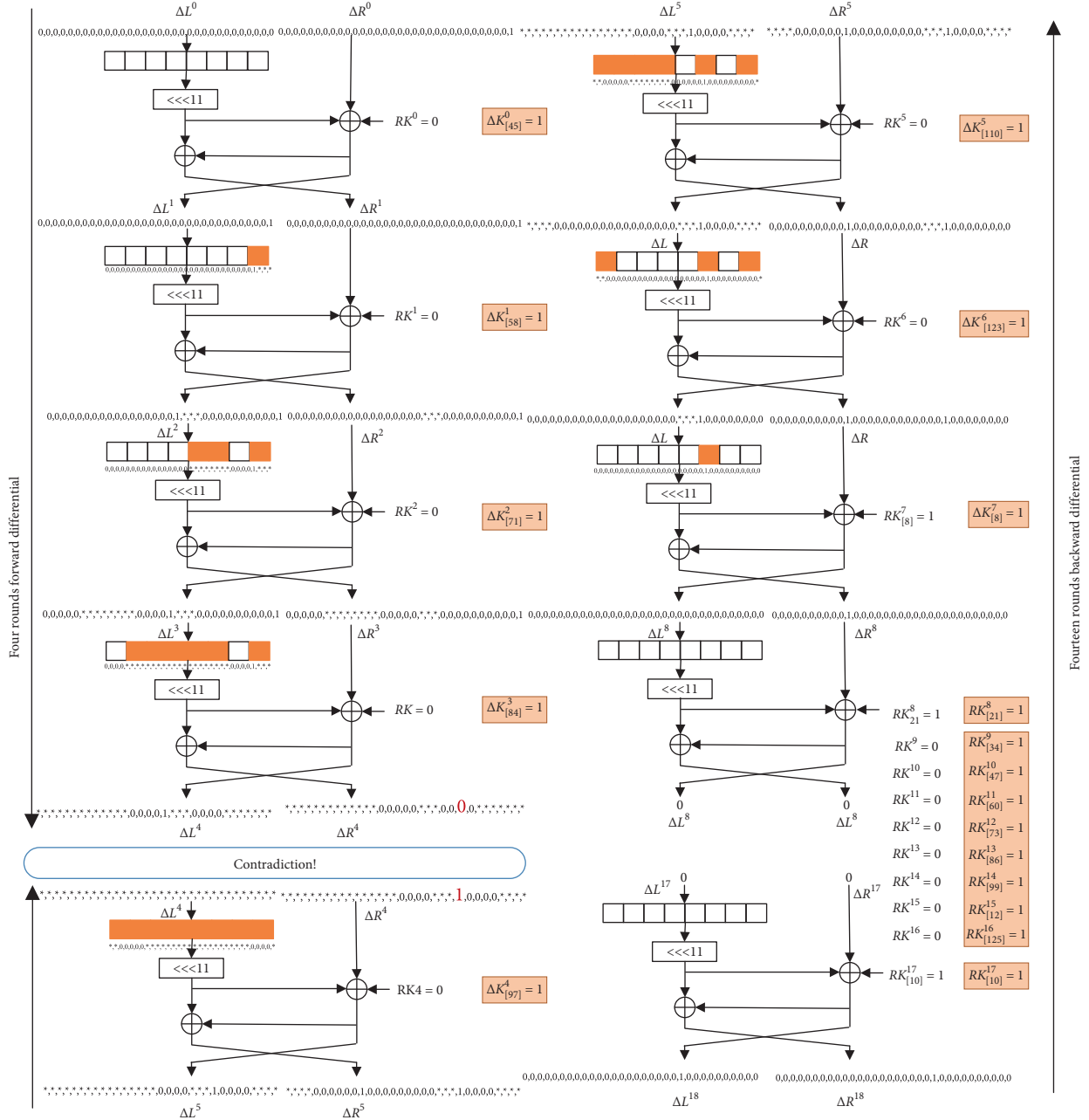


FIGURE 2: A concrete 18-round related-key impossible differential distinguisher on LiCi-2.

- (x)  $n_{in}$ : the number of equivalent impossible differential distinguishers with different input differences and same output differences
- (xi)  $n_{out}$ : the number of equivalent impossible differential distinguishers with same input difference and different output differences

Suppose that the block cipher we are dealing with is of block size  $n$  and a key  $K$  of size  $|K|$ .

5.1.2. *Main Conclusions.* The main conclusions are as follows.

- (1) The relationship between  $N$  and  $P$  is as follows:

$$P = \left(1 - 2^{-(c_{in} + c_{out})}\right)^N < \frac{1}{2}. \quad (7)$$

The minimum value of  $N$ , denoted by  $N_{min}$ , is about  $2^{c_{in} + c_{out}}$ .

- (2) The time complexity of the attack is approximated by  $T$ :

$$T = \left(C_N + \left(N + 2^{|k_{in} \cup k_{out}|} \cdot \frac{N}{2^{c_{in} + c_{out}}}\right) \cdot C'_E + 2^{|K|} \cdot P\right) \cdot C_E, \quad (8)$$

where  $C_N = \max \left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N \cdot 2^{n+1-|\Delta|}} \right\}, N \cdot 2^{n+1-|\Delta_{in}|-|\Delta_{out}|} \right\}$ ,  $C_N$  represents the data complexity needed to construct  $N$  plaintext-ciphertext pairs, with  $N$  and  $P$  satisfying condition (2) above,  $C_E$  represents the cost for one full-round encryption, and  $C'_E$  represents the ratio of the cost for the partial encryption to the full encryption.

The memory complexity of the attack is determined by  $N$ .

- (3) To apply multiple impossible differential cryptanalysis, suppose that  $n_{in} \cdot n_{out}$  impossible differential distinguishers are derived from a single distinguisher. With more distinguishers, the time complexity of the attack can also be calculated with equation (8); however, previous  $|\Delta_{in}|$  and  $|\Delta_{out}|$  are replaced with  $|\Delta'_{in}|$  and  $|\Delta'_{out}|$ , where  $|\Delta'_{in}| = |\Delta_{in}| + \log_2(n_{in})$  and  $|\Delta'_{out}| = |\Delta_{out}| + \log_2(n_{out})$ .

## 5.2. Procedures of the Attack

**5.2.1. Data Collection Stage.** On one hand, construct  $2^m$  data sets. For each data set, to make the difference of a plaintext pair satisfy  $(\Delta L^0, \Delta R^0)$  in Figure 3, there are 34 bits that can be arbitrary value and other 30 bits are fixed to a constant for a data set. On the other hand, suppose that the difference for the output of the 25th round is  $(\Delta L^{25}, \Delta R^{25})$ ,  $\Delta L^{25}_{[27,3]} = 1$ ,  $\Delta R^{25}_{[27,3]} = 0$  (4 bits).

**5.2.2. Key Recovery Stage.** In the key recovery stage, we guess the subkey bits in its smallest unit to reduce the complexity. This process is depicted in Figure 3 to make it more intuitive. The details for the key recovery stage are illustrated in the following steps:

- (i) Step 1. Sieve plaintext-ciphertext pairs according to the difference of the first round and last round. This process can be launched without the key.
- (i) Step 1.1. For each plaintext-ciphertext pair, sieve those pairs whose difference for the input and output of  $\Delta S^0_{[7-4]}$  satisfies  $[*, *, *, * - 0, 0, 0, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise. There are 3 bit conditions for this step.
- (ii) Step 1.2. Sieve those pairs whose difference for the input and output of  $\Delta S^0_{[11-8]}$  satisfies  $[*, *, *, * - *, 1, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (2 bit conditions).
- (iii) Step 1.3. Sieve those pairs whose difference for the input and output of  $\Delta S^0_{[15-12]}$  satisfies  $[*, *, *, * - 0, 0, 0, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).
- (iv) Step 1.4. Sieve those pairs whose difference for the input and output of  $\Delta S^0_{[23-20]}$  satisfies  $[*, *, 1, * - 0, 0, 1, 0]$ . If the pairs can satisfy the

condition, save the pair or discard it otherwise. (3 bit conditions).

- (v) Step 1.5 Sieve those pairs whose difference for the input and output of  $\Delta S^0_{[31-28]}$  satisfies  $[*, *, *, * - *, *, *, 0]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).
- (vi) Step 1.6. Sieve the remaining pairs with  $(\Delta L^1, \Delta R^1)$ . If  $\Delta R^1_{[15-8]} = 0$ , keep the pair (8 bit conditions).
- (vii) Step 1.7. Sieve those pairs whose difference for the input and output of  $\Delta S^{24}_{[3-0]}$  satisfies  $[1, *, *, * - *, *, *, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).
- (viii) Step 1.8. Sieve those pairs whose difference for the input and output of  $\Delta S^{24}_{[15-12]}$  satisfies  $[0, *, *, * - *, *, *, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).
- (ix) Step 1.9. Sieve those pairs whose difference for the input and output of  $\Delta S^{24}_{[19-16]}$  satisfies  $[1, 0, 0, 0 - *, *, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).
- (x) Step 1.10. Sieve those pairs whose difference for the input and output of  $\Delta S^{24}_{[23-20]}$  satisfies  $[1, *, *, * - *, *, *, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).
- (xi) Step 1.11. Sieve those pairs whose difference for the input and output of  $\Delta S^{24}_{[27-24]}$  satisfies  $[1, 0, 0, 0 - *, *, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).
- (xii) Step 1.12. Sieve the remaining pairs with  $\Delta R^{24}$ . If  $\Delta R^{24}_{[26-24, 19-15]} = 0$ ,  $\Delta R^{24}_{[23]} = 1$ , keep the pair (9 bit conditions).
- (ii) Step 2. Sieve the plaintext-ciphertext pairs according to the difference of the second round.
- (i) Step 2.1. Guess  $RK^0_{[11-8]}$  and test whether the difference of  $\Delta S^1_{[11-8]}$  satisfies the form  $[*, *, *, * - *, 1, *, 0]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (2 bit conditions).
- (ii) Step 2.2. Guess  $RK^0_{[15-12]}$  and test whether the difference of  $\Delta S^1_{[15-12]}$  satisfies the form  $[*, *, *, * - 0, 0, 0, *]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (3 bit conditions).
- (iii) Step 2.3. Guess  $RK^0_{[23-20]}$  and test whether the difference of  $\Delta S^1_{[23-20]}$  satisfies the form  $[*, *, 1, * - 0, 0, 1, 0]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (3 bit conditions).
- (iv) Step 2.4. Sieve the remaining pairs with  $\Delta R^2$ . If  $\Delta R^2_{[23, 22, 20]} = 0$ , keep the pair (3 bit conditions).

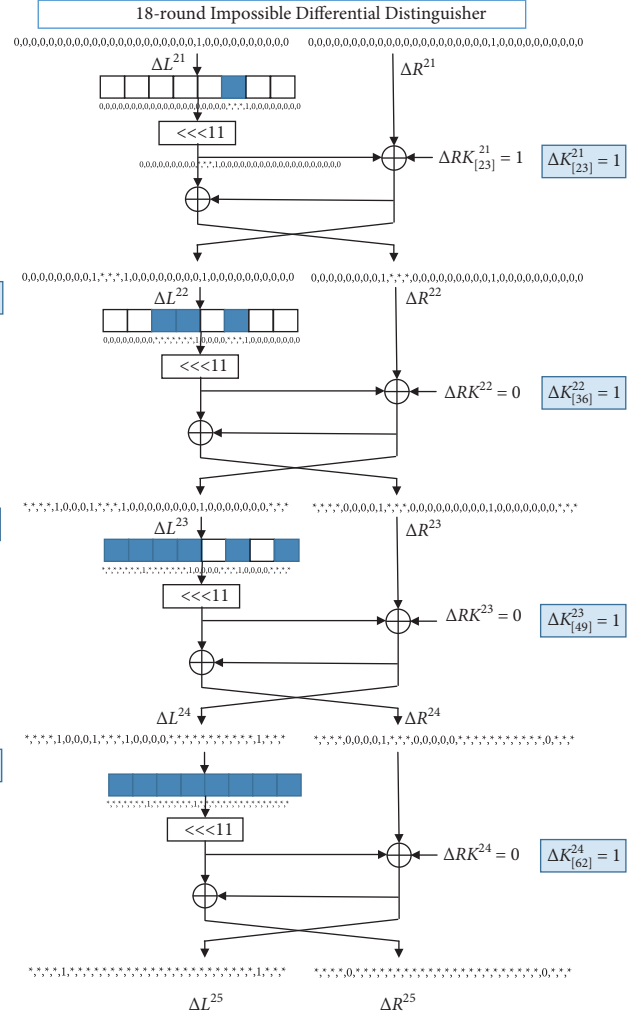
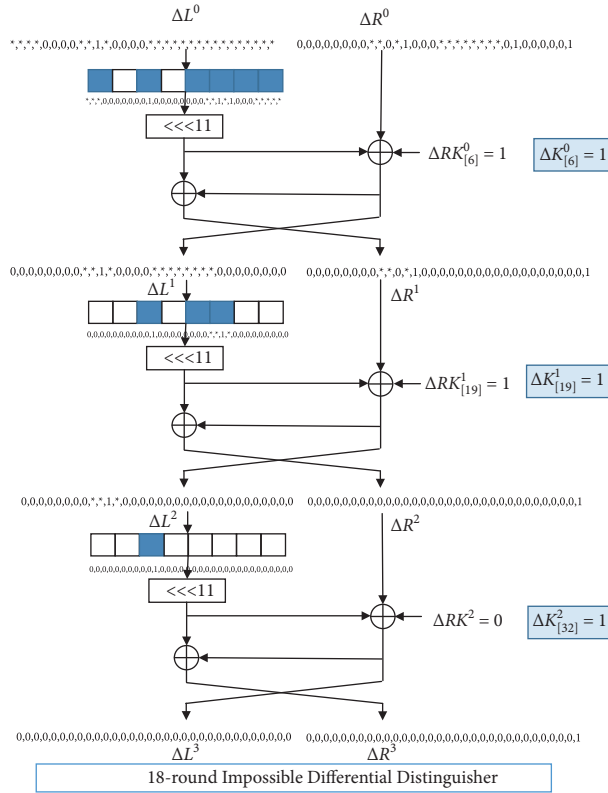


FIGURE 3: Key recovery attack on 25-round LiCi-2.

(iii) Step 3. Sieve the plaintext-ciphertext pairs according to the difference of the 23th round.

(i) Step 3.1. Guess  $RK_{[14-11]}^{24}$  and test whether the difference for the input and output of  $\Delta S_{[3-0]}^{23}$  satisfies  $[0, *, * - *, *, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).

(ii) Step 3.2. Guess  $RK_{[22-19]}^{24}$  and test whether the difference for the input and output of  $\Delta S_{[11-8]}^{23}$  satisfies  $[0, 1, 0, 0 - *, *, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).

(iii) Step 3.3. Guess  $RK_{[30-27]}^{24}$  and test whether the difference for the input and output of  $\Delta S_{[19-16]}^{23}$  satisfies  $[1, 0, 0, 0 - *, *, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).

(iv) Step 3.4. Guess  $RK_{[31,2-0]}^{24}$  and test whether the difference for the input and output of  $\Delta S_{[23-20]}^{23}$  satisfies  $[1, *, * - *, *, *, *]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (1 bit condition).

(v) Step 3.5. Guess  $RK_{[6-3]}^{24}$  and test whether the difference for the input and output of  $\Delta S_{[27-24]}^{23}$  satisfies  $[1, 0, 0, 0 - *, *, *, 1]$ . If the pairs can satisfy the condition, save the pair or discard it otherwise (3 bit conditions).

(vi) Step 3.6. Sieve the remaining pairs with  $\Delta R^{23}$ . If  $\Delta R_{[14-11,9-4]}^1 = 0$ ,  $\Delta R_{[10]}^1 = 1$ , keep the pair. There are  $2m + 3$  plaintext-ciphertext pairs left (11 bit conditions).

(iv) Step 4. Sieve the plaintext-ciphertext pairs according to the difference of the 22th round.

(i) Step 4.1. Guess  $RK_{[22-19]}^{23}$  and test whether the input and output differences of  $\Delta S_{[11-8]}^{22}$  satisfy the form  $[0, 1, 0, 0 - *, *, *, 1]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (3 bit conditions).

(ii) Step 4.2. Guess  $RK_{[30-27]}^{23}$  and  $RK_{[10-7]}^{24}$  and test whether the input and output differences of  $\Delta S_{[19-16]}^{22}$  satisfy the form  $[1, 0, 0, 0 - *, *, *, 1]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (3 bit conditions).



- (iii) Step 4.3 Guess  $RK_{[31,2]}^{23}$  and test whether the input and output differences of  $\Delta S_{[23-20]}^{22}$  satisfy the form  $[1, *, *, * - *, *, *, *]$  for the remaining pairs. If the pairs can satisfy the condition, save the pair (1 bit condition). It is noted that  $RK_{[1,0]}^{23} = RK_{[14,13]}^{24}$  have already been guessed in Step 3.1 and these redundant related keys will not be listed in the rest of the steps for simplicity.
- (iv) Step 4.4. Sieve the remaining pairs with  $\Delta R^{22}$ . If  $\Delta R_{[31-28,2-0]}^{22} = 0$ , keep the pair (7 bit conditions).
- (v) Step 5. Sieve the plaintext-ciphertext pairs according to the difference of the 3rd round. Guess extra 4 subkey bits  $RK_{[23-20]}^1$  and calculate  $\Delta S_{[23-20]}^2$ . If  $\Delta S_{[23-20]}^2$  satisfies the form  $[*, *, 1, * - 0, 0, 1, 0]$  for the remaining pairs and if the pairs can satisfy the condition, save the pair (3 bit conditions).
- (vi) Step 6. Sieve the incorrect subkey candidates according to the difference of the 21th round. Guess extra 4 subkey bits  $RK_{[22-19]}^{22}$  and test whether the input and output differences of  $\Delta S_{[11-8]}^{21}$  satisfy the form  $[0, 1, 0, 0 - *, *, *, 1]$  and if  $\Delta R_{[22-20]}^{21} = 0$  (6 bit conditions) and if the pairs can satisfy the condition, it will lead to an impossible differential distinguisher; thus the guessed subkey candidate is false. Discard this key candidate and test another one, and iterate this process to reduce the space of key candidates.

**5.3. Complexities of the Attack.** According to our attack, the values are  $|\Delta_{in}| = 34$ ,  $|\Delta_{out}| = 60$ ,  $c_{in} = 34$ , and  $c_{out} = 60$ . As the data complexity for this attack always exceeds the full code book, it is infeasible to propose a valid impossible differential cryptanalysis with a single distinguisher. So, we will use multiple distinguishers to reduce this data complexity. Concretely speaking, twenty 18-round distinguishers with the following form are used (all the 65 18-round distinguishers can be found at the Appendix).

Input difference:  $\Delta R_{[21]}^3 = 1$ . Key difference:  $\Delta K_{[110]} = 1$

Output difference:  $\Delta L_{[i]}^{21}, \Delta R_{[i]}^{21} = 1, i \in \{0 - 10, 15 - 18, 27 - 31\}$

So  $n_{out} = 20$  and  $|\Delta_{out}'| = |\Delta_{out}| + \log_2(n_{out}) = 60 + 4.32 = 64.32$ .

As there are altogether 23 active Sboxes (50 independent subkey bits) involved in the calculation,  $C_E' = (23/8 \cdot 24) = 2^{-3.12}$ . According to the equations in Section 5.1, the data complexity, memory complexity, and time complexity for our attack are calculated as follows:

Data complexity:  $C_N = \max \{ \min \{ \sqrt{2^{94}} \cdot 2^{64+} - 1 - 34, \sqrt{2^{94}} \cdot 2^{64+1-64.32} \}, 2^{94} \cdot 2^{64+1-34-64.32} \} = 2^{60.68}$ .

Memory complexity:  $N = 2^{94}$  plaintext pairs and corresponding ciphertext pairs.

Time complexity:  
 $T = (2^{60.68} + (2^{94} + 2^{50} \cdot (2^{94}/2^{94})) \cdot 2^{-3.12} + 2^{128} \cdot 0.3679) \cdot 2^{-3.12} \approx 2^{123.44}$  LiCi-2 encryption.

## 6. Conclusion

In this paper, based on the idea of related-key multiple impossible differential cryptanalysis, a full-round attack on LiCi-2 is proposed. As a first step, several properties and observations on LiCi-2 are proposed to give an overall estimation on LiCi-2 against single-key and related-key impossible differential cryptanalysis. Some 18-round related-key impossible differential distinguishers are proposed alongside. As a second step, based on the newly constructed 18-round impossible differential distinguishers, a full-round key recovery attack is presented with adding three rounds before and four rounds after the distinguisher. In the future, some modifications on LiCi-2 are needed to avoid the reported attack.

## Appendix

All the 18-round related-key impossible differential distinguishers are based on the one-bit input and key difference, as well as 2-bit output difference.

- (i)  $e[i][j]$  represents a single-bit difference at the  $j$ th bit; if  $i = 0$ , the difference appears at the left branch; if  $i = 1$ , the difference appears at the right branch.

No. 1: Input difference:  $e[1][0]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 2: Input difference:  $e[1][0]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 3: Input difference:  $e[1][1]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 4: Input difference:  $e[1][1]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 5: Input difference:  $e[1][2]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 6: Input difference:  $e[1][2]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 7: Input difference:  $e[1][3]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 8: Input difference:  $e[1][3]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 9: Input difference:  $e[1][6]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 10: Input difference:  $e[1][7]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 11: Input difference:  $e[1][14]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 12: Input difference:  $e[1][14]$ . Output difference:  $e[0][11], e[1][11]$ . Key difference:  $e[7]$

No. 13: Input difference:  $e[1][15]$ . Output difference:  $e[0][10], e[1][10]$ . Key difference:  $e[6]$

No. 14: Input difference:  $e[1][15]$ . Output difference:  $e[0][11], e[1][11]$ . Key difference:  $e[7]$

No. 15: Input difference:  $e[1][18]$ . Output difference:  $e[0][12], e[1][12]$ . Key difference:  $e[8]$

No. 16: Input difference:  $e[1][19]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 17: Input difference:  $e[1][21]$ . Output difference:  $e[0][0]$ ,  $e[1][0]$ . Key difference:  $e[110]$   
 No. 18: Input difference:  $e[1][21]$ . Output difference:  $e[0][1]$ ,  $e[1][1]$ . Key difference:  $e[110]$   
 No. 19: Input difference:  $e[1][21]$ . Output difference:  $e[0][2]$ ,  $e[1][2]$ . Key difference:  $e[110]$   
 No. 20: Input difference:  $e[1][21]$ . Output difference:  $e[0][3]$ ,  $e[1][3]$ . Key difference:  $e[110]$   
 No. 21: Input difference:  $e[1][21]$ . Output difference:  $e[0][4]$ ,  $e[1][4]$ . Key difference:  $e[110]$   
 No. 22: Input difference:  $e[1][21]$ . Output difference:  $e[0][5]$ ,  $e[1][5]$ . Key difference:  $e[110]$   
 No. 23: Input difference:  $e[1][21]$ . Output difference:  $e[0][6]$ ,  $e[1][6]$ . Key difference:  $e[110]$   
 No. 24: Input difference:  $e[1][21]$ . Output difference:  $e[0][7]$ ,  $e[1][7]$ . Key difference:  $e[110]$   
 No. 25: Input difference:  $e[1][21]$ . Output difference:  $e[0][8]$ ,  $e[1][8]$ . Key difference:  $e[110]$   
 No. 26: Input difference:  $e[1][21]$ . Output difference:  $e[0][9]$ ,  $e[1][9]$ . Key difference:  $e[110]$   
 No. 27: Input difference:  $e[1][21]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[110]$   
 No. 28: Input difference:  $e[1][21]$ . Output difference:  $e[0][15]$ ,  $e[1][15]$ . Key difference:  $e[110]$   
 No. 29: Input difference:  $e[1][21]$ . Output difference:  $e[0][16]$ ,  $e[1][16]$ . Key difference:  $e[110]$   
 No. 30: Input difference:  $e[1][21]$ . Output difference:  $e[0][17]$ ,  $e[1][17]$ . Key difference:  $e[110]$   
 No. 31: Input difference:  $e[1][21]$ . Output difference:  $e[0][18]$ ,  $e[1][18]$ . Key difference:  $e[110]$   
 No. 32: Input difference:  $e[1][21]$ . Output difference:  $e[0][27]$ ,  $e[1][27]$ . Key difference:  $e[110]$   
 No. 33: Input difference:  $e[1][21]$ . Output difference:  $e[0][28]$ ,  $e[1][28]$ . Key difference:  $e[110]$   
 No. 34: Input difference:  $e[1][21]$ . Output difference:  $e[0][29]$ ,  $e[1][29]$ . Key difference:  $e[110]$   
 No. 35: Input difference:  $e[1][21]$ . Output difference:  $e[0][30]$ ,  $e[1][30]$ . Key difference:  $e[110]$   
 No. 36: Input difference:  $e[1][21]$ . Output difference:  $e[0][31]$ ,  $e[1][31]$ . Key difference:  $e[110]$   
 No. 37: Input difference:  $e[1][22]$ . Output difference:  $e[0][0]$ ,  $e[1][0]$ . Key difference:  $e[111]$   
 No. 38: Input difference:  $e[1][22]$ . Output difference:  $e[0][1]$ ,  $e[1][1]$ . Key difference:  $e[111]$   
 No. 39: Input difference:  $e[1][22]$ . Output difference:  $e[0][2]$ ,  $e[1][2]$ . Key difference:  $e[111]$   
 No. 40: Input difference:  $e[1][22]$ . Output difference:  $e[0][3]$ ,  $e[1][3]$ . Key difference:  $e[111]$   
 No. 41: Input difference:  $e[1][22]$ . Output difference:  $e[0][5]$ ,  $e[1][5]$ . Key difference:  $e[111]$   
 No. 42: Input difference:  $e[1][22]$ . Output difference:  $e[0][7]$ ,  $e[1][7]$ . Key difference:  $e[111]$   
 No. 43: Input difference:  $e[1][22]$ . Output difference:  $e[0][8]$ ,  $e[1][8]$ . Key difference:  $e[111]$   
 No. 44: Input difference:  $e[1][22]$ . Output difference:  $e[0][9]$ ,  $e[1][9]$ . Key difference:  $e[111]$

No. 45: Input difference:  $e[1][22]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 46: Input difference:  $e[1][22]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[111]$   
 No. 47: Input difference:  $e[1][22]$ . Output difference:  $e[0][11]$ ,  $e[1][11]$ . Key difference:  $e[7]$   
 No. 48: Input difference:  $e[1][22]$ . Output difference:  $e[0][31]$ ,  $e[1][31]$ . Key difference:  $e[111]$   
 No. 49: Input difference:  $e[1][23]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 50: Input difference:  $e[1][23]$ . Output difference:  $e[0][11]$ ,  $e[1][11]$ . Key difference:  $e[7]$   
 No. 51: Input difference:  $e[1][24]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 52: Input difference:  $e[1][24]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 53: Input difference:  $e[1][25]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 54: Input difference:  $e[1][25]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 55: Input difference:  $e[1][26]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 56: Input difference:  $e[1][26]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 57: Input difference:  $e[1][27]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 58: Input difference:  $e[1][27]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 59: Input difference:  $e[1][28]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 60: Input difference:  $e[1][30]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 61: Input difference:  $e[1][30]$ . Output difference:  $e[0][11]$ ,  $e[1][11]$ . Key difference:  $e[7]$   
 No. 62: Input difference:  $e[1][30]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$   
 No. 63: Input difference:  $e[1][31]$ . Output difference:  $e[0][10]$ ,  $e[1][10]$ . Key difference:  $e[6]$   
 No. 64: Input difference:  $e[1][31]$ . Output difference:  $e[0][11]$ ,  $e[1][11]$ . Key difference:  $e[7]$   
 No. 65: Input difference:  $e[1][31]$ . Output difference:  $e[0][12]$ ,  $e[1][12]$ . Key difference:  $e[8]$

## Data Availability

All data included in this study are available upon request by contact with the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants nos. 61802437, 62102448, 61972248, and 61902428 and China Postdoctoral Science Foundation under Grant no. 2020M681314.

## References

- [1] A. Bogdanov, L. R. Knudsen, G. Leander et al., "PRESENT: An Ultra-lightweight Block cipher," *Cryptographic Hardware and Embedded Systems*, Springer, Berlin, Heidelberg, 2007.
- [2] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: a small present," in *Cryptographic Hardware and Embedded Systems* Springer Cham, Manhattan, NY, USA, 2017.
- [3] S. Banik, A. Bogdanov, T. Isobe et al., "Midori: a block cipher for low energy," *Advances in Cryptology - ASIACRYPT 2015*, Springer, Berlin, Heidelberg, 2015.
- [4] W. Wu and L. Zhang, "A Lightweight Block cipher," *Cryptography and Network Security*, Springer, Berlin, Heidelberg, 2011.
- [5] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, vol. 58, no. 12, pp. 1-15, 2015.
- [6] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference*, pp. 1-6, San Francisco, CA, USA, June 2015.
- [7] J. Patil, G. Bansod, and K. S. Kant, "LiCi: a new ultra-lightweight block cipher," in *Proceedings of the 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, pp. 40-45, IEEE, Pune, India, February 2017.
- [8] S. Khairnar, G. Bansod, and V. Dahiphale, "A light weight cryptographic solution for 6LoWPAN protocol stack," *Advances in Intelligent Systems and Computing. Science and Information Conference*, Springer Cham, Manhattan, NY, USA, 2018.
- [9] L. R. Knudsen, "Deal - a 128-bit block cipher," *Complexity*, vol. 258, no. 2, p. 216, 1998.
- [10] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," *Advances in Cryptology - EUROCRYPT '99. International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 1999.
- [11] E. Biham, O. Dunkelman, and N. Keller, "Related-key impossible differential attacks on 8-round AES-192," *Topics in Cryptology - CT-RSA 2006. In Cryptographers' Track at the RSA Conference*, Springer, Berlin, Heidelberg, 2006.
- [12] W. Zhang, W. Wu, L. Zhang, and D. Feng, "Improved related-key impossible differential attacks on reduced-round AES-192," in *International Workshop on Selected Areas in Cryptography* Springer, Berlin, Heidelberg, (2006 August).
- [13] M. Xie, J. Li, and Y. Zang, "Related-key impossible differential cryptanalysis of LBlock," *Chinese Journal of Electronics*, vol. 26, no. 1, pp. 35-41, 2017.
- [14] Y. Wei, C. Li, and B. Sun, "Related-key impossible differential cryptanalysis on crypton and crypton v1. 0," in *Proceedings of the 2011 World Congress on Internet Security (WorldCIS-2011)*, pp. 227-232, IEEE, London, UK, February 2011.
- [15] Q. Yang, L. Hu, S. Sun, and L. Song, "Related-key impossible differential analysis of full Khudra," in *International Workshop on Security* Springer Cham, Manhattan, NY, USA, 2016.
- [16] W. Xin, B. Sun, and C. Li, "Bit-based integral attack on LiCi algorithm," *Computer Engineering*, vol. 46, no. 7, pp. 136-142, 2020.
- [17] Y. Wei, J. Shi, and L. Li, "Impossible differential cryptanalysis of LiCi block cipher," *Journal of Electronics and Information Technology*, vol. 41, no. 7, pp. 1610-1617, 2019.
- [18] C. Boura, M. Naya-Plasencia, and V. Suder, "Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and Simon," in *International Conference on the Theory and Application of Cryptology and Information Security* Springer, Berlin, Heidelberg, 2014.
- [19] C. Boura, V. Lallemand, M. Naya-Plasencia, and V. Suder, "Making the impossible possible," *Journal of Cryptology*, vol. 31, no. 1, pp. 101-133, 2018.