Hindawi
Security and Communication Networks
Volume 2022, Article ID 4291000, 8 pages
https://doi.org/10.1155/2022/4291000

Research Article

Slide Attack on Full-Round ULC Lightweight Block Cipher Designed for IoT

Kai Zhang,[1,2] **Xuejia Lai**,[2] **Lei Wang**,[2] **Jie Guan**,[1] **and Bin Hu**[1]

[1]*PLA SSF Information Engineering University, Zhengzhou, China*
[2]*Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Kai Zhang; zhkai2010@139.com

Received 12 January 2022; Accepted 17 February 2022; Published 23 April 2022

Academic Editor: Jie Cui

To meet the security demand for IoT devices, many new lightweight block ciphers are proposed. ULC is a lightweight block cipher designed for the IoT. It has many advantages in terms of memory use, efficiency, and security. In this paper, a slide attack on full-round ULC is proposed in a related key setting. First, two properties on ULC are discovered. The first property is presented to illustrate the property of a slid pair for ULC. The second property is introduced to construct a link between some round key bits and some master key bits. Based on these properties, a key recovery attack on ULC is proposed. In this attack, with the condition of one related key, all the 80 master key bits can be recovered with the data complexity of $O(2^{32})$, the memory complexity of $O(2^{32})$, and the time complexity of $O(2^{63})$. The result of this paper indicates that ULC can't resist slide attack in related key settings.

1. Introduction

With the rapid development of the Internet of Things (IoT), it is getting deep into all fields of people's lives. However, the devices used in the IoT are usually microcomputing equipment, which have weak computing power. Unlike traditional computers and high-performance computers, the computing and storage capacity for those devices in IoT are limited. So, it is very important to propose some better lightweight ciphers and guarantee the security of IoT devices. In order to solve this problem, the topic of lightweight block ciphers has been a hot topic for the last several years, and a number of new lightweight block ciphers have been proposed, which are represented by PRESENT [1], Lblock [2], RECTANGLE [3], TWINE [4], Piccolo [5], SIMON [6], Lilliput [7], SPECK [6], Simeck [8], HIGHT [9], LEA [10], etc. Among these ciphers, lightweight block ULC [11] is proposed in 2021 and designed for IoT. When compared with traditional block ciphers, it has many advantages in terms of performance, memory usage, and security level. However, a thorough security evaluation of these lightweight block ciphers is essential before actual usage.

Slide attack [12] is proposed by Biryukov and Wagner at FSE 1999. In the Journal of Cryptology (2018), more efficient slide attacks are proposed to reduce the time complexity of the slide attack [13]. At Eurocrypt 2020, several new types of slide attacks are proposed to overcome the asymmetry of the last round [14]. In recent years, quantum slide attacks have been proposed for better results [15, 16]. This attack has been applied successfully on many ciphers such as Keeloq, GOST, FF3, Trivium, IEC 62055-41, CLX-128, Spectr-H64, SHACAL-1, and WG-16 [17–24]. Among these results, the most notable ones are breaking the block ciphers Keeloq, FF3, and GOST, which are real-life cryptosystems. Keeloq is used to protect cars against theft and also is used in garage door openers. FF3 is the format-preserving encryption scheme selected as a US standard [25] by NIST. The GOST block cipher is a Russian encryption standard. The basic idea of slide attack is to use the property of a slid pair to derive some information about the key bits.

Our contribution is as follows: our overall contribution is proposing a slide attack on full-round ULC. First, two properties under ULC are presented. The first property is

presented to illustrate the property of a slid pair for ULC. The second property is introduced to construct a link between some round key bits and the master key bits, which will be used in our attack. Based on these properties, a key recovery attack on ULC is proposed. In this attack, with the condition of one related key, 80-bit master key bits can be recovered with the data complexity of $O(2^{32})$, the memory complexity of $O(2^{32})$, and the time complexity of $O(2^{63})$.

The comparison between our results and some previous cryptanalytic results is illustrated in Table 1.

Outline. This paper is organized as follows: the notations used in this paper are illustrated in Section 2. Section 3 gives a brief description of slide attack and ULC. Section 4 proposes two properties on ULC. A key recovery attack on ULC is presented in Section 5. Section 6 concludes the paper.

2. Notations

Suppose E is an n-bit block cipher of R rounds, the following notations are used throughout this paper.

(i) P: plaintext;

(ii) C: ciphertext;

(iii) K: 80-bit master key, K^t represents the 80-bit key register at the t-th round and $K_{[i:\ j]}$ represents the i-th bit to the j-th bit of K;

(iv) RK^i: 64-bit round key at the i-th round;

(v) "$\ll < i$": left rotation with i bits;

(vi) "$\oplus$": XOR operation.

3. Preliminary

3.1. General Procedures of a Slide Attack. In this section, the general procedures of a typical slide attack are illustrated. As is known to all, a block cipher uses a relatively weak round function iteratively to reach a strong security strength. According to the original slide attack, all the round functions of a block cipher E_k are supposed to be the same, which can be denoted as f_k. E_k can be written as

$$E_k = f_k \circ f_k \circ f_k \circ \cdots \circ f_k. \tag{1}$$

The key point for a slide attack is seeking a sliding pair. If two plaintexts P, P' satisfy the equation $P' = f_k(P)$, (P, P') is a slid pair. If a slid pair appears, for the two ciphertexts of P and P', $C = E_k(P)$, $C' = E_k(P')$, the conclusion $C' = f_k(C)$ can be derived. This procedure is demonstrated in Figure 1 and all the intermediate states in the same nibble marked in red lines are totally the same.

It is difficult for an attacker to figure out which pair is a slid pair. So, we have to construct a set containing sufficient plaintext-ciphertext pairs. According to the birthday paradox, if the block size of E_k is n and the set contains $2^{n/2}$ plaintext-ciphertext pairs, a slid pair is expected to appear with a high probability (about 63%). If a slide appears, as P, P', C, C' are all known, we can solve k from the following equation:

$$\begin{cases} f_k(P) = P', \\ f_k(C) = C'. \end{cases} \tag{2}$$

For some block ciphers, the round keys for different rounds are not always the same. According to the key schedule, we can use the condition of related keys to make f_{k_i}, have a dynamic match to construct a slid pair.

3.2. Related Key Attack. The related key attack model is a class of cryptanalytic attacks that correspond to the "single key attack model." In the related key attack model, the attacker can know or choose the relationship between several keys, and the target of the attacker is to find the correct secret key.

The related key attack model is important in the area of cryptanalysis. For slide attacks, the condition of "related key" usually appears, and the idea of using a related key in slide attack has been used in much previous literature. Besides, the security level of related key attacks is included as a rationale in many design specifications of block ciphers.

3.3. Brief Description on ULC. The structure of ULC is SPN. The idea of the ULC originated from the PRESENT and the RECTANGLE. The block size of ULC is 64 bits, and there are altogether 15 rounds. The round function is composed of three layers: the key addition layer, the substitution layer, and the bit permutation layer. In the key addition layer, the 64-bit round key is XORed into the internal state. In the substitution layer, a bit-sliced four-bit Sbox is used in parallel to introduce nonlinearity. In the bit permutation layer, an involutive 64-bit permutation is used to bring in quicker diffusion. The round function of ULC is illustrated in Figure 2.

The key schedule of ULC is modified from an ISO standard block cipher, PRESENT. First, the 80-bit register is initialized with the master key. We substitute the four most significant bits of the key register with the Sbox, then rotate the register by 61 bits on the left. Finally, the 64 most significant bits of the key register are extracted as the round key RK. The mathematical form of this key register updating progress can be illustrated as follows:

(1) $[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$

(2) $[k_{79}k_{78},\ldots,k_1k_0] = [k_{18}k_{17},\ldots,k_{20}k_{19}]$

(3) $\mathrm{RK}^i_{[63:\ 0]} = [k_{79}k_{78},\ldots,k_{17}k_{16}]$

4. Slide Properties on ULC

To better illustrate the slide attack on ULC, two properties are presented in this section. The first property is to illustrate the concrete conditions for a slid pair on ULC. The second property is to demonstrate the relationship between some subkey bits of the last round and some master key bits that will be used in our attack.

Property 1. (Sufficient prerequisite for a slid pair on ULC). For ULC, given two plaintext pairs (P, P') and a related key

Table 1: Summary of the attacks on ULC.

Cipher	Type of attack	Attacked/total rounds	Time complexity	Reference
ULC	Linear attack	10/15	$O(2^{76})$	[11]
	Slide attack	15/15	$O(2^{63})$	Section 5

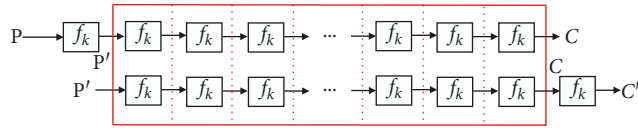

Figure 1: Encryption similarity for slide attack.

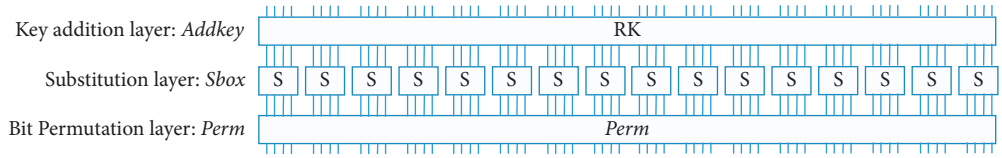

Figure 2: Round function of ULC.

(K, K'), $K' = K \lll 61$, $K'_{[79:76]} = \mathrm{Sbox}(K'_{[79:76]})$, $C = \mathrm{ULC}(P, K)$, $C' = \mathrm{ULC}(P', K')$, the sufficient prerequisite for a slid pair is as follows:

$$\mathrm{RK}^1 = \mathrm{Sbox}^{-1} \circ \mathrm{Perm}^{-1}(P') \oplus P. \tag{3}$$

If equation (3) is satisfied, the following equation (4) must also hold

$$\mathrm{RK}'^{16} = \mathrm{Perm}(\mathrm{Sbox}(C)) \oplus C'. \tag{4}$$

Proof. if we want to construct a slid pair for one round, the intermediate value after the first round must be equal to P'. This implies that $P' = \mathrm{Perm}(\mathrm{Sbox}(P \oplus \mathrm{RK}^1))$. Using the inverse operation of Perm and Sbox, equation (3) can be derived as

$$\mathrm{RK}^1 = \mathrm{Sbox}^{-1} \circ \mathrm{Perm}^{-1}(P') \oplus P. \tag{5}$$

Based on the condition of equation (3), P and P' must constitute a slid pair under the related key (K, K').

To make it more intuitive, this process is shown in Figure 3. As P' equals to the intermediate value after the first-round encryption of (P, K), the state for the key register of the first round, i.e., K^1, $K^1 = K \lll 61$, and $K^1_{[79:76]} =$, $\mathrm{Sbox}(K^1_{[79:76]})$ According to the relationship of the related key, K', equals to K^1. Therefore, the first 14 rounds for the encryption process of (P', K') (with a round key RK'^{15}, added after) are totally the same as 2–15 rounds encryption of (P, K) (the RK^{16}, is added after which is equals to RK'^{15}.). This similarity is illustrated in the red box in Figure 3.

Based on this observation, for the encryption of (P', K'), the intermediate value after adding the round key RK'^{15} must be C. This indicates the following equation must hold.

$$C' = \mathrm{Perm}(\mathrm{Sbox}(C)) \oplus \mathrm{RK}'^{16}. \tag{6}$$

□

Property 2. (Relationship of RK'^{16}, RK^1 and K). If a slid pair appears, given RK'^{16}, the round key bits $\mathrm{RK}^1_{[47:0]}$ and master key bits $K_{[79:19,2:0]}$ can be determined according to the following equation:

$$\begin{cases} K_{[(i+3):(j+3)]} = \mathrm{Sbox}^{-1}\left(\mathrm{RK}'^{16}_{[(i-16):(j-16)]}\right) & [i:j] \in \Omega_1, \\ K_{[(i+3)\bmod 80]} = \mathrm{RK}'^{16}_{[i-16]} & i \in [79,77] \text{or} [56,54] \text{or} [37,35] \text{or} [18,16]. \end{cases} \tag{7}$$

$$\begin{cases} \mathrm{RK}'^{16}_{[i+16]} = \mathrm{RK}^1_{[i]} & i \in [47,45] \text{or} [44,41] \text{or} [24,22] \text{or} [5,3], \\ \mathrm{Sbox}^{-1}\left(\mathrm{RK}'^{16}_{[(i+16):(j+16)]}\right) = \mathrm{RK}^1_{[i:j]} & [i:j] \in \Omega_2, \\ \mathrm{Sbox}^{-1}\left(\mathrm{RK}'^{16}_{[18:15]}\right)_{m3} = \mathrm{RK}^1_{[2:0]}. \end{cases} \tag{8}$$

$\Omega_1 = \{[34:31], [30:27], [26:23], [22:19], [53:50], [49:46], [45:42], [41:38], [76:73], [72:69], [68:65], [64:61], [60:57]\}$;

$\Omega_2 = \{[21:18], [40:37], [17:14], [36:33], [13:10], [32:29], [28:25], [9:6]$ $(a)^{m3}$.

represents the three most significant bits of a.

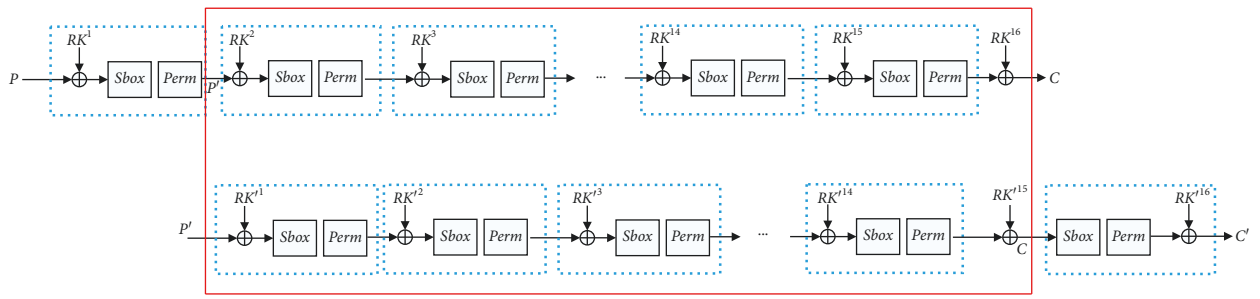

Figure 3: Encryption similarity of a slid pair.

Proof. The 64 round key bits of $RK'^{16,}$ are focused according to the key schedule. In this property, a link between the master key and round key bits for the first and last round is illustrated.

To better illustrate the relevance of these bits between different key registers, Figure 4 is presented to depict the path for these concerned key bits. In this figure, each row represents a state of the 80-bit key register at different rounds, and each nibble represents a bit for the key register. The colored nibbles represent the concerned round key bits at different rounds. The orange nibbles represent that these bits go through an Sbox. The rest of the concerned bits are marked in blue.

Through deduction, it is found that for the 64 bits of the round key RK'^{16}, 12 bits are totally the same from single-shift operation and the other 52 bits go through a single Sbox. Each four of the 52 bits go through the Sbox at rounds 1, 3–5, 7–9, 11–13, and 15-16 (these round numbers are defined according to the key K). The relationship between the master key bits and the bits going through each Sbox is calculated as follows:

$$\begin{cases} K^{1}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[79:\,76]}\right), \\ K^{3}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[37:\,34]}\right), \\ K^{4}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[56:\,53]}\right), \\ K^{5}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[75:\,72]}\right), \\ K^{7}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[33:\,30]}\right), \\ K^{8}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[52:\,49]}\right), \\ K^{9}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[71:\,68]}\right), \\ K^{11}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[29:\,26]}\right), \\ K^{12}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[48:\,45]}\right), \\ K^{13}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[67:\,64]}\right), \\ K^{15}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[25:\,22]}\right), \\ K^{16}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[44:\,41]}\right), \\ K'^{16}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[63:\,60]}\right). \end{cases} \tag{9}$$

The relationship between the key register K'^{16} and $K^{1,3,4,5,7,8,9,11,12,13,15,16}_{[60:\,57]}$ is as follows:

$$\begin{cases} K'^{16}_{[76:\,73]} = K^{1}_{[60:\,57]}, \\ K'^{16}_{[34:\,31]} = K^{3}_{[60:\,57]}, \\ K'^{16}_{[53:\,50]} = K^{4}_{[60:\,57]}, \\ K'^{16}_{[72:\,69]} = K^{5}_{[60:\,57]}, \\ K'^{16}_{[30:\,27]} = K^{7}_{[60:\,57]}, \\ K'^{16}_{[49:\,46]} = K^{8}_{[60:\,57]}, \\ K'^{16}_{[68:\,65]} = K^{9}_{[60:\,57]}, \\ K'^{16}_{[26:\,23]} = K^{11}_{[60:\,57]}, \\ K'^{16}_{[45:\,42]} = K^{12}_{[60:\,57]}, \\ K'^{16}_{[64:\,61]} = K^{13}_{[60:\,57]}, \\ K'^{16}_{[22:\,19]} = K^{15}_{[60:\,57]}, \\ K'^{16}_{[41:\,38]} = K^{16}_{[60:\,57]}, \\ K'^{16}_{[60:\,57]} = K'^{16}_{[60:\,57]}, \end{cases} \tag{10}$$

According to the relationship between the round key, related key, and key register, the following equations can be obtained.

$$\begin{cases} \mathrm{RK}'^{16}_{[60:\,57]} = K'^{16}_{[76:\,73]} = K^{1}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[79:\,76]}\right), \\ \mathrm{RK}'^{16}_{[18:\,15]} = K'^{16}_{[34:\,31]} = K^{3}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[37:\,34]}\right), \\ \mathrm{RK}'^{16}_{[37:\,34]} = K'^{16}_{[53:\,50]} = K^{4}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[56:\,53]}\right), \\ \mathrm{RK}'^{16}_{[56:\,53]} = K'^{16}_{[72:\,69]} = K^{5}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[75:\,72]}\right), \\ \mathrm{RK}'^{16}_{[14:\,11]} = K'^{16}_{[30:\,27]} = K^{7}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[33:\,30]}\right), \\ \mathrm{RK}'^{16}_{[33:\,30]} = K'^{16}_{[49:\,46]} = K^{8}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[52:\,49]}\right), \\ \mathrm{RK}'^{16}_{[52:\,49]} = K'^{16}_{[68:\,65]} = K^{9}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[71:\,68]}\right), \\ \mathrm{RK}'^{16}_{[1:\,7]} = K'^{16}_{[26:\,23]} = K^{11}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[29:\,26]}\right), \\ \mathrm{RK}'^{16}_{[29:\,26]} = K'^{16}_{[45:\,42]} = K^{12}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[48:\,45]}\right), \\ \mathrm{RK}'^{16}_{[48:\,45]} = K'^{16}_{[64:\,61]} = K^{13}_{[60:\,57]} = \mathit{Sbox}\left(K_{[67:\,64]}\right), \\ \mathrm{RK}'^{16}_{[6:\,3]} = K'^{16}_{[22:\,19]} = K^{15}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[25:\,22]}\right), \\ \mathrm{RK}'^{16}_{[25:\,22]} = K'^{16}_{[41:\,38]} = K^{16}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[44:\,41]}\right), \\ \mathrm{RK}'^{16}_{[44:\,41]} = K'^{16}_{[60:\,57]} = K'^{16}_{[60:\,57]} = \mathrm{Sbox}\left(K_{[63:\,60]}\right). \end{cases} \tag{11}$$

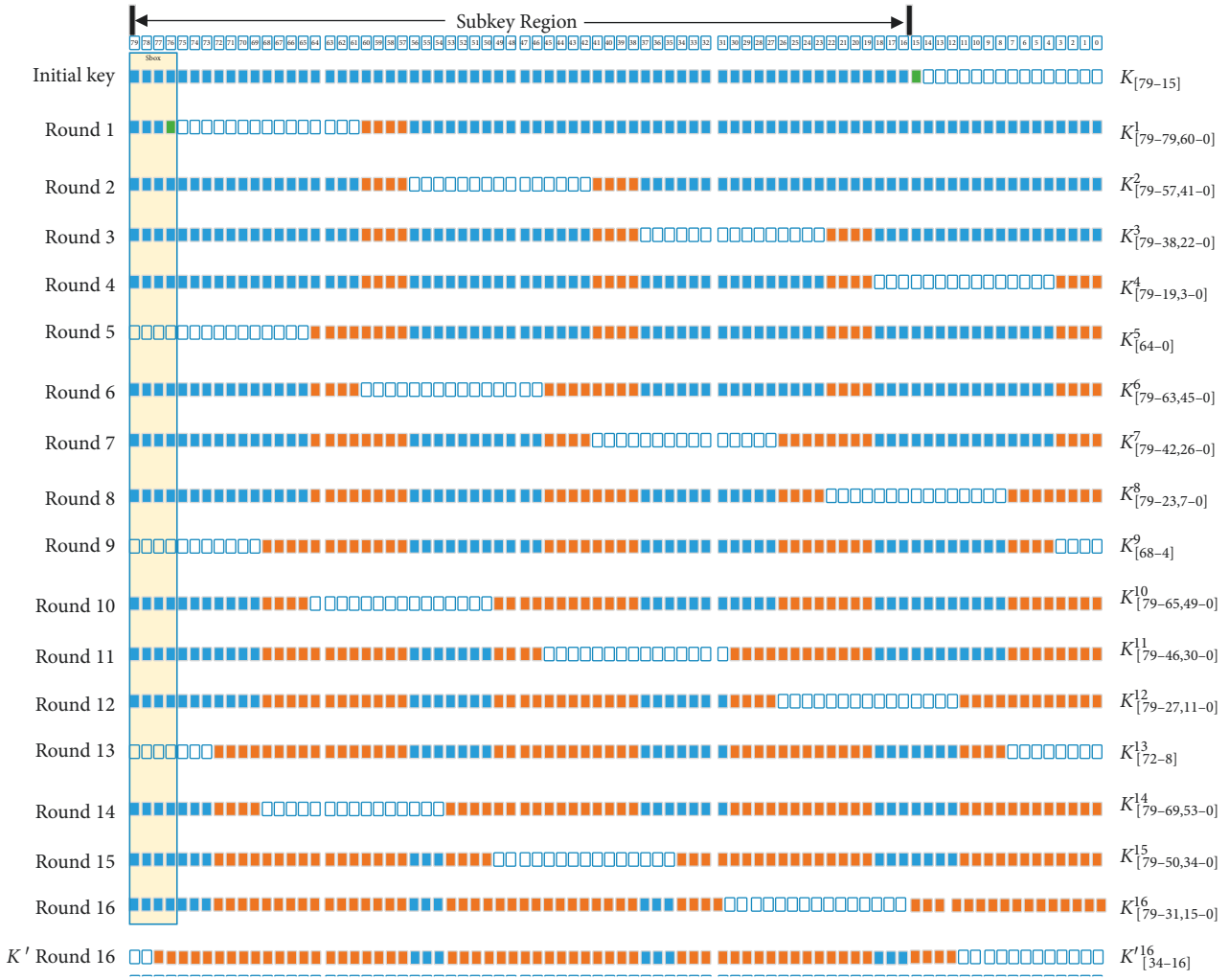

Figure 4: Relationship between some master key bits and round key bits.

Input: randomly choose an 80-bit master key K;
Output: the recovered 80-bit master key.
Preliminary:
randomly choose 2^{32} plaintext-ciphertext pairs (P_i, C_i) (encrypted with K) to construct a set δ;
set related key $K' = K \lll 61$, $K'_{[79:\,76]} = \text{Sbox}(K_{[79:\,76]})$.
Step 1. Choose two plaintext-ciphertext pairs (P_i, C_i) and (P_j, C_j) from δ;
Step 2. Calculate $C'_i = \text{ULC}(P_i, K')$, $C'_j = \text{ULC}(P_j, K')$;
Step 3. Suppose (P_i, P_j) is a slid pair. Calculate $\text{RK}'^{16} = \text{Perm}(\text{Sbox}(C_i)) \oplus C'_j$, $\text{RK}^1 = \text{Sbox}^{-1} \circ \text{Perm}^{-1}(P_j) \oplus P_i$. Test whether the relationship of RK'^{16} and RK^1 satisfies the equation (4) in Property 2.
(a) If the relationship of RK'^{16} and RK^1 satisfies the condition, recover 80-bit K'' according to Property 2, and go to Step 5.
(b) Otherwise, go to Step 1.
Step 4. Suppose (P_j, P_i) is a slid pair. Calculate $\text{RK}'^{16} = \text{Perm}(\text{Sbox}(C_j)) \oplus C'_i$, $\text{RK}^1 = \text{Sbox}^{-1} \circ \text{Perm}^{-1}(P_i) \oplus P_j$. Test whether the relationship of RK'^{16} and RK^1 satisfies the equation (5) in Property 2.
(a) If the relationship of RK'^{16} and RK^1 satisfies the condition, recover 80-bit K'' according to Property 2 and go to Step 5.
(b) Otherwise, go to Step1.
Step 5. Encrypt P_i with K'' and ciphertext C'_i can be derived.
(a) If $C_i = C'_i$, K'' is the correct master key and output K'';
(b) Otherwise, go to Step 1.

Algorithm 1: Key recovery attack on ULC.

After applying the Sbox^{-1} on both two sides of the equations, the conclusion of equation (7) in Property 2 can be derived (those bits from only shift operation can be calculated directly).

$$\begin{cases} \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[60:\,57]}\right) = K_{[79:\,76]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[18:\,15]}\right) = K_{[37:\,34]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[37:\,34]}\right) = K_{[56:\,53]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[56:\,53]}\right) = K_{[75:\,72]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[14:\,11]}\right) = K_{[33:\,30]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[33:\,30]}\right) = K_{[52:\,49]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[52:\,49]}\right) = K_{[71:\,68]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[10:\,7]}\right) = K_{[29:\,26]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[29:\,26]}\right) = K_{[48:\,45]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[48:\,45]}\right) = K_{[67:\,64]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[6:\,3]}\right) = K_{[25:\,22]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[25:\,22]}\right) = K_{[44:\,41]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[44:\,41]}\right) = K_{[63:\,60]}. \end{cases} \tag{12}$$

As the round key bits of $\text{RK}^1_{[47:\,0]}$ and K have the following relations:

$$\begin{cases} \text{RK}^1_{[44:\,41]} = \text{Sbox}\left(K_{[79:\,76]}\right), \\ \text{RK}^1_{[47:\,45]} = K_{[2:\,0]}, \\ \text{RK}^1_{[40:\,0]} = K_{[75:\,35]}. \end{cases} \tag{13}$$

So the round key bits of $RK^1_{[47:\,0]}$ and K'^{16} have the following relation, and the conclusion of (7) in Property 2 can be derived.

$$\begin{cases} \text{RK}'^{16}_{[60:\,57]} = Sbox\left(K_{[79:\,76]}\right) = \text{RK}^1_{[44:\,41]}, \\ \text{RK}'^{16}_{[63:\,61]} = K_{[2:\,0]} = \text{RK}^1_{[47:\,45]}, \\ \text{RK}'^{16}_{[40:\,38]} = K_{[59:\,57]} = \text{RK}^1_{[24:\,22]}, \\ \text{RK}'^{16}_{[21:\,19]} = K_{[40:\,38]} = \text{RK}^1_{[5:\,3]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[18:\,15]}\right) = K_{[37:\,34]} \longrightarrow \text{RK}^1_{[2:\,0]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[37:\,34]}\right) = K_{[56:\,53]} = RK^1_{[21:\,18]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[56:\,53]}\right) = K_{[75:\,72]} = \text{RK}^1_{[40:\,37]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[33:\,30]}\right) = K_{[52:\,49]} = \text{RK}^1_{[17:\,14]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[52:\,49]}\right) = K_{[71:\,68]} = \text{RK}^1_{[36:\,33]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[29:\,26]}\right) = K_{[48:\,45]} = \text{RK}^1_{[13:\,10]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[48:\,45]}\right) = K_{[67:\,64]} = \text{RK}^1_{[32:\,29]}, \\ \text{Sbox}^{-1}\left(\text{RK}'^{16}_{[44:\,41]}\right) = K_{[63:\,60]} = \text{RK}^1_{[28:\,25]}, \\ \text{Sbox}^{-1}\left(RK'^{16}_{[25:\,22]}\right) = K_{[44:\,41]} = \text{RK}^1_{[9:\,6]}. \end{cases} \tag{14}$$

There are two ways to further utilize Property 2:

(1) If a slid pair appears, given RK^{16} and RK^1, all the 80 master key bits of K can be recovered.

(2) If a slid pair appears, 48 bits of RK^1 can be determined according to RK'^{16}. This can be viewed as a distinguisher to eliminate wrong slid pair candidates. □

5. Key Recovery Attack on ULC

In this section, a key recovery attack on ULC is proposed based on the properties in Section 4. With our method, the full-round ULC can be attacked and all the 80-bit master keys can be recovered. The details of the attack are illustrated in Algorithm 1.

The target of Algorithm 1 is to recover the master key K. There are two preconditions. The first one is a set δ consisting of 2^{32} plaintext-ciphertext pairs encrypted with the master key K and the second one is a related key K'. In our attack, we regard the chosen two plaintexts P_i and P_j as two potential slid pairs with each other, i.e., both (P_i, P_j) and (P_j, P_i) are potential slid pairs. This strategy will improve the search efficiency by about two times faster.

The general procedures of the attack are as follows: first, we choose two plaintexts from the set δ, and regard these two plaintexts as two potential slid pairs. We use the two properties in Section 4 to eliminate those nonslid pairs and recover the 80-bit master key as a key candidate. Then we test the remaining key candidates through a plaintext-ciphertext pair to guarantee the correctness of the recovered master key. To sum up, the complexities of our attack are summarized in Table 2.

5.1. Complexity Analysis. With Algorithm 1, we can recover all the 80 master key bits. As the slide attack has no constraint on the number of rounds, this implies that our attack can be applied to arbitrary rounds. In this paper, as ULC has a total of 15 rounds, we set the number of our attacks to be 15 as well.

As our attack needs 2^{32} plaintext-ciphertext pairs, it means the data complexity is $O(2^{32})$ and memory complexity is also $O(2^{32})$, which is used to store these pairs. According to the birthday paradox, the success probability of the attack is about 63%. There are altogether $2^{32} * (2^{32}-1)/2 \approx 2^{63}$ different pairs of (P_i, P_j) originated from δ. So, the time complexity for our attack is $O(2^{63})$.

In addition, if the slid pair does not occur, RK^1 and RK'^{16} can be viewed as random. This implies that if we use (5) of Property 2 as a distinguisher and the slid pair does not occur, the probability of a wrong key candidate to pass the test of the distinguisher is about 2^{-48}. This means we still need an extra plaintext-ciphertext pair to eliminate all the possible wrong key candidates, and this is realized by Step 5 of Algorithm 1.

Table 2: Complexities and summary of our attack.

Block cipher	Rounds attacked/total rounds	Data complexity	Memory complexity	Time complexity	Number of related-keys
ULC	15/15	$O(2^{32})$	$O(2^{32})$	$O(2^{63})$	1

6. Conclusion

In this paper, a related key slide attack on full-round ULC is proposed. As a first step, a property on ULC is presented to characterize a slid pair on ULC. As a second step, a relationship between some of the first round, last round, and master key bits is constructed. Finally, we propose a key recovery attack based on these two properties. Our attack can attack a full-round cipher and recover all the 80 master key bits. For improving the security of ULC, on the one hand, the security margin for ULC is too small, the number of total rounds should be increased. On the other hand, to prevent ULC from slide attack, at least different constants should be added for each round. As related keys are a relatively strong condition for cryptanalysis, other better single key attacks can be explored, which is left as future work.

Data Availability

All the data included in this study are available upon request by contacting 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 Grant 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," in *Proceedings of the International workshop on cryptographic hardware and embedded systems*, pp. 450–466, Springer, Vienna, Austria, September 2007.

[2] W. Wu and L. Zhang, "LBlock: a lightweight block cipher," in *Proceedings of the International conference on applied cryptography and network security*, pp. 327–344, Springer, Nerja, Spain, June 2011.

[3] 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.

[4] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: a lightweight block cipher for multiple platforms,"vol. 22, pp. 339–354, in *Proceedings of the International Conference on Selected Areas in Cryptography*, vol. 22, pp. 339–354, Springer, Windsor, ON, Canada, August 2012.

[5] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: an ultra-lightweight blockcipher," in *Proceedings of the International workshop on cryptographic hardware and embedded systems*, pp. 342–357, Springer, Nara, Japan, September 2011.

[6] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the Proceedings of the 52nd Annual Design Automation Conference*, pp. 1–6, ACM, San Francisco California, June 2015.

[7] 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 Transactions on Computers*, vol. 65, no. 7, pp. 2074–2089, 2015.

[8] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The simeck family of lightweight block ciphers," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 307–329, Springer, Berlin, Heidelberg, 2015.

[9] D. Hong, J. Sung, S. Hong et al., "HIGHT: a new block cipher suitable for low-resource device," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 46–59, Springer, Yokohama, Japan, October 2006.

[10] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "LEA: a 128-bit block cipher for fast encryption on common processors,"vol. 8267, pp. 3–27, in *Proceedings of the International Workshop on Information Security Applications*, vol. 8267, pp. 3–27, Springer, Jeju Island, Korea, August 2013.

[11] L. Sliman, T. Omrani, Z. Tari, A. E. Samhat, and R. Rhouma, "Towards an ultra lightweight block ciphers for Internet of Things," *Journal of Information Security and Applications*, vol. 61, Article ID 102897, 2021.

[12] A. Biryukov and D. Wagner, "Slide attacks," in *Proceedings of the International Workshop on Fast Software Encryption*, pp. 245–259, Springer, Berlin, Heidelberg, March 1999.

[13] A. Bar-On, E. Biham, O. Dunkelman, and N. Keller, "Efficient slide attacks," *Journal of Cryptology*, vol. 31, no. 3, pp. 641–670, 2018.

[14] O. Dunkelman, N. Keller, N. Lasry, and A. Shamir, "New slide attacks on almost self-similar ciphers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 250–279, Springer, Zagreb, Croatia, May 2020.

[15] X. Bonnetain, M. Naya-Plasencia, and A. Schrottenloher, "On quantum slide attacks,"vol. 11959, pp. 492–519, in *Proceedings of the International Conference on Selected Areas in Cryptography*, vol. 11959, pp. 492–519, Springer, Waterloo, ON, Canada, August 2019.

[16] X. Dong, B. Dong, and X. Wang, "Quantum attacks on some Feistel block ciphers," *Designs, Codes and Cryptography*, vol. 88, no. 6, pp. 1179–1203, 2020.

[17] A. Mege, "Slide attack on CLX-128," in *Proceedings of the Lightweight Cryptography Workshop*, p. 169p. 169, 2019.

[18] S. Kavut and M. D. Yücel, "Slide attack on Spectr-H64," in *Proceedings of the International Conference on Cryptology in India*, pp. 34–47, Springer, Berlin, Heidelberg, December 2002.

[19] R. Mbitiru and T. S. Ustun, "Using input-output correlations and a modified slide attack to compromise IEC 62055-41," in

Proceedings of the 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), pp. 1–6, IEEE, Ixtapa, Mexico, November 2017.

[20] L. Lu and S. Chen, "A compress slide attack on the full GOST block cipher," *Information Processing Letters*, vol. 113, no. 17, pp. 634–639, 2013.

[21] A. Baksi, S. Maitra, and S. Sarkar, *An Improved Slide Attack on Trivium*, pp. 351–375, IPSI Transaction on Internet Research, France, Paris, 2015.

[22] M. Gorski, S. Lucks, and T. Peyrin, "Slide attacks on a class of hash functions," in *ASIACRYPT 2008. LNCS*, J. Pieprzyk, Ed., vol. 5350, pp. 143–160, Springer, Heidelberg, 2008.

[23] F. B. Durak and S. Vaudenay, "Breaking the FF3 format-preserving encryption standard over small domains," in *CRYPTO 2017. LNCS*, J. Katz and H. Shacham, Eds., Springer, New York, NY, US, pp. 679–707, 2017.

[24] L. Ding, D. Gu, L. Wang, C. Jin, and J. Guan, "A real-time related key attack on the WG-16 stream cipher for securing 4G-LTE networks," *Journal of Information Security and Applications*, vol. 63, Article ID 103015, 2021.

[25] M. Dworkin, "Recommendation for block cipher modes of operation: methods for format preserving encryption," *National Institute of Standards and Technology*, vol. 28, p. 1, 2016.