Related-Key Differential Attacks on CHESS-64

Wei Luo, Jiansheng Guo
Zhengzhou Information Science and Technology Institute, Zhengzhou 450004, Henan, China
[e-mail: guojs2013@gmail.com]
*Corresponding author: Jiansheng Guo

Received March 6, 2014; revised May 22, 2014; revised July 10, 2014; accepted August 9, 2014; published September 30, 2014

Abstract

With limited computing and storage resources, many network applications of encryption algorithms require low power devices and fast computing components. CHESS-64 is designed by employing simple key scheduling and Data-Dependent operations (DDO) as main cryptographic components. Hardware performance for Field Programmable Gate Arrays (FPGA) and for Application Specific Integrated Circuits (ASIC) proves that CHESS-64 is a very flexible and powerful new cipher. In this paper, the security of CHESS-64 block cipher under related-key differential cryptanalysis is studied. Based on the differential properties of DDOs, we construct two types of related-key differential characteristics with one-bit difference in the master key. To recover 74 bits key, two key recovery algorithms are proposed based on the two types of related-key differential characteristics, and the corresponding data complexity is about $2^{42.9}$ chosen-plaintexts, computing complexity is about $2^{42.9}$ CHESS-64 encryptions, storage complexity is about $2^{26.6}$ bits of storage resources. To break the cipher, an exhaustive attack is implemented to recover the rest 54 bits key. These works demonstrate an effective and general way to attack DDO-based ciphers.

Keywords: Cryptanalysis, CHESS-64 block cipher, related-key differential attack, Data-Dependent operations

http://dx.doi.org/10.3837/tiis.2014.09.018

1. Introduction

Security and privacy are primary requirements for wired and wireless communication. As a most common method, encryption is used to provide secure and secret communication. In the field of ubiquitous computing systems [1], sensor networks [2], wireless networks [3], IPsec [4] and mobile communication [5], limited computing and storage resources bring a variety of privacy and security challenges for encryption algorithms. As a result, more efficient cryptographic primitives are badly needed to provide high performance on resource-constrained devices.

In the past decade, for encryption applications requiring a fast hardware implementation with limited computing and storage resources, Data-Dependent permutations (DDPs) [6] have been used as main cryptographic primitives in a number of fast block ciphers, namely Spectr-H64 [7], Cobra-H64/128 [8], CIKS-128H [9], DDP-64 [10] and so on. As a linear primitive, DDP conserves weights of transformed bit string, and DDP-based ciphers show their natural weaknesses against differential cryptanalytic attacks. In 2004, [11] proposed related-key differential attacks on full-round CIKS-128 and CIKS-128H. In 2005, Cobra-H64 and Cobra-H128 were proved to be insecure under related-key differential attacks [12]. It was proved that DDP-64 do not have a high security level as the designer promised [13].

To strengthen the security of DDP-based ciphers, more powerful cryptographic primitive, namely Data-Dependent operations (DDOs) are introduced to design block ciphers [14,15,16,17,18] implemented on resource-constrained devices. In order to achieve higher speed and cost fewer computing and storage resources, these ciphers usually use simple key schedule. For DDO-based ciphers, there is no general cryptanalysis method as DDP-based ciphers, and security problem turns out to be a stumbling block for the application of high speed DDO-based ciphers. Consequently, the security evaluation gradually makes a significant task for the application of DDO-based ciphers on resource-constrained devices.

As an example of DDO-based cipher, CHESS-64 was designed to achieve more efficient hardware implementations than any existing DDP-based ciphers. In 2009, Lee et al proposed a related-key differential attack on CHESS-64 by constructing a related-key differential characteristic with high probability [19]. In our work, we point out some flaws in Lee et al's work on constructing related-key differential characteristic and recovering key. As a result, Lee et al's attack won't work as promised.

Further, in this paper, we construct two types of related-key differential characteristics with one-bit difference in the master key, based on which, two key recovery algorithms are proposed. Specifically, the first key recovery algorithm could recover 42 bits of the master key with about $2^{42.4}$ chosen-plaintexts, $2^{42.4}$ CHESS-64 encryptions and $2^{12.2}$ bits of storage resources, while the second key recovery algorithm could recover another 32 bits of the master key requiring about $2^{41.1}$ chosen-plaintexts, 2^{41} CHESS-64 encryptions and $2^{26.6}$ bits of storage resources. To break CHESS-64, we perform an exhaustive search to recover the rest 54 bits of the master key. We firstly proposed correct cryptanalytic results on CHESS-64 so far, and we present a new and common method to analyze the security of DDO-based ciphers. We summarize our results and existing cryptanalytic results on some typical DDP-based and DDO-based ciphers in **Table 1**.

Table 1. Cryptanalytic Results of some typical DDP-based and DDO-based ciphers

Block Cipher	Number of Round	Data/Time Complexity	Recovered Key Bits	Comment
CHESS-64	8(full)	$2^{42.9}/2^{42.9}$	74	This paper
	8(full)	$2^{42.9}/2^{54}$	128(full)	
CIKS-128	12(full)	$2^{44}/2^{44}$	63	[11]
CIKS-128H	8(full)	$2^{48}/2^{48}$	63	
Cobra-H64	10(full)	$2^{15.5}/2^{15.5}$	23	[12]
Cobra-H128	10(full)	$2^{44}/2^{44}$	63	
DDP-64	10(full)	$2^{54}/2^{54}$	22	[13]
MD-64	8(full)	$2^{95}/2^{43.1}$	128(full)	[20]

Data: Related-Key Chosen Plaintexts, Time: Encryption Units

Outline. This paper is organized as follows. In Section 2, we firstly give some notations, and then we briefly describe the structure of CHESS-64. In Section 3, we study the related-key differential properties of CHESS-64, construct two types of related-key differential characteristics, and point out some flaws in existing cryptanalytic results on CHESS-64. In Section 4, two key recovery algorithms are presented on CHESS-64. Finally, we conclude in Section 5.

2. Description of CHESS-64

In this section, we firstly present some notations in this paper, and then we give a brief introduction of the particular DDOs used in CHESS-64 and the structure of CHESS-64.

2.1 Notations

We use the following notations in this paper. Note that a bit string will be numbered from left to right, starting with bit 1. For example, for $L=(l_1,l_2,\cdots,l_n)$, l_1 is the left most bit and l_n is the right most bit.

- e_i : a binary string e in which the i-th bit is one and the others are zeros;
- $D(i)$: the i-th bit of a 32-bit string, namely D;
- $F_{n/m}$: a DDO with m bits as controlling binary string, and n bits as input and output, respectively;
- $>>>16$: a 16-bit right cyclic rotation;
- X, Y : a 64-bit plain-text and a 64-bit cipher-text, respectively;
- X^i, Y^i : the input and output of the i-th round of CHESS-64.

2.2 DDOs used in CHESS-64

CHESS-64 employs three DDOs as its nonlinear operations which are depicted in **Fig. 1**.

As shown in **Fig. 1-(a)** and **Fig. 1-(b)**, due to the symmetric structure, $F_{32/96}$ and $F_{32/96}^{-1}$ differ only in the distribution of controlling bits over the basic building block $F_{2/1}$, which is defined as follows.

$$F_{2/1}(x_1,x_2,v)=\begin{cases}(x_2,x_1) & if \quad v=0;\\ (x_1,x_1\oplus x_2) & if \quad v=1.\end{cases}$$

While the basic building block $F_{2/1}$' used in $F_{32/80}$' is defined as follows.

$$F_{2/1}' = \begin{cases} (x_2 \oplus 1, x_1 \oplus 1) & if \quad v = 0; \\ (x_1 \oplus x_2 \oplus 1, x_2) & if \quad v = 1. \end{cases}$$

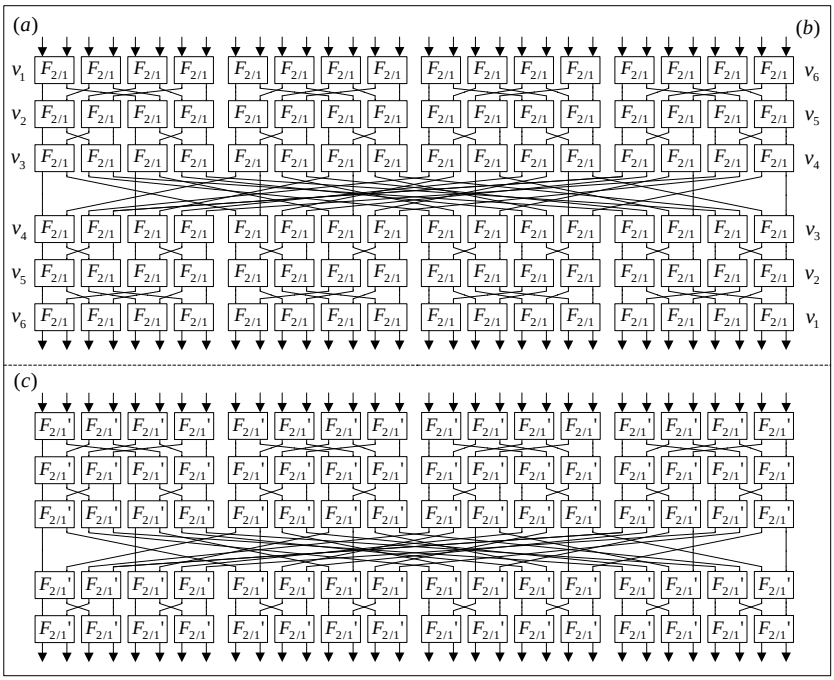


Fig. 1. (a) $F_{32/96}$, (b) $F^{-1}{}_{32/96}$, (c) $F_{32/80}$'

2.3 Structure of CHESS-64

CHESS-64 is a pure DDO-based cipher which only employs DDOs and other linear operations. Composed of initial transformation (IT), round function *Crypt* , and final transformation (FT), CHESS-64 is an 8-round iterated block cipher with a block size of 64 bits and 128 bits master key. The cipher's general structure and round function are shown in **Fig. 2-(a)** and **Fig. 2-(b)**, respectively.

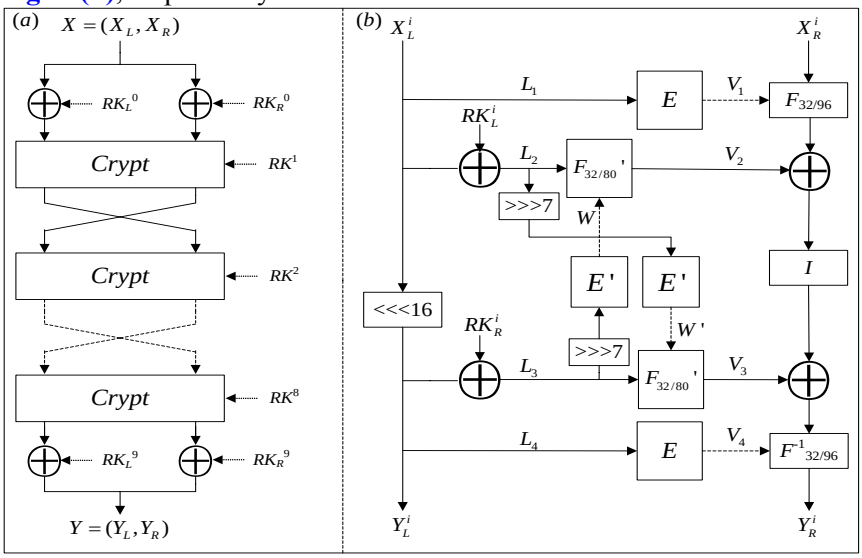


Fig. 2. (a) Structure of CHESS-64, (b) Round function *Crypt*

As shown in **Fig. 2-(a)**, the cipher uses two 64-bit key blocks RK^0 and RK^9 to transform the input data and output data, respectively. For the former 7 rounds, two 32-bit output blocks would be swapped for the input of the next round, but not for the last round .

As shown in **Fig. 2-(b)**, round function *Crypt* is composed of five kinds of operations: bitwise module 2 addition, two extend-functions (E, E'), three DDOs ($F_{32/96}, F_{32/96}^{-1}, F_{32/80}'$), two cyclic rotations ($<<<16, >>>7$), and an involution (I).

Details of transformation components of round function *Crypt* could be obtained in [8].

To obtain a high speed performance, CHESS-64 employs a very simple key schedule. The 128-bit master key K of CHESS-64 is divided into four 32-bit key blocks K_1, K_2, K_3, K_4, and round keys are presented in **Table 2**, where $K_m (m = 1,2,3,4)$ donates a 32-bit key block, and $RK^i (i = 1,2,\cdots,8)$ donates round key of the i-th round.

Table 2. Key schedule of CHESS-64

Round i	0	1	2	3	4	5	6	7	8	9
RK^i	(K_4,K_3)	(K_3,K_1)	(K_2,K_4)	(K_4,K_2)	(K_1,K_3)	(K_4,K_2)	(K_1,K_3)	(K_2,K_1)	(K_3,K_4)	(K_1,K_2)

Encryption procedure of CHESS-64 is presented in the following table.

Encryption Algorithm of CHESS-64
[Step 1] An input block X is divided into two subblocks X_L and X_R.
[Step 2] Perform IT: $X_L^1 = X_L \oplus RK_L^0$ and $X_R^1 = X_R \oplus RK_R^0$.
[Step 3] For $i = 1$ to 7 do: . $(Y_L^i, Y_R^i) = Crypt(X_L^i, X_R^i, RK^i), (X_L^{i+1}, X_R^{i+1}) = (Y_R^i, Y_L^i)$
[Step 4] $(Y_L^8, Y_R^8) = Crypt(X_L^8, X_R^8, RK^8)$.
[Step 5] Perform FT: $Y_L = Y_L^8 \oplus RK_L^9$ and $Y_R = Y_R^8 \oplus RK_R^9$.
[Step 6] Return the ciphertext block $Y = (Y_L, Y_R)$.

3 Properties for Components of CHESS-64

In this section, we firstly describe some differential properties for the basic building blocks of DDOs, which allow us to analyze differential properties for DDOs in the next step. And then, by adding one-bit difference in the master key, we construct three kinds of one-round related-key differential characteristics with high probability. Finally, two types of full-round related-key differential characteristics are constructed by employing properties of DDOs and one-round related-key differential characteristics.

3.1 Differential properties for the basic building blocks

As components of round function *Crypt*, $F_{32/96}(F_{32/96}^{-1}), F_{32/80}'$ employ $F_{2/1}, F_{2/1}'$ as basic building blocks, respectively.

The following properties hold for $F_{2/1}$.

Property 1. For $F_{2/1}$, if the difference weight of controlling string is 0, and the difference weight of input is 1, the output difference is

$$\Delta Y = \begin{cases} 01 & if \quad \Delta X = 10, v = 0 \, or \, \Delta X = 01, v = 1; \\ 11 & if \quad \Delta X = 10, v = 1; \\ 10 & if \quad \Delta X = 01, v = 0. \end{cases}$$

Property 2. For $F_{2/1}$, if the difference weight of controlling string is 0, and the difference weight of input is 2, the output difference is

$$\Delta Y = \begin{cases} 11 & if \quad v = 0; \\ 10 & if \quad v = 1. \end{cases}$$

Property 3. For $F_{2/1}$, if the difference weight of controlling string is 1, and the difference weight of input is 0, the output difference is

$$\Delta Y = \begin{cases} 00 & if \quad X = 00; \\ 10 & if \quad X = 10; \\ 11 & if \quad X = 01; \\ 01 & if \quad X = 11. \end{cases}$$

Analogously, the following properties hold for $F_{2/1}'$.

Property 4. For $F_{2/1}'$, if the difference weight of controlling string is 0, and the difference weight of input is 1, the output difference is

$$\Delta Y' = \begin{cases} 01 & if \quad \Delta X' = 10, v' = 0; \\ 10 & if \quad \Delta X' = 10, v' = 1 \, or \, \Delta X' = 01, v' = 0; \\ 11 & if \quad \Delta X' = 01, v' = 1. \end{cases}$$

Property 5. For $F_{2/1}'$, if the difference weight of controlling string is 0, and the difference weight of input is 2, the output difference is

$$\Delta Y' = \begin{cases} 11 & if \quad v' = 0; \\ 01 & if \quad v' = 1. \end{cases}$$

Property 6. For $F_{2/1}'$, if the difference weight of controlling string is 1, and the difference weight of input is 0, the output difference is

$$\Delta Y' = \begin{cases} 01 & if \quad X' = 00; \\ 10 & if \quad X' = 10; \\ 00 & if \quad X' = 01; \\ 11 & if \quad X' = 11. \end{cases}$$

3.2 Differential properties for DDOs

As depicted in **Fig. 1**, DDOs are constructed with basic building blocks in layered topology. Specifically, $F_{32/96}^{-1}$ is constructed with 6 layers of $F_{2/1}$, and each layer consists of 16 $F_{2/1}$ in

alignment; while $F_{32/80}'$ is constructed with 5 layers of $F_{2/1}'$, and each layer consists of 16 $F_{2/1}'$ in alignment. Base on the differential properties of basic building blocks, we present differential properties of $F_{32/96}^{-1}, F_{32/80}'$ when the difference weight of input is 1.

Let the input difference of $F_{32/96}^{-1}(F_{32/80}')$ be $e_k (k=1,2,\cdots 32)$. By Property 1 and Property 2 (Property 4 and Property 5), we can accurately obtain the probability distribution of the output differences by analyzing difference transmission properties layer by layer. For example, if the input difference of $F_{32/96}^{-1}$ is e_{16}, the probability distribution of the output differences of $F_{32/96}^{-1}$ is depicted in **Appendix Table 1**.

Theorem 1. For $F_{32/96}^{-1}$, if the difference weight of input is 1, there are two differential routes at most that could transmit input difference $e_k (k=1,2,\cdots 32)$ to any output difference.

Proof. According to the topology of $F_{32/96}^{-1}$, when the difference weight of input and output is 1, there exist no more than 2 one-bit differential routes. Consequently, there exist 2 differential paths at most which could transmit e_k to nonzero bit of any output difference. In other words, there are two differential routes at most which could transmit e_k to any output difference.

As shown in **Fig. 3,** for $F_{32/96}^{-1}$, the bold lines donate the two possible difference routes when the input difference is e_{31} and the output difference is e_{25}. By Property 1, for the first route (See **Fig. 3-(a)**), we can exactly know that the six-bit of controlling string from top to bottom is "011000". By Property 1 and Property 2, for the second route (See **Fig. 3-(b)**), the ten-bit of controlling string from top to bottom and from left to right is "1111100001".

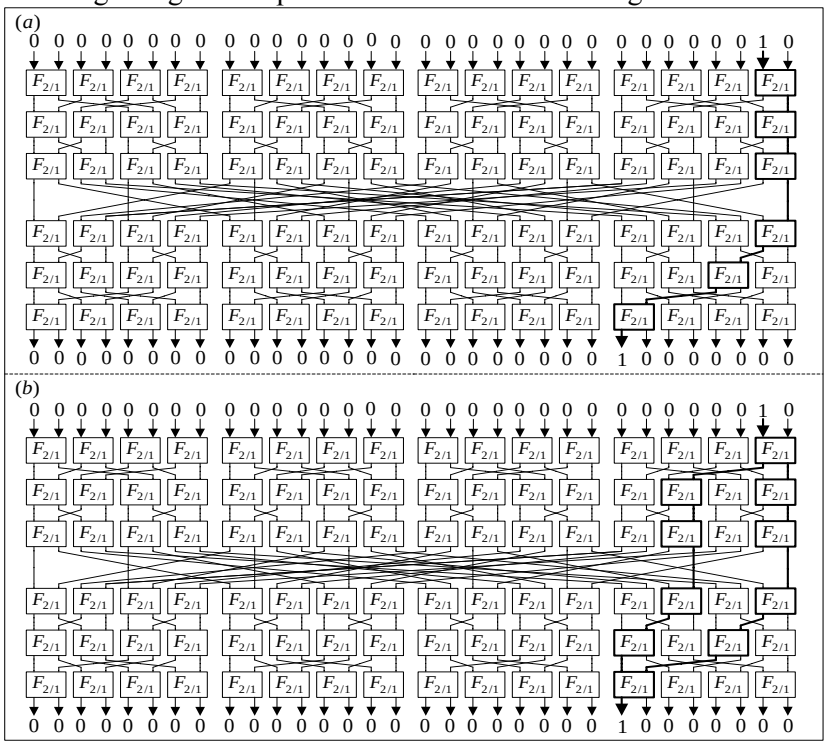


Fig. 3. (a) The first difference route of $e_{31} \rightarrow e_{25}$ for $F_{32/96}^{-1}$,
(b) The second difference route of $e_{31} \rightarrow e_{25}$ for $F_{32/96}^{-1}$

Based on our analysis above, it's clear that there are two flaws in [9], which are as follows.

(1) It's impossible to obtain the exact six-bit of controlling string with one-bit input and output difference for $F_{32/96}^{-1}$.

For example, as there are two possible difference routes for $e_{31} \to e_{25}$ (See **Fig. 3**), it's impossible to know which one is correct with a certain cipher-text pair.

(2) Lee et al made a mistake in calculating probability of one-bit differential route in $F_{32/96}^{-1}$.

For example, for $F_{32/96}^{-1}$ with an input difference e_{31}, the output difference is e_{25} with probability $2^{-6}+2^{-10}$ (not 2^{-6}, as Lee et al presented in the last paragraph of sub-section 4.1).

Theorem 2. For $F_{32/80}'$, if the difference weight of input is 1, we can exactly obtain one differential route according to input difference $e_k (k=1,2,\cdots 32)$ and output difference.

Proof. According to the topology of $F_{32/80}'$, when the difference weight of input and output is 1, there exactly exists one one-bit differential route. In other words, there exists one differential route that could transmit e_k to nonzero bit of any output difference. Consequently, we can exactly obtain one difference route according to input difference $e_k (k=1,2,\cdots 32)$ and output difference.

As shown in **Fig. 4**, for $F_{32/80}'$, the bold line donates the certain difference route when the input difference is e_{17} and the output difference is e_{25}. According to Property 4, we can exactly know the five-bit of controlling string from top to bottom is "10010".

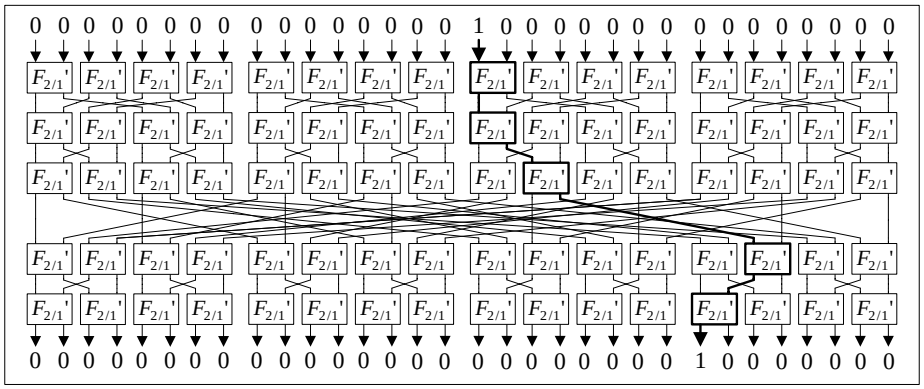


Fig. 4. Difference route of $e_{17} \to e_{25}$ for $F_{32/80}'$

Property 7. For $F_{32/96}^{-1}(F_{32/80}')$, if the difference weight of controlling string is 0 and the difference weight of input is nonzero, the difference weight of output is nonzero.

Proof. By Property 1 and Property 2 (Property 4 and Property 5), for $F_{2/1}(F_{2/1}')$, when the difference weight of controlling string is 0 and the difference weight of input is nonzero, the difference weight of output is nonzero. As $F_{32/96}^{-1}(F_{32/80}')$ are constructed with $F_{2/1}(F_{2/1}')$ in layered topology, we can certainly know that the difference weight of output is nonzero for $F_{32/96}^{-1}(F_{32/80}')$ by analyzing differential properties layer by layer.

3.3 Related-key differential characteristics

With chosen-plain-texts, by adding one-bit difference in the master key, Lee et al's tried to construct one-round related-key differential characteristics and full-round related-key differential characteristics further [9]. However, in this paper, we show that there are two flaws in Lee et al's work on constructing one-round related-key differential characteristics. And by using the properties and theorems in the previous sub-sections, three correct one-round related-key differential characteristics are constructed with one-bit difference in the master key.

When $\Delta K_3 = e_{17}$, by **Table 2**, sub-key differences of every round (ΔRK^i) are as follows.

$$\Delta RK^i = \begin{cases} (e_{17},0) & for \quad i = 1,8; \\ (0,e_{17}) & for \quad i = 4,6; \\ (0,0) & for \quad i = 2,3,5,7. \end{cases}$$

According to the three kinds of sub-key differences, three kinds of corresponding one-round related-key differential characteristics are constructed as follows.

Case1 $\Delta RK^i = (e_{17},0)$

Since $\Delta RK^i = (e_{17},0)$, input difference of the first $F_{32/80}$' is $\Delta L_2 = e_{17}$. Then, according to $>>>7$ and E', the controlling string difference of the second $F_{32/80}$' is $\Delta W' = e_{56,67}$. By Property 4, the probability distribution of the output differences of the first $F_{32/80}$' (ΔV_2) is depicted in **Table 3**. By Property 6 and Property 4, the probability distribution of the output differences of the second $F_{32/80}$' (ΔV_3) is depicted in **Table 4**.

Table 3. The probability distribution of the output differences of the first $F_{32/80}$' (when input difference is e_{17})

OD	e_1	e_3	e_5	e_7	e_9	e_{11}	e_{13}	e_{15}
P.	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}
OD	e_{17}	e_{19}	e_{21}	e_{23}	e_{25}	e_{27}	e_{29}	e_{31}
P.	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}
OD	$e_{1,2}$	$e_{3,4}$	$e_{5,6}$	$e_{7,8}$	$e_{9,10}$	$e_{11,12}$	$e_{13,14}$	$e_{15,16}$
P.	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}
OD	$e_{17,18}$	$e_{19,20}$	$e_{21,22}$	$e_{23,24}$	$e_{25,26}$	$e_{27,28}$	$e_{29,30}$	$e_{31,32}$
P.	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-5}

OD: Output Difference, P.: Probability

Table 4. The probability distribution of the output differences of the second $F_{32/80}$' (when controlling string difference is $e_{56,67}$)

OD	0	e_{13}	$e_{13,14}$	e_{15}	$e_{15,16}$	$e_{13,15}$	$e_{13,15,16}$	$e_{13,14,15}$	$e_{13,14,15,16}$
P.	2^{-4}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-6}	2^{-6}	2^{-6}	2^{-6}
OD	e_5	$e_{5,13}$	$e_{5,13,14}$	$e_{5,15}$	$e_{5,15,16}$	$e_{5,13,15}$	$e_{5,13,15,16}$	$e_{5,13,14,15}$	$e_{5,13,14,15,16}$
P.	2^{-4}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-6}	2^{-6}	2^{-6}	2^{-6}
OD	e_6	$e_{6,13}$	$e_{6,13,14}$	$e_{6,15}$	$e_{6,15,16}$	$e_{6,13,15}$	$e_{6,13,15,16}$	$e_{6,13,14,15}$	$e_{6,13,14,15,16}$
P.	2^{-4}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-6}	2^{-6}	2^{-6}	2^{-6}
OD	$e_{5,6}$	$e_{5,6,13}$	$e_{5,6,13,14}$	$e_{5,6,15}$	$e_{5,6,15,16}$	$e_{5,6,13,15}$	$e_{5,6,13,15,16}$	$e_{5,6,13,14,15}$	$e_{5,6,13,14,15,16}$
P.	2^{-4}	2^{-5}	2^{-5}	2^{-5}	2^{-5}	2^{-6}	2^{-6}	2^{-6}	2^{-6}

OD: Output Difference, P.: Probability

According to **Table 3** and **Table 4**, when $\Delta V_2, \Delta V_3$ coming from **Table 5**, the input difference of $F_{32/96}^{-1}$ is $I(\Delta V_2) \oplus \Delta V_3 = 0$, and the corresponding probability is

$$q_1^{j=17} = 2^{-5} \times 2^{-5} + 2^{-5} \times 2^{-5} + 2^{-5} \times 2^{-4} + 2^{-5} \times 2^{-5} + 2^{-5} \times 2^{-5} + 2^{-5} \times 2^{-4} = 2^{-7}.$$

Table 5. The probability distribution that the input differences of $F^{-1}{}_{32/96}$ is 0 ($\triangle K^i$=(e_{17},0))

$\triangle V_2, \triangle V_3$	e_5, e_{13}	e_7, e_{15}	e_{13}, e_5	$e_{5,6}, e_{13,14}$	$e_{7,8}, e_{15,16}$	$e_{13,14}, e_{5,6}$
Probability	2^{-10}	2^{-10}	2^{-9}	2^{-10}	2^{-10}	2^{-9}

According to our analysis above, the probability of one-round related-key differential characteristic $(0,0) \xrightarrow[\Delta K^i=(e_{17},0)]{} (0,0)$ is $q_1^{j=17} = 2^{-7}$, and [9] made two mistakes in the procedure of constructing it, which are as follows.

(1) Lee et al made a mistake in calculating $\Delta W'$.

In [9], $\Delta W' = e_{56,77}$. However, the correct result is $\Delta W' = e_{56,67}$.

(2) Lee et al made a mistake in calculating the probability of one-round related-key differential characteristic.

Lee et al just used one pair of $\Delta V_2, \Delta V_3$ to calculate the probability of $(0,0) \xrightarrow[\Delta K^i=(e_{17},0)]{} (0,0)$, and their result is 2^{-9}. Actually, according to the six pairs of $\Delta V_2, \Delta V_3$, the correct probability of $(0,0) \xrightarrow[\Delta K^i=(e_{17},0)]{} (0,0)$ is 2^{-7}.

Case2 $\Delta RK^i = (0, e_{17})$

As the high symmetry of round function *Crypt*, similar to Case 1, in Case 2, we could use $\Delta K^i = (0, e_{17})$ to construct another one-round related-key differential characteristic $(0,0) \xrightarrow[\Delta K^i=(0,e_{17})]{} (0,0)$ whose probability is $q_2^{j=17} = q_1^{j=17} = 2^{-7}$.

Case3 $\Delta RK^i = (0,0)$

$(0,0) \xrightarrow[\Delta K^i=(0,0)]{} (0,0)$ holds with probability $q_3^{j=17} = 1$.

These three correct one-round related-key differential characteristics are constructed with $\Delta K_3 = e_{17}$. And with any $\Delta K_m = e_j (m = 1,2,3,4, 1 \le j \le 32)$, we can construct similar one-round related-key differential characteristics. To move a single step forward, we can't ignore the fact that $|\Delta W'| = 2$ for $10 \le j \le 25$, and $|\Delta W'| = 3$ for other j. As a result, we should choose $\Delta K_m = e_j (m = 1,2,3,4, 10 \le j \le 25)$ to make the constructed one-round related-key differential characteristics with higher probability. When $10 \le j \le 25$, the probability distribution of $I(\Delta V_2) \oplus \Delta V_3 = 0$ (the input differences of $F_{32/96}^{-1}$) is depicted in **Appendix Table 2**.

The first type of related-key differential characteristics

According to **Appendix Table 2**, when $j = 12,16,23,24$, the input difference of $F_{32/96}^{-1}$ is $I(\Delta V_2) \oplus \Delta V_3 \ne 0$, and the output difference of $F_{32/96}^{-1}$ is nonzero.

When $\Delta K_3 = e_j$, as depicted in the left part of **Table 6**, we construct the first type of related-key differential characteristics with corresponding three one-round related-key differential characteristics, where $j = 10,11,13,14,15,17,18,19,20,21,22,25$, $\Delta K = (0,0,e_j,0)$,

$\Delta X=(0,e_j)$, $\Delta Y=(0,0)$, and the probability is $(q_1^j)^2\times(q_2^j)^2\times(q_3^j)^4=(q_1^j)^4$.

Table 6. Related-key differential characteristics

	The first type of RDC					The second type of RDC			
round	ΔX^i	ΔKI^i	P.	Case	round	ΔX^i	ΔKI^i	P.	Case
IT	$(0,e_j)$	$(0,e_j)$	1		IT	$(0,e_{17})$	$(0,e_{17})$	1	
1	(0,0)	$(e_j,0)$	q_1^j	1	1	(0,0)	$(e_{17},0)$	$q_1^{j=17}$	1
2	(0,0)	(0,0)	q_3^j	3	2	(0,0)	(0,0)	$q_3^{j=17}$	3
3	(0,0)	(0,0)	q_3^j	3	3	(0,0)	(0,0)	$q_3^{j=17}$	3
4	(0,0)	$(0,e_j)$	q_2^j	2	4	(0,0)	$(0,e_{17})$	$q_2^{j=17}$	2
5	(0,0)	(0,0)	q_3^j	3	5	(0,0)	(0,0)	$q_3^{j=17}$	3
6	(0,0)	$(0,e_j)$	q_2^j	2	6	(0,0)	$(0,e_{17})$	$q_2^{j=17}$	2
7	(0,0)	(0,0)	q_3^j	3	7	(0,0)	(0,0)	$q_3^{j=17}$	3
8	(0,0)	$(e_j,0)$	q_1^j	1	8	(0,0)	$(e_{17},0)$	q_k	
FT	(0,0)	(0,0)	1		FT	$(0,\triangle Y^8_{R,k})$	(0,0)		
Output	(0,0)				Output	$(0,\triangle Y^8_{R,k})$			

RDC: related-key differential characteristics, P.: Probability, IT: initial transformation, FT: final transformation

The second type of related-key differential characteristics

When $\Delta K=(0,0,e_{17},0)$, based on the first type of related-key differential characteristics, we construct the second type of related-key differential characteristics by adding one-bit difference to the input of $F^{-1}_{32/96}$ in the last round, which are depicted in the right part of **Table 6**.

In **Fig. 5**, for the second type of related-key differential characteristics, the differential routes of the last round is dispicted, where $\Delta Y^8_{R,k}$ donates the set of all possible output differences of $F^{-1}_{32/96}$ with $I(\Delta V_2)\oplus\Delta V_3=e_k$ as input difference. According to **Table 3** and **Table 4**, **Appendix Table 3** depicts the probability distribution of $I(\Delta V_2)\oplus\Delta V_3=e_k$. The second type of related-key differential characteristics hold with probability $q_1^{j=17}\times(q_2^{j=17})^2\times(q_3^{j=17})^4\times q_k=2^{-21}\times q_k$.

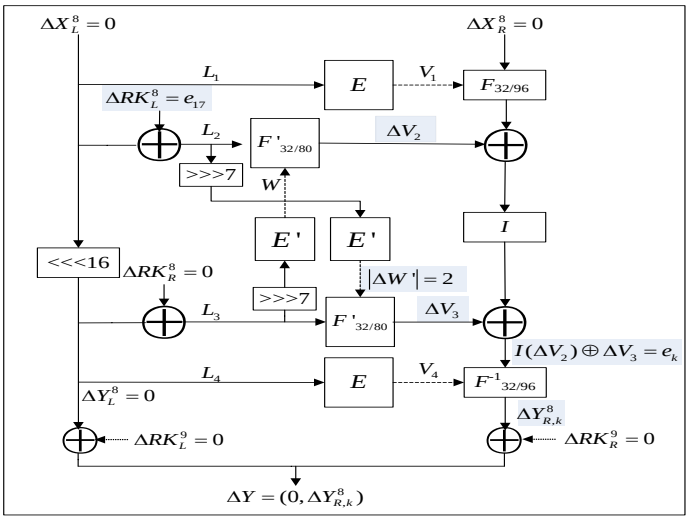


Fig. 5. The differential routes of the last round for the second type of related-key differential characteristics

4 Related-key differential attacks on CHESS-64

In this section, by using the two types of related-key differential characteristics in the previous section, we present two key recovery algorithms on CHESS-64.

4.1 The first related-key differential attack algorithm

For the first type of related-key differential characteristics, according to **Appendix Table 2**, we can exactly get the output difference of the two $F_{32/80}'$ ($\Delta V_2, \Delta V_3$, respectively) in the last round. For the first $F_{32/80}'$, some bits of L_3 could be recovered by using $e_j \to \Delta V_2$ to recover the controlling bits of $F_{32/80}'$. For the second $F_{32/80}'$, some bits of L_2 could be recovered by using difference of controlling string and output difference of $F_{32/80}'$ to recover the controlling bits of $F_{32/80}'$. For a cipher-text pair of the first type of related-key differential characteristics, we can recover key bits by solving equations $L_3 \oplus Y_L = K_1 \oplus K_4$ and $L_2 \oplus <<<16(Y_L) = <<<16(K_1) \oplus K_3$.

For example, when $j = 11$, according to **Appendix Table 2**, we can exactly get $\Delta V_2 = e_9, \Delta V_3 = e_1$ or $\Delta V_2 = e_{9,11}, \Delta V_3 = e_{1,3}$. Further, for $\Delta V_2 = e_9, \Delta V_3 = e_1$, by Theorem 2, five bits of L_3 could be recovered, i. e., $L_3(31) = 1, L_3(3) = 0, L_3(8) = 0, L_3(14) = 0, L_3(19) = 1$, and by Property 6 and Property 4, one bit of L_2 could be recovered, scilicet, $L_2(15) = 0$. Then, the following equations could be constructed.

$$\begin{cases} K_1(31) \oplus K_4(31) = L_3(31) \oplus Y_L(31) = Y_L(31) \oplus 1; \\ K_1(3) \oplus K_4(3) = L_3(3) \oplus Y_L(3) = Y_L(3); \\ K_1(8) \oplus K_4(8) = L_3(8) \oplus Y_L(8) = Y_L(8); \\ K_1(14) \oplus K_4(14) = L_3(14) \oplus Y_L(14) = Y_L(14); \\ K_1(19) \oplus K_4(19) = L_3(19) \oplus Y_L(19) = Y_L(19) \oplus 1; \\ K_1(31) \oplus K_3(15) = L_2(15) \oplus Y_L(31) = Y_L(31). \end{cases}$$

Obviously, by solving equations above, 6 bits information of the master key could be recovered. Similar to $\Delta V_2 = e_9, \Delta V_3 = e_1$, for $\Delta V_2 = e_{9,11}, \Delta V_3 = e_{1,3}$, 7 bits of the master key could be recovered by constructing and solving corresponding equations. For different j, the recovered bits of $K_1 \oplus K_4, <<<16(K_1) \oplus K_3$ are depicted in **Appendix Table 4**.

For $j = 11,13,15,17,18,19,21,22,25$, implement Key Recovery Algorithm 1.

Key Recovery Algorithm 1

[Step 1] Choose 2^{n_j} plaintexts pairs (X, X^*), where $X \oplus X^* = (0, e_j)$.

[Step 2] Encrypt X, X^* with K, K^* to get corresponding cipher-texts Y, Y^*, and check $\Delta Y = (0,0)$ for each Y, Y^*, where $K \oplus K^* = (0, 0, e_j, 0)$.

[Step 3] For each cipher-text pair Y, Y^* passing Step 2, by **Appendix Table 2**, use the method above to recover some bits of L_3, L_2, and extract corresponding bits of $K_1 \oplus K_4, <<<16(K_1) \oplus K_3$ by solving equations $L_3 \oplus Y_L = K_1 \oplus K_4$ and $L_2 \oplus <<<16(Y_L) = <<<16(K_1) \oplus K_3$.

[Step 4] Count the number of hits for recovered key bits by each $\Delta V_2, \Delta V_3$ in Step 3, and output key bits with maximal number of hits.

Analysis of Key Recovery Algorithm 1

According to **Appendix Table 2**, for $j=11,13,15,17,18,19,21,22,25$, the probability of related-key differential characteristics are $2^{-38.8},2^{-32},2^{-36.4},2^{-28},2^{-36},2^{-32},2^{-32},2^{-36},2^{-28.8}$, respectively. If we set $n_j=40.8,34,38.4,30,38,34,34,38,30.8$, the expected number of cipher-text pairs that pass Step 2 is $2^{n_j}\times(q_1^j)^4=4$, and further, the expected number of hits for correct key bits in Step3 is 4. According to **Appendix Table 4**, by Key Recovery Algorithm 1, we can recover 5 bits key information at least for each j, and the expected number of hits for wrong key bits is 4×2^{-5} at most. Therefore, by using Key Recovery Algorithm 1, we can distinguish the correct key bits from the incorrect ones. As depicted in **Appendix Table 3**, for $j=11,13,15,17,18,19,21,22,25$, implementing Key Recovery Algorithm 1, we can obtain the 30 bits of $K_1\oplus K_4$ and 12 bits of $<<<16(K_1)\oplus K_3$.

Theorem 3. By implementing Key Recovery Algorithm 1, we can recover 42 bits information of the master key with a computing complexity of $2^{42.4}$ CHESS-64 encryptions, a data complexity of $2^{44.4}$ chosen plain-texts, and a storage complexity of $2^{12.2}$ bits of storage resources.

Proof. Step 1 needs about $2^{41.8}+2^{35}+2^{39.4}+2^{31}+2^{39}+2^{35}+2^{35}+2^{39}+2^{31.8}\approx 2^{42.4}$ chosen plain-texts. For Step 2, the computing complexity is about $2^{42.4}$ CHESS-64 encryptions, and we need about $4\times 64\times 2\times 9\approx 2^{12.2}$ bits to store cipher-text pairs. For Step 3, the computing complexity of recovering key bits is far less than the computing complexity of Step 2. Therefore, to recover 42(=30+12) bits information of the master key by implementing Key Recovery Algorithm 1, we need a computing complexity of $2^{42.4}$ CHESS-64 encryptions, a data complexity of $2^{44.4}$ chosen plain-texts, and a storage complexity of $2^{12.2}$ bits of storage resources.

4.2 The second related-key differential attack algorithm

For the second type of related-key differential characteristics, in the last round, if the input difference of $F_{32/96}^{-1}$ is $I(\Delta V_2)\oplus\Delta V_3=e_k$, we can get the probability distribution of output difference of $F_{32/96}^{-1}$ by Property 1 and Property 2. For example, if $I(\Delta V_2)\oplus\Delta V_3=e_{16}$, the probability distribution of the output differences of $F_{32/96}^{-1}$ is depicted in **Appendix Table 1**.

In the following, the input difference of $F_{32/96}^{-1}$ is $I(\Delta V_2)\oplus\Delta V_3=e_k$, $\Delta Y_{R,k}^8$ donates the set of all possible output differences of $F_{32/96}^{-1}$, $L_{4,k}$ donates the set of controlling bits which determine the output differences of $F_{32/96}^{-1}$, and $K_{1,k}$ donates the set of corresponding bits of K_1 which is related to $L_{4,k}$. For example,

$$L_{4,16}=\{L_4(1),L_4(2),\cdots,L_4(16),L_4(18),L_4(21),L_4(22),L_4(23),L_4(24),L_4(28),L_4(32)\}.$$

According to the topology of $F_{32/96}^{-1}$, for every $e_k(k=1,2,\cdots,32)$, the output differences of $F_{32/96}^{-1}$ are determined by the left 16 bits and the right 7 bits of L_4. Therefore, $|K_{1,k}|=|L_{4,k}|=23$.

For $k=7,16,23,31$, implement Key Recovery Algorithm 2.

Key Recovery Algorithm 2

[Step 1] Choose 2^n plaintexts pairs (X, X^*), where $X \oplus X^* = (0, e_{17})$.

[Step 2] Encrypt X, X^* with K, K^* to get corresponding cipher-texts Y, Y^*, and check $\Delta Y_L = 0, \Delta Y_R \in \Delta Y_{R,k}^8$ for each Y, Y^*, where $K \oplus K^* = (0, 0, e_{17}, 0)$.

[Step 3] Guess $K_{1,k}$, and for every cipher-text pair passing the test of Step 2, use equation $K_1 \oplus Y_L = L_4$ and output difference of $F_{32/96}^{-1}$ to get the input difference of $F_{32/96}^{-1}$. If the input difference of $F_{32/96}^{-1}$ is e_k, add 1 to the counting of guessed $K_{1,k}$.

[Step 4] Output guessed $K_{1,k}$ with maximal counting number.

Analysis of Key Recovery Algorithm 2

According to **Appendix Table 3**, for $k = 7,16,23,31$, the corresponding probability of related-key differential characteristics are $2^{-30}, 2^{-31}, 2^{-30}, 2^{-30}$, respectively. If we set $n = 40$, there are at least $2^{40} \times 2^{-31} = 2^9$ cipher-text pairs that could pass Step 2 (For $F_{32/96}^{-1}$, different input differences may lead to the same output difference), and further, the expected number of hits for correct $K_{1,k}$ in Step3 is 2^9. On the other hand, as there are no more than $2^{40} \times 2^{-21} = 2^{19}$ cipher-text pairs that could pass Step 2, the expected number of hits for incorrect $K_{1,k}$ is $2^{19} \times 2^{-23} = 2^{-4}$ at most. Therefore, by using Key Recovery Algorithm 2, we can distinguish correct $K_{1,k}$ from the wrong ones.

In the following, we discuss the uniqueness of recovered $K_{1,k}$ by performing Key Recovery Algorithm 2.

According to the topology of $F_{32/96}^{-1}$, if and only if the positions of differences in $\Delta Y_{R,k}^8 (k = 7,16,23,31)$ cover $1,2,\cdots,32$, we can get a unique $K_{1,k}$ by implementing Key Recovery Algorithm 2. When the input difference weight of $F_{32/96}^{-1}$ is 1, any bit of output difference is 1 with probability at least $1/32$. Further, as there are at least $2^{40} \times 2^{-31} = 2^9$ cipher-text pairs that could pass Step 2, positions of differences in $\Delta Y_{R,k}^8 (k = 7,16,23,31)$ cover $1,2,\cdots,32$ with probability at least $1 - C_{32}^1 \times (1-1/32)^{2^9} \approx 1$。 Therefore, we can get a unique $K_{1,k}$ by implementing Key Recovery Algorithm 2. According to **Appendix Table 3**, as $K_1 = K_{1,7} \bigcup K_{1,16} \bigcup K_{1,23} \bigcup K_{1,31}$, we can get the whole 32 bits of K_1 by implementing Key Recovery Algorithm 2 for $k = 7,16,23,31$.

Theorem 4. By implementing Key Recovery Algorithm 2, we can recover 32 bits of the master key with a computing complexity of $2^{41.1}$ CHESS-64 encryptions, a data complexity of 2^{41} chosen plain-texts, and a storage complexity of $2^{26.6}$ bits of storage resources.

Proof. Step 1 needs about 2^{41} chosen plain-texts. For Step 2, the computing complexity is about 2^{41} CHESS-64 encryptions, and we need about $2^{19} \times 2 \times 64 = 2^{26}$ bits of storage resources to store cipher-text pairs. Step 3 needs to compute $F_{32/96}^{-1}$ at most $2^{19} \times 2^{23} = 2^{42}$ times. According to the structure of *Crypt*, a $F_{32/96}^{-1}$ computation is about a quarter of a round computation. Then, Step3 needs about $2^{42} \times 1/4 \times 1/8 = 2^{37}$ CHESS-64 encryptions, and about $2^{23} \times 4 = 2^{25}$ bits of storage resources to store guessed key. Therefore, to recover 32 bits of the

master key by implementing Key Recovery Algorithm 2, we need a computing complexity of $2^{41}+2^{37}\approx 2^{41.1}$ CHESS-64 encryptions, a data complexity of 2^{41} chosen plain-texts, and a storage complexity of $2^{25}+2^{26}\approx 2^{26.6}$ bits of storage resources.

Summary. By Theorem 3 and Theorem 4, we can recover the whole 32 bits of K_1, 30 bits of K_4, and 12 bits of K_3 by implementing Key Recovery Algorithm 1 and Key Recovery Algorithm 2. To recover $42+32=74$ bits of the master key, we need about $2^{42.4}+2^{41.1}\approx 2^{42.9}$ CHESS-64 encryptions, $2^{42.4}+2^{41.1}\approx 2^{42.9}$ chosen plain-texts, and $2^{12.2}+2^{26.6}\approx 2^{26.6}$ bits of storage resources. By performing an exhaustive search for the rest 54 bits key, we can recover the whole 128 bits of the master key and break CHESS-64 absolutely.

5. Conclusion

The DDO-based cipher CHESS-64 has been designed for the application of fast and cheap hardware implementation and high security level, which is considerably resistant against all known attacks. In this paper, however, we put forward two key recovery algorithms on CHESS-64 which are the first correct cryptanalytic results. In detail, as the two key recovery algorithms could be performed independently, we could recover 74(=42+32) bits of the cipher's master key with $2^{42.9}$ chosen-plaintexts, $2^{42.9}$ full-round CHESS-64 encryptions, and $2^{26.6}$ bits of storage resources. Moreover, the related-key differential attacks could be extended to recover the whole 128 bits master key by performing an exhaustive search for the remain 54 bits key, and the corresponding computing complexity are about 2^{54} CHESS-64 encryptions. In this paper, we present a new method to study the properties of DDOs in the procedure of constructing related-key differential characteristics, which is expected to be useful for the further analysis of DDO-based ciphers.

Our related-key differential attacks on CHESS-64 provide some suggestions for the design of DDO-based block ciphers. The most significant features of DDO are its good performance in hardware application and complicating plaintext data. However, according to our research on CHESS-64, the combination of DDOs and Feistel-like structure contribute to the results that attackers could recover data of left-side by differential route in right-side, which results in a threat to the master key. At the same time, to speed up the cipher, designer often choose simple schedule which makes the cipher vulnerable to related-key attack. In a word, to avoid the information leakage algorithms of DDOs under differential attacks, the designer should carefully consider the way to combine DDOs with other cryptographic primitives and the way to combine round key with data.

References

[1] Thi Hong Nhan Vu, Quang Hiep Vu, Yang Koo Lee and The Duy Bui, "A user context recognition method for ubiquitous computing systems," in *Proc. of 8th International Conference on Computing Technology and Information Management* (*ICCM*), pp. 568-573, April 24-26, 2012. Article (CrossRef Link).

[2] A. Bertrand, J. Szurley, P. Ruckebusch, and I. Moerman, "Efficient Calculation of Sensor Utility and Sensor Removal in Wireless Sensor Networks for Adaptive Signal Estimation and Beamforming," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5857–5869, November, 2012. Article (CrossRef Link).

[3] P. Makris, D.N. Skoutas, and C. Skianis, "A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration," *IEEE Communications*

Surveys & Tutorials, vol. 15, no. 1, pp. 362 –386, First Quarter, 2013. Article (CrossRef Link).
[4] Heng Yin and Haining Wang, "Building an Application-Aware IPsec Policy System," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1502–1513, December, 2007. Article (CrossRef Link).
[5] Zhang R., L. Wang, Parr G, et al., "Advances in base- and mobile-station aided cooperative wireless communications: An overview," *IEEE Vehicular Technology Magazine*, vol. 8, no. 1, pp. 57-69, March, 2013. Article (CrossRef Link).
[6] A. A. Moldovyan and N. A. Moldovyan, "A cipher based on data-dependent permutation," *Journal of Cryptology*, vol. 15, no.1, pp. 61-72, March, 2002. Article (CrossRef Link).
[7] Goots N. D., "Modern cryptography: Protect Your Data with Fast Block Cipher," *A-LIST Publish*, Wayne, 2003.
[8] N. Sklavos, N. A. Moldovyan, and O. Koufopavlou, "High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers," *Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 219-231, February, 2005. Article (CrossRef Link).
[9] Sklavos N., Moldovyan N. A., and Koufopavlou O., "A New DDP-based Cipher CIKS-128H: Architecture, Design & VLSI Implementation Optimization of CBC Encryption & Hashing over 1 GBPS," in *Proc. of The 46th IEEE Midwest Symposium on Circuits & Systems*, Cairo, Egypt, December 27-30, 2003. Article (CrossRef Link).
[10] N. A. Moldovyan, et al. "Pure DDP-Based Cipher: Architecture Analysis, Hardware Implementation Cost and Performance up to 6.5 Gbps," *The International Arab Journal of Information Technology*, vol. 2, no. 1, pp. 24-27, 2005. Article (CrossRef Link).
[11] Youngdai Ko, Changhoon Lee, Seokhie Hong, Jaechul Sung and Sangjin Lee, "Related-Key Attacks on DDP Based Ciphers: CIKS-128 and CIKS-128H," *INDOCRYPT*, LNCS 3348, pp. 191–205, December 20-22, 2004. Article (CrossRef Link).
[12] Changhoon Lee, Jongsung Kim, Jaechul Sung, Seokhie Hong, Sangjin Lee and Dukjae Moon, "Related-Key Differential Attacks on Cobra-H64 and Cobra-H128," in *Proc. of 10th IMA International Conference*, LNCS 3796, pp. 201-219, December 19-21, 2005. Article (CrossRef Link).
[13] Changhoon Lee, Sangjin Lee, Jong Hyuk Park, Sajid Hussain, and Jun Hwan Song, "Security analysis of pure DDP-based cipher proper for multimedia and ubiquitous device," *Telecommunication System*, 44(3-4), pp. 267–279, August 2010. Article (CrossRef Link).
[14] Moldovyan N. A., Sklavos N., Moldovyan A. A., and Koufopavlou O., "CHESS-64, a Block Cipher Based on Data-Dependent Operations: Design Variants and Hardware Implementation Efficiency," *Asian Journal of Information Technology*, vol. 4, no. 4, pp. 323-334, April, 2005. Article (CrossRef Link).
[15] N. Moldovyan, A. Moldovyan, M. Eremeev, and N. Sklavos, "New Class of Cryptographic Primitives and Cipher Design for Networks Security," *International Journal of Network Security*, vol.2, no.2, pp. 114-225, February, 2006. Article (CrossRef Link).
[16] A. A. Moldovyan, N. A. Moldovyan, and N. Sklavos, "Controlled elements for designing ciphers suitable to efficient VLSI implementation," *Telecommunication System*, vol. 32, no. 2-3, pp. 149–163, July, 2006. Article (CrossRef Link).
[17] Bac Do Thi, Minh Nguyen Hieu, and Duy Ho Ngoc, "An Effective and Secure Cipher Based on SDDO," *I. J. Computer Network and Information Security*, vol. 11, no. 11, pp. 1-10, October, 2012. Article (CrossRef Link).
[18] Nguyen Hieu Minh, Do Thi Bac, and Ho Ngoc Duy, "New SDDO-Based Block Cipher for Wireless Sensor Network Security," *International Journal of Computer Science and Network Security*, vol.10, no.3, pp. 54–60, March, 2010. Article (CrossRef Link).
[19] Changhoon Lee, Jongsung Kim, Seokhie Hong, and Yang-Sun Lee, "Security Analysis of the Full-Round CHESS-64 Cipher Suitable for Pervasive Computing Environments," *Journal of Universal Computer Science*, vol. 15, no. 5, pp. 1007-1022, May, 2009. Article (CrossRef Link).
[20] Jinkeon Kang, Kitae Jeong, Sang-Soo Yeo, and Changhoon Lee, "Related-Key Attack on the MD-64 Block Cipher Suitable for Pervasive Computing Environments," in *proc. of 26th International Conference on Advanced Information Networking and Applications Workshops*, pp.

726-731, March 26-29, 2012. Article (CrossRef Link).

Appendix: Some tables for related-key differential attacks on CHESS-64

Appendix Table 1. The probability distribution of the output differences of $F^{-1}_{32/96}$ (when the input difference is e_{16})

ID	$e_{1,2,3,4}$	$e_{1,2,4}$	$e_{1,3}$	$e_{1,4}$	$e_{2,3}$	$e_{2,3,4}$	$e_{2,4}$	e_{3}
Pro.	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-6}	2^{-6}
ID	$e_{3,4}$	e_{4}	$e_{5,6,7,8}$	$e_{5,6,8}$	$e_{5,7}$	$e_{5,8}$	$e_{6,7}$	$e_{6,7,8}$
Pro.	2^{-6}	2^{-5}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}
ID	$e_{6,8}$	e_{7}	$e_{7,8}$	e_{8}	$e_{9,10,11,12}$	$e_{9,10,12}$	$e_{9,11}$	$e_{9,12}$
Pro.	2^{-6}	2^{-6}	2^{-6}	2^{-5}	2^{-7}	2^{-7}	2^{-7}	2^{-7}
ID	$e_{10,11}$	$e_{10,11,12}$	$e_{10,12}$	e_{11}	$e_{11,12}$	e_{12}	$e_{13,14,15,16}$	$e_{13,14,16}$
Pro.	2^{-7}	2^{-7}	2^{-6}	2^{-6}	2^{-6}	2^{-5}	2^{-7}	2^{-7}
ID	$e_{13,15}$	$e_{13,16}$	$e_{14,15}$	$e_{14,15,16}$	$e_{14,16}$	e_{15}	$e_{15,16}$	e_{16}
Pro.	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-6}	2^{-6}	2^{-6}	2^{-5}
ID	$e_{17,18,19,20}$	$e_{17,18,20}$	$e_{17,19}$	$e_{17,20}$	$e_{18,19}$	$e_{18,19,20}$	$e_{18,20}$	e_{19}
Pro.	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-6}	2^{-6}
ID	$e_{19,20}$	e_{20}	$e_{21,22,23,24}$	$e_{21,22,24}$	$e_{21,23}$	$e_{21,24}$	$e_{22,23}$	$e_{22,23,24}$
Pro.	2^{-6}	2^{-5}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}
ID	$e_{22,24}$	e_{23}	$e_{23,24}$	e_{24}	$e_{25,26,27,28}$	$e_{25,26,28}$	$e_{25,27}$	$e_{25,28}$
Pro.	2^{-6}	2^{-6}	2^{-6}	2^{-5}	2^{-7}	2^{-7}	2^{-7}	2^{-7}
ID	$e_{26,27}$	$e_{26,27,28}$	$e_{26,28}$	e_{27}	$e_{27,28}$	e_{28}	$e_{29,30,31,32}$	$e_{29,30,32}$
Pro.	2^{-7}	2^{-7}	2^{-6}	2^{-6}	2^{-6}	2^{-5}	2^{-7}	2^{-7}
ID	$e_{29,31}$	$e_{29,32}$	$e_{30,31}$	$e_{30,31,32}$	$e_{30,32}$	e_{31}	$e_{31,32}$	e_{32}
Pro.	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-6}	2^{-6}	2^{-6}	2^{-5}

ID: Input Difference, Pro.: Probability

Appendix Table 2. The probability distribution of $\Delta V_2, \Delta V_3$ (when the input differences of $F^{-1}_{32/96}$ is 0)

j	ΔV_2, ΔV_3, p		Total Probability
10	e_9, e_1, 2^{-10}	e_{10}, e_2, 2^{-10}	$2^{-8.4}$
	$e_{9,11}$, $e_{1,3}$, 2^{-12}	$e_{9,12}$, $e_{1,4}$, 2^{-12}	
	$e_{10,11}$, $e_{2,3}$, 2^{-12}	$e_{10,12}$, $e_{2,4}$, 2^{-12}	
11	e_9, e_1, 2^{-10}	$e_{9,11}$, $e_{1,3}$, 2^{-12}	$2^{-9.7}$
12	None		0
13	e_{21}, e_{29}, 2^{-9}	e_{22}, e_{30}, 2^{-9}	2^{-8}
14	e_1, e_9, 2^{-10}	e_2, e_{10}, 2^{-10}	$2^{-8.4}$
	$e_{1,3}$, $e_{9,11}$, 2^{-12}	$e_{1,4}$, $e_{9,12}$, 2^{-12}	
	$e_{2,3}$, $e_{10,11}$, 2^{-12}	$e_{2,4}$, $e_{10,12}$, 2^{-12}	
15	e_1, e_9, 2^{-10}	$e_{1,3}$, $e_{9,11}$, 2^{-12}	$2^{-9.1}$
	$e_{1,9}$, $e_{9,1}$, 2^{-12}	$e_{1,10}$, $e_{9,2}$, 2^{-12}	
	$e_{1,3,9}$, $e_{9,11,1}$, 2^{-14}	$e_{1,3,10}$, $e_{9,11,2}$, 2^{-14}	
16	None		0
17	e_5, e_{13}, 2^{-10}	$e_{5,6}$, $e_{13,14}$, 2^{-10}	2^{-7}
	e_7, e_{15}, 2^{-10}	$e_{7,8}$, $e_{15,16}$, 2^{-10}	
	e_{13}, e_5, 2^{-9}	$e_{13,14}$, $e_{5,6}$, 2^{-9}	
18	e_{25}, e_{17}, 2^{-10}	e_{27}, e_{19}, 2^{-10}	2^{-9}
19	e_1, e_9, 2^{-9}	$e_{1,2}$, $e_{9,10}$, 2^{-9}	2^{-8}

20	e_3, e_{11}, 2^{-9}	$e_{3,4}$, $e_{11,12}$, 2^{-9}	2^{-8}
21	e_5, e_{13}, 2^{-9}	$e_{5,6}$, $e_{13,14}$, 2^{-9}	2^{-8}
22	e_{17}, e_{25}, 2^{-10}	e_{19}, e_{27}, 2^{-10}	2^{-9}
23	None		0
24	None		0
25	e_{21}, e_{29}, 2^{-10}	$e_{21,22}$, $e_{29,30}$, 2^{-10}	$2^{-7.2}$
	$e_{21,23}$, $e_{29,31}$, 2^{-12}	$e_{21,23,24}$, $e_{29,31,32}$, 2^{-12}	
	$e_{21,22,23}$, $e_{29,30,31}$, 2^{-12}	$e_{21,22,23,24}$, $e_{29,30,31,32}$, 2^{-12}	
	e_{29}, e_{21}, 2^{-9}	$e_{29,30}$, $e_{21,22}$, 2^{-9}	

Appendix Table 3. The probability distribution of $\Delta V_2, \Delta V_3$ (when the input differences of $F^{-1}_{32/96}$ is 1, and j=17)

ΔV_2	e_9	e_{11}	e_{13}	e_{13}	e_{15}	e_1	e_3
ΔV_3	0	0	0	$e_{5,6}$	0	0	0
e_k	e_1	e_3	e_5	e_6	e_7	e_9	e_{11}
q_k	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}

ΔV_2	e_5	e_7	$e_{7,8}$	e_5	e_7	e_5	$e_{5,6}$	e_7
ΔV_3	0	$e_{13,15}$	$e_{13,14,15}$	$e_{13,14}$	0	$e_{13,15}$	$e_{13,14,15}$	$e_{15,16}$
e_k	e_{13}			e_{14}	e_{15}			e_{16}
q_k	$2^{-9}+2^{-11}+2^{-11}=2^{-9.6}$			2^{-10}	$2^{-9}+2^{-11}+2^{-11}=2^{-9.6}$			2^{-10}

ΔV_2	e_{25}	e_{27}	e_{29}	e_{31}	e_{17}	e_{19}	e_{21}	e_{23}
ΔV_3	0	0	0	0	0	0	0	0
e_k	e_{17}	e_{19}	e_{21}	e_{23}	e_{25}	e_{27}	e_{29}	e_{31}
q_k	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}

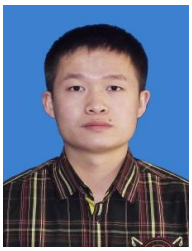
Appendix Table 4-1. The recovered key bits by Key Recovery Algorithm 1

	ΔV_2, ΔV_3	Recovered Bits of $K_1 \oplus K_4$	Recovered Bits of $K_1 \oplus <<<16(K_3)$	p	Total Probability
10	e_9, e_1,	30,3,8,14,19	15	2^{-10}	$2^{-8.4}$
	e_{10}, e_2	30,3,8,14,19	15	2^{-10}	
	$e_{9,11}$, $e_{1,3}$	30,3,8,14,19,20	15,16	2^{-12}	
	$e_{9,12}$, $e_{1,4}$	30,3,8,14,19,20	15,16	2^{-12}	
	$e_{10,11}$, $e_{2,3}$	30,3,8,14,19,20	15,16	2^{-12}	
	$e_{10,12}$, $e_{2,4}$	30,3,8,14,19,20	15,16	2^{-12}	
11	e_9, e_1	30,3,8,14,19	15	2^{-10}	$2^{-9.7}$
	$e_{9,11}$, $e_{1,3}$	30,3,8,14,19,20	15,16	2^{-12}	
13	e_{21}, e_{29}	32,6,27,20,25	None	2^{-9}	2^{-8}
	e_{22}, e_{30}	32,6,27,20,25	None	2^{-9}	
14	e_1, e_9	32,4,8,10,15	19	2^{-10}	$2^{-8.4}$
	e_2, e_{10}	32,4,8,10,15	19	2^{-10}	
	$e_{1,3}$, $e_{9,11}$	32,4,8,10,15,16	19,20	2^{-12}	
	$e_{1,4}$, $e_{9,12}$	32,4,8,10,15,16	19,20	2^{-12}	
	$e_{2,3}$, $e_{10,11}$	32,4,8,10,15,16	19,20	2^{-12}	
	$e_{2,4}$, $e_{10,12}$	32,4,8,10,15,16	19,20	2^{-12}	
15	e_1, e_9	1,4,8,10,15	19	2^{-10}	$2^{-9.1}$
	$e_{1,3}$, $e_{9,11}$	1,4,8,10,15,16	19,20	2^{-12}	
	$e_{1,9}$, $e_{9,1}$	1,4,8,10,15,14,19	19	2^{-12}	

	$e_{1,10}$, $e_{9,2}$	1,4,8,10,15,14,19	19	2^{-12}	
	$e_{1,3,9}$, $e_{9,11,1}$	1,4,8,10,15, 16,14,19	19,20	2^{-14}	
	$e_{1,3,10}$, $e_{9,11,2}$	1,4,8,10,15, 16,14,19	19,20	2^{-14}	

Appendix Table 4-2. The recovered key bits by Key Recovery Algorithm 1

	ΔV_2, ΔV_3	Recovered Bits of $K_1 \oplus K_4$	Recovered Bits of $K_1 \oplus <<<16(K_3)$	p	Total Probability
17	e_5, e_{13}	2,9,30,13,17	21	2^{-10}	2^{-7}
	$e_{5,6}$, $e_{13,14}$	2,9,30,13,17	21	2^{-10}	
	e_7, e_{15}	2,9,30,13,18	22	2^{-10}	
	$e_{7,8}$, $e_{15,16}$	2,9,30,13,18	22	2^{-10}	
	e_{13}, e_5	2,9,30,17,21	None	2^{-9}	
	$e_{13,14}$, $e_{5,6}$	2,9,30,17,21	None	2^{-9}	
18	e_{25}, e_{17}	2,7,29,23,11	23	2^{-10}	2^{-9}
	e_{27}, e_{19}	2,7,29,23,12	24	2^{-10}	
19	e_1, e_9	3,7,28,11,25	None	2^{-9}	2^{-8}
	$e_{1,2}$, $e_{9,10}$	3,7,28,11,25	None	2^{-9}	
20	e_3, e_{11}	3,7,28,11,16	None	2^{-9}	2^{-8}
	$e_{3,4}$, $e_{11,12}$	3,7,28,11,16	None	2^{-9}	
21	e_5, e_{13}	4,26,30,13,17	None	2^{-9}	2^{-8}
	$e_{5,6}$, $e_{13,14}$	4,26,30,13,17	None	2^{-9}	
22	e_{17}, e_{25}	4,8,29,19,23	11	2^{-10}	2^{-9}
	e_{19}, e_{27}	4,8,29,19,24	12	2^{-10}	
25	e_{21}, e_{29}	6,29,3,21,25	13	2^{-10}	$2^{-7.2}$
	$e_{21,22}$, $e_{29,30}$	6,29,3,21,25	13	2^{-10}	
	$e_{21,23}$, $e_{29,31}$	6,29,3,21,25,10	13,14	2^{-12}	
	$e_{21,23,24}$, $e_{29,31,32}$	6,29,3,21,25,10	13,14	2^{-12}	
	$e_{21,22,23}$, $e_{29,30,31}$	6,29,3,21,25,10	13,14	2^{-12}	
	$e_{21,22,23,24}$, $e_{29,30,31,32}$	6,29,3,21,25,10	13,14	2^{-12}	
	e_{29}, e_{21}	6,29,3,25,13	None	2^{-9}	
	$e_{29,30}$, $e_{21,22}$	6,29,3,25,13	None	2^{-9}	



Wei Luo received B.S. degree from Zhengzhou Information Science and Technology Institute in 2011. He is studying for M.S. degree in cryptography in the same university. His research interests include design and analysis of block cipher.
(Email: luowei.crypt@gmail.com)

Jiansheng Guo is a professor of Zhengzhou Information Science and Technology Institute. His main subject interest is cryptography and his main teaching lies in the areas of information systems, the theory of cryptography and quantum computation. He received Ph.D. degree in cryptography from Zhengzhou Information Science and Technology Institute in 2004.
(Email: guojs2013@gmail.com)