Related-Key Rectangle Attack on Round-reduced *Khudra* Block Cipher

Xiaoshuang Ma[1,2] Kexin Qiao[1,2]

[1]State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093, China
[2]Data Assurance and Communication Security Research Center,
Chinese Academy of Sciences, Beijing 100093, China
{xshma13,kxqiao13}@is.ac.cn

Abstract. *Khudra* is a block cipher proposed in the SPACE'2014 conference, whose main design goal is to achieve suitability for the increasingly popular Field Programmable Gate Array (FPGA) implementation. It is an 18-round lightweight cipher based on recursive Feistel structure, with a 64-bit block size and 80-bit key size. In this paper, we compute the minimum number of active F-functions in differential characteristics in the related-key setting, and give a more accurate measurement of the resistance of *Khudra* against related-key differential cryptanalysis. We construct a related-key boomerang quartet with probability 2^{-48} for the 14-round *Khudra*, which is better than the highest probability related-key boomerang quartet of the 14-round *Khudra* of probability at most 2^{-72} claimed by the designers. Then we propose a related-key rectangle attack on the 16-round *Khudra* without whitening key by constructing a related-key rectangle distinguisher for 12-round *Khudra* with a probability of $2^{-23.82}$. The attack has time complexity of $2^{78.68}$ memory accesses and data complexity of $2^{57.82}$ chosen plaintexts, and requires only four related keys. This is the best known attack on the round-reduced *Khudra*.

Keywords: *Khudra* block cipher, rectangle attack, related-key attack

1 Introduction

Differential cryptanalysis was first proposed by Biham and Shamir in [5] and is one of the most powerful attacks on block ciphers. Differential cryptanalysis analyzes differential propagation patterns of a cipher to discover its non-random behaviors, and uses these behaviors to build a distinguisher or recover the key. Under the model of related-key attack [1], which considers the information extracted from the two encryptions under two related keys, related-key differential attack [11] allows the attacker to operate differences not only in the plaintexts, but also in keys, though the key values are initially unknown.

For differential attacks, finding out differential characteristics with high probabilities is of great importance, and there are several ways to try a good searching for such differential characteristics. After the Mixed-Integer Linear Programming (MILP) technique was used to analyze ciphers [6], Mouha *et al.* [15] and Wu *et al.* [21] proposed MILP based techniques to find automatically a lower bound of the number of differentially active S-boxes of word-oriented symmetric ciphers. Later, Sun *et al.* [17, 19] and Qiao *et al.* [16] improved Mouha *et al.*'s method to make it capable of searching for the actual differential characteristics. In this paper, we will apply the methods in [16, 17, 19] and some other techniques to find related-key differential characteristics of the new cipher *Khudra*.

The rectangle-boomerang style attacks [2, 10, 20] are clever extensions of differential cryptanalysis. In a rectangle-boomerang style attack, the cipher is treated as a cascade of two sub-ciphers, where differential with high probability is used in each of these sub-ciphers. The aim of the attack is to benefit from the slow mixing in reduced round versions of the cipher attacked. The rectangle attack [2] is transited from the boomerang attack [20], and the amplified boomerang attack [10] considers all possible intermediate differences and significantly increases the probability of a right quartet, easing the requirements to a chosen plaintext attack instead of adaptive chosen plaintext and ciphertext attack.

To improve the results of the rectangle attack, Biham *et al.* [3] proposed a new algorithm and a generic way for launching key recovery attacks and calculating the time and data complexity with boomerang or rectangle distinguishers. In this paper, we will combine the rectangle attack with related-key differentials we found to give an analysis on the 16-round *Khudra* block cipher.

The *Khudra* block cipher [13] was proposed by Kolay *et al.* in the SPACE'2014 conference. It is a lightweight cipher suitable for Field Programmable Gate Array (FPGA) implementation. *Khudra* is an 18-round block cipher based on the recursive generalized Feistel structure, and has a 64-bit block size and 80-bit key size. In this paper, we will firstly compute the minimum number of active F-functions in the related-key differential characteristics of *Khudra*, and give a more accurate measurement of the resistance of *Khudra* against differential cryptanalysis. We will construct a related-key boomerang quartet of probability of 2^{-48} for the 14-round *Khudra*, while the designers of *Khudra* claimed that the highest probability related-key boomerang quartets of 14-round *Khudra* have the probability at most 2^{-72}. Then we will propose a related-key rectangle attack on the 16-round *Khudra* without whitening key by constructing a related-key rectangle distinguisher for 12-round *Khudra* with a probability of $2^{-23.82}$. The attack has time complexity of $2^{78.68}$ memory accesses, $2^{59.77}$ encryptions,and $2^{57.72}$ decryptions, and data complexity of $2^{57.82}$ chosen plaintexts.It requires only four related keys. This is the best known attack on the round-reduced *Khudra*.

Organization of the paper. We give a brief introduction of *Khudra* in Section 2, and present our analysis result on *Khudra* against related-key differencial attack in Section 3. In Section 4, we firstly describe the construction of the related-key rectangle distinguishers, and then introduce our related-key rectangle attack on round-reduced *Khudra*. Finally we conclude the paper in Section 5.

2 Description of *Khudra*

In this section, we briefly recall the design of the block cipher *Khudra* and we refer the readers to [13] for more details.

Khudra is a lightweight block cipher suitable for resource-constrained devices. The designers of *Khudra* have shown that the strategies for designing lightweight block cipher on Application Specific Integrated Circuits (ASICs) are not suitable for Field Programmable Gate Arrays (FPGAs). They have identified new methods and design criteria for designing lightweight block ciphers on FPGAs. *Khudra* is an actual practice of these guidelines.

Khudra is an 18-round block cipher based on the recursive Feistel structure, which has a 64-bit block size and 80-bit key size. To encrypt a 64-bit plaintext block using a 80-bit key, *Khudra* employs a generalized type-2 transformation (GFS) [7] of a classical Feistel structure, with four branches in one round. The output of the F-function then XORs with the next branch and the round key, then passes through the Feistel permutation. The structure of

the encryption algorithm is demonstrated in the left part of Fig. 1, which is called the Outer Structure of the cipher.

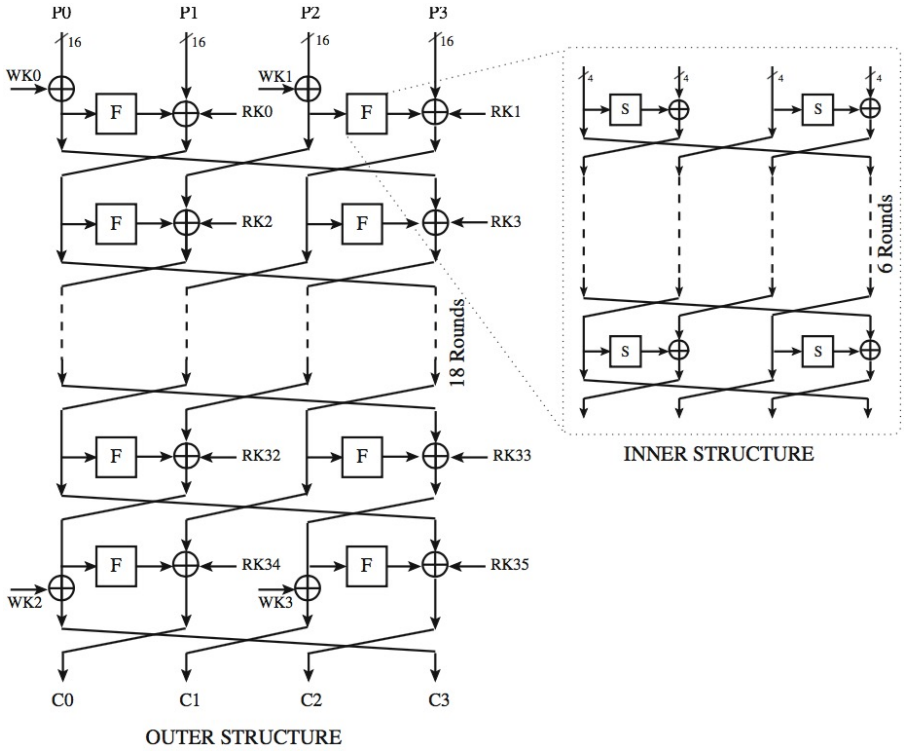

Fig. 1: The Structure of *Khudra* [13]

The Inner Structure. The outer structure of 4-branch type-2 generalized Feistel structure of *Khudra* is also used for the construction for the F-function, see the right part of Fig. 1. The structure of the F-function is called the Inner Structure of the cipher. It is implemented with a two-level recursive structure. In the inner structure, half of the state is updated by 4×4 S-boxes to achieve nonlinear operations.

Let the input block of the i-th round be $(P_0(i-1), P_1(i-1), P_2(i-1), P_3(i-1)) \in \{0,1\}^{16} \times \{0,1\}^{16} \times \{0,1\}^{16} \times \{0,1\}^{16}$, and RK_i denote the round key. The data processing procedure can be described as follows:

$$\begin{aligned} P_0(i) &= P_1(i-1) \oplus F(P_0(i-1)) \oplus RK_{2(i-1)}, \\ P_1(i) &= P_2(i-1), \\ P_2(i) &= P_3(i-1) \oplus F(P_2(i-1)) \oplus RK_{2(i-1)+1}, \\ P_3(i) &= P_0(i-1) \end{aligned} \tag{1}$$

for $i = 1, \cdots, 18$.

The S-box. *Khudra* uses the S-box in the PRESENT block cipher as its substitution box. The S-box is shown in Table 1. The difference distribution table of the S-box is given in Appendix A. Note that throughout this paper we write bit-strings in their hexadecimal format, *e.g.*, the binary string 1100 is written as a hexadecimal symbol C.

Table 1: The S-box of *Khudra*

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

The Key Schedule. The key schedule algorithm of the cipher generates 36 round keys RK_i $(0 \leq i < 36)$ and 4 whitening keys WK_i $(0 \leq i < 4)$, all of 16 bits. Represent the 80-bit master key as $(K_0, K_1, K_2, K_3, K_4)$, each K_i is of 16 bits. The whitening keys WK_i and the round keys RK_i are generated as follows:

$$\begin{aligned} &WK_0 = K_0, WK_1 = K_1, WK_2 = K_3, WK_3 = K_4, \\ &RC_i = \{0||i_{(6)}||00||i_{(6)}||0\}, \\ &RK_i = K_{i \bmod 5} \oplus RC_i \end{aligned} \tag{2}$$

for $i = 0, \cdots, 35$.

3 Security Analysis of *Khudra* against Related-key Differential Attack

In this section, we apply the MILP based methods presented in [14, 16, 17, 19] to *Khudra* in the related-key model.

We develop a Python program to generate the MILP instances for *Khudra* in the "lp" format [9]. In order to find the characteristic with the maximal probability, we implement the technique proposed by Sun *et al.* [19] in our Python framework for automatic cryptanalysis. We have computed the minimum number of active F-functions in differential characteristics for related-key model, which gives a more accurate measurement of the resistance of *Khudra* against related-key differential cryptanalysis.

Table 2: Minimum number of active F-functions in related-key model

No. of Rounds		1	2	3	4	5	6	7	8	9
Min. # Act.	Related-key ([13])	0	0	0	1	2	3	3	3	4
F-functions	Related-key (this paper)	0	0	0	0	1	1	2	3	-

In Table 2 we list our results and that of the designers. Clearly, we can see that we found related-key differential characteristics with fewer active F-functions compared with that of the designers. Particularly, we give in Table 3 a related-key differential characteristic for the 4-round *Khudra*, which is a characteristic with no active F-function.

Table 2 shows that *Khudra* is not as secure as the designers claimed in [13]. We can construct a boomerang quartet of probablity of 2^{-48} for the 14-round *Khudra* as a cascade

Table 3: A related-key differential characteristic for 4-round *Khudra*

r	ΔI	$\Delta RK_{2(r-1)}$	$\Delta RK_{2(r-1)+1}$	NAS	Prob.
2	0000000000000004	0000	0004	0	1
3	0000000000000000	0000	0000	0	1
4	0000000000000000	0000	0000	0	1
5	0000000000000000	0004	0000	0	1
6	0004000000000000				

of two 7-round sub-ciphers, while the designers [13] claimed that the highest probability boomerang quartet of 14-round *Khudra* have a probability at most 2^{-72}.

4 Related-Key Rectangle Attacks on *Khudra*

In this section, we firstly give a brief introduction to the construction of the related-key rectangle distinguisher for block ciphers. Then we introduce an analysis on the 16-round *Khudra* without whitening key. The related-key differential characteristics presented here are derived from an application of the MILP based method of Sun *et al.* [18].

4.1 Related-Key Rectangle Distinguisher

The related-key rectangle attack [4, 8, 12] is a combination of the related-key and rectangle attack. Let E denote the encryption function of a block cipher. The related-key differential is a quadruple of a plaintext difference ΔP, a ciphertext difference ΔC, a key difference ΔK, and the corresponding probability $\Pr[E_K(P) \oplus E_{K\oplus\Delta K}(P \oplus \Delta P) = \Delta C]$.

The rectangle attack treats E as a cascade of four sub-ciphers as $E = E_f \circ E_1 \circ E_0 \circ E_b$, where E is composed of a core $E' = E_1 \circ E_0$ covered by additional rounds E_b and E_f. Assume that for E_0 we have a differential $\alpha \to \beta$ under a key difference ΔK^0 with probability p, and for E_1 there exists a differential $\gamma \to \delta$ under key difference ΔK^1 with probability q, where (α, β) and (γ, δ) stand for the input-output differences for E_0 and E_1 respectively. The rectangle attack can be mounted for all possible differences β at the end of E_0 and γ at the beginning of E_1. Thus we define $\hat{p}_\alpha$ and $\hat{q}_\delta$ as the probabilities related to α and δ respectively as follows:

$$\hat{p}_\alpha = \sqrt{\sum_\beta \Pr^2[\alpha \to \beta]}, \quad \hat{q}_\delta = \sqrt{\sum_\gamma \Pr^2[\gamma \to \delta]}. \tag{3}$$

The related-key rectangle attack involves four different unknown but related keys – $K^a, K^b = K^a \oplus \Delta K^0, K^c = K^a \oplus \Delta K^1, K^d = K^a \oplus \Delta K^0 \oplus \Delta K^1$, where the key differences ΔK^0 and ΔK^1 are the respective key differences for sub-ciphers E_0 and E_1. The basic related-key rectangle distinguisher is constructed as follows:

Step 1. Choose N_0 plaintext pairs (P^a, P^b) satisfying $P^a \oplus P^b = \alpha$ at random and ask for the encryption of P^a under K^a and of P^b under K^b, *i.e.*, $C^a = E_{K^a}(P^a)$ and $C^b = E_{K^b}(P^b)$.

Step 2. Choose N_1 palintext pairs (P^c, P^d) satisfying $P^c \oplus P^d = \alpha$ at random and ask for the encryption of P^c under K^c and of P^d under K^d, *i.e.*, $C^c = E_{K^c}(P^c)$ and $C^d = E_{K^d}(P^d)$.

Step 3. Search for quartets of cipertexts (C^a, C^b, C^c, C^d) satisfying $C^a \oplus C^c = C^b \oplus C^d = \delta$.

The probability of the rectangle distinguisher is given by $\Pr = 2^{-n}\hat{p}_\alpha^2\hat{q}_\delta^2$ where n is the block size. The related-key differentials should satisfy the condition $\hat{p}_\alpha^2 \cdot \hat{q}_\delta^2 > 2^{-n}$ to make the distinguisher make sense. As we expect the number of right quartets is taken to be 4 to get at least one right quartet in the data set with probability 0.982, we set the number of plaintext pairs needed as $2^{n/2+1}/\hat{p}_\alpha\hat{q}_\delta$. As in general differential attacks, after the distinguisher has been detected, one or more rounds are attached before and after the distinguisher for key recovery.

4.2 The First Differential (E_0) and Second Differential (E_1)

Our methods apply the methods in [3] in related-key model with improvement on reducing time complexity by exploiting the properties of a right quartet depending on the GFS feature. We treat the 16-round *Khudra* encryption function E as a cascade of four sub-ciphers as $E = E_f \circ E_1 \circ E_0 \circ E_b$, where E is composed of a core $E' = E_1 \circ E_0$ covered by additional rounds E_b and E_f. E_0 is composed of rounds $3-8$, E_1 commences with round 9 and stops at the end of round 14. Rounds $1-2$ and rounds $15-16$ serve as the rounds before and after the distinguisher respectively (E_b and E_f).

Table 4: The number of characteristics for E_0 of different probability

Prob.	Num.	Prob.	Num.	Prob.	Num.	Prob.	Num.
2^{-12}	1	2^{-22}	4	2^{-24}	54	2^{-26}	130
2^{-14}	2	2^{-23}	21	2^{-25}	88	2^{-27}	65

All the related-key differential characteristics used in sub-cipher E_0 have the same input difference $\alpha = 0000000000005C00$ and they all work with the master key difference $\Delta K^0 = 5C00000000005C000000$. We found many such characteristics with varying differences at the end of 8th round by the MILP method [18]. The numbers of characteristics with different probabilities are shown in Table 4. Therefore the overall probability for E_0 is

$$\hat{p}_\alpha = \sqrt{1 \cdot (2^{-12})^2 + 2 \cdot (2^{-14})^2 + 4 \cdot (2^{-22})^2 + \cdots + 65 \cdot (2^{-27})^2} \approx 2^{-11.91}$$

according to Equation (3). Table 5 shows one of the characteristics used for E_0. The probability of the differential characteristic is 2^{-12}.

All the related-key differential characteristics used in sub-cipher E_1 have the same output difference $\delta = 000000005C000000$ and they work with the master key difference $\Delta K^1 = 5C0000005C0000000000$. Due to the symmetry of GFS that *Khudra* applied, we can also find the same number of characteristics of different probability in E_1 as in E_0. As it can be seen in Table 4, the overall probability for E_1 is also $\hat{q}_\delta \approx 2^{-11.91}$. Thus, the probability of the related-key rectangle distinguisher is given by $\Pr = 2^{-64}\hat{p}_\alpha^2\hat{q}_\delta^2 \approx 2^{-111.64}$.

4.3 The Construction of Differential in E_b and Differential in E_f

Since *Khudra* applies the generalized Feistel structure, we can deduce the relation among the differences. Given the α difference, Fig. 2 shows the differential propagation pattern of E_b.

Table 5: The related-key differential characteristic for sub-cipher E_0 of *Khudra*

r	ΔI	$\Delta RK_{2(r-1)}$	$\Delta RK_{2(r-1)+1}$	NAS	Prob.
3	0000000000005C00	0000	5C00	0	1
4	0000000000000000	0000	0000	0	1
5	0000000000000000	5C00	0000	0	1
6	5C00000000000000	5C00	0000	6	2^{-12}
7	0000000000005C00	0000	5C00	0	1
8	0000000000000000	0000	5C00	0	1
9	000000005C000000				

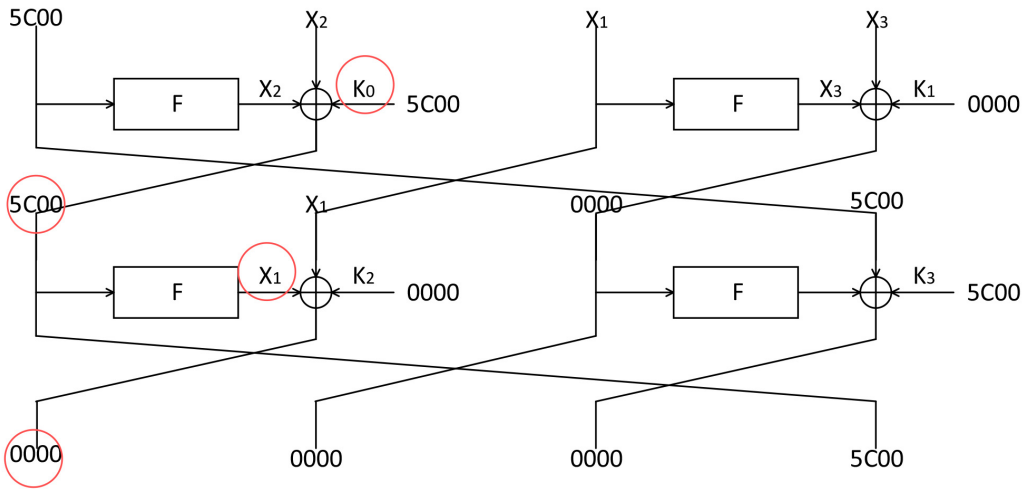

Fig. 2: The differential propagation pattern of E_b

As is seen in Fig. 2, X_1 and X_2 are the output differences of the F-function when the input difference is $5C00$, and X_3 is the output difference of the F-function with respect to X_1 as the input difference. It turns out that there are $2^{14.52}$ kind of reasonable X_1. When the input pair of the F-function is fixed with a difference of $5C00(X_1)$, the corresponding output pair of the F-function will have a fixed output difference. The sub-keys used in E_b that can affect α difference is K_0 (16-bit), and the corresponding related-key difference is $\Delta K_0^0 = 5C00$. The way it influence the difference before round 3 is also shown in Fig. 2 with the circle. Thus we find out the pattern of plaintext differences that can possibly lead to α difference in E_b.

Given the δ difference, we can deduce the relation between the differences after δ. Fig. 3 shows the differential propagation pattern of E_f, where Y_1 is the output difference of the F-function when the input difference is $5C00$, and Y_2 is the output difference of the F-function with respect to Y_1 as the input difference. It turns out that there are $2^{14.52}$ kind of reasonable Y_1. When the input pair of the F-function is fixed with a difference of $5C00$ (Y_1), the corresponding output pair of the F-function will have a fixed output difference. Thus we find out the pattern of ciphertext differences that δ can possibly lead to in E_f.

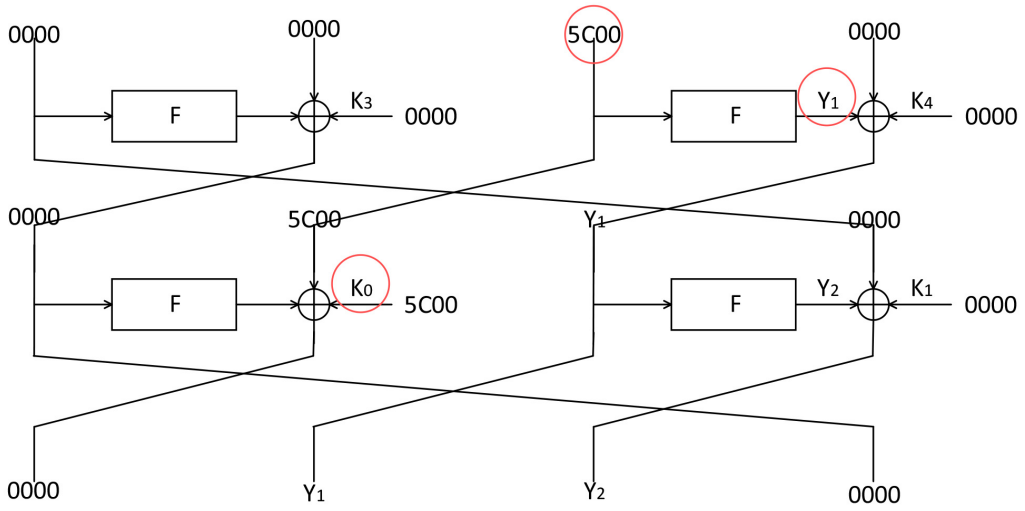

Fig. 3: The differential propagation pattern of E_f

As is seen in Fig. 3, the subkeys used for decryption in E_f that affect δ difference is K_0 (16-bit), and the corresponding related-key difference is $\Delta K_0^1 = 5C00$. The way it influence the difference after round 14 is also shown in Fig. 3 with the circle.

4.4 The Attack

The basic idea of the attack is to try all subkeys which affect the differences before and after the distinguisher (i.e., in E_b and E_f). The selection criterion of right subkeys is whether it can lead to an α difference at the beginning of E_0, and the δ difference at the end of E_1 can lead to the ciphertext difference.

To attack the 16-round *Khudra*, we request $2^{24.82}$ structures of 2^{32} plaintexts each.

The attack works as follows:

1. Data Generation:

a) Generate $Y = 2^{24.82}$ structures $S_1^a, \cdots, S_Y^a$, each of 2^{32} plaintexts. In each structure, fix the left most 32 bits of the plaintexts and enumerate the right most 32 bits. Ask for the encryption of the structures under K^a.

b) For any i ($i = 1, \cdots, Y$), describe any plaintext in structure S_i^a with $P_a = (P_0^a, P_1^a, *, *)$ (P_0^a and P_1^a represent 16 fix bits respectively, and $*$ represents 16 arbitrary bits). According to Fig. 2, the plaintexts in structure S_b^i are generated as follows:

$$P_0^b = P_0^a \oplus 5C00,$$
$$P_1^b = P_1^a \oplus F(P_0^a) \oplus F(P_0^b).$$

Ask for the encryption of the resulting plaintext $P_b = (P_0^b, P_1^b, *, *)$ under $K^b = K^a \oplus \Delta K^0$ for obtaining $S_1^b, \cdots, S_Y^b$.

c) Ask for the encryption of plaintexts generated in Step1(a) under $K^c = K^a \oplus \Delta K^1$ (to obtain $S_1^c, \cdots, S_Y^c$).

d) Ask for the encryption of plaintexts generated in Step1(b) under $K^d = K^a \oplus \Delta K^0 \oplus \Delta K^1$ (to obtain $S_1^d, \cdots, S_Y^d$).

- This step requires data complexity of $2^{57.82}$ chosen plaintexts. We keep all the 2^{16} input values of the F-function and the corresponding output values in a table and each structure in $S_1^b, \cdots, S_Y^b$ can be generated by 2 memory accesses. So the time complexity of this step is $2^{58.82}$ encryptions, and $2Y = 2^{25.82}$ memory accesses. In each structure we get 2^{64} plaintext pairs, which lead to 2^{32} kinds of differences after E_b, where 2^{32} of them satisfy α difference before E_0. Thus, the total number of pairs with α difference before the core function is $2^{56.82}$ that produce $2^{113.64}$ quartets of which $2^{113.64} \cdot 2^{-111.64} = 4$ are expected to be *right*.

2. Initializing Counters:

Initialize an array of 2^{16} counters. Each counter corresponds to a different guess of $K^a_{0[15:0]}$.

- Time complexity of this step is 2^{16} memory accesses.

3. Data Analysis:

a) According to Fig. 3, insert the $N = 2^{58.82}$ ciphertexts of S^a, S^b, S^c and S^d into four hash tables T^a, T^b, T^c and T^d respectively indexed by the left most 16 bits and the right most 16 bits. If a collision occurs in the same bins of (T^a, T^c), denote the ciphertexts as $C^a = (C_0^a, C_1^a, C_2^a, C_3^a)$ and $C^c = (C_0^c, C_1^c, C_2^c, C_3^c)$. For each C^a in each bin of T^a, build a hash table indexed by 2^16 values of C_1^c, and insert the corresponding value of C_2^c by the equation $C_2^c \oplus C_2^a = F(C_1^a) \oplus F(C_1^c)$. Check whether the corresponding $2^{m+32+16} = 2^{m+48} = 2^{72.82}$ values of candidate C^c in T^c. If this is the case, check whether $\Delta = C_1^a \oplus C_1^c \oplus 5C00$ is one of the $2^{14.52}$ possible output differences may be caused by an input difference $5C00$ to the F-function. Do the same for T_b and T_d.

- This step has time complexity of $2^{58.82}$ memory accesses from inserting all the ciphertexts in hash tables. In the hash tables there exist 2^{32} bins and in each bin we expect to have $2^{24.82}$ ciphertexts. Therefore, we need $2^{72.82}$ memory accesses to build the hash table for C^a, and $2^{72.82}$ memory accesses to check if the candidate C^c is in T^c. There are $2^{65.64}$ pairs of C^a and C^c to be checked in the next situation. Out of the 2^{16} possible differences for a pair, only $2^{14.52}$ differences can be caused by the δ difference from the distinguisher and thus about $2^{64.16}$ pairs remain in T^a (T^b) and T^c (T^d). We keep all the $2^{14.52}$ differences that can be caused by δ in a hash table, and thus the check requires one memory access for each colliding pair. The time complexity of this step is $2^{74.82} + 2^{65.16}$ memory accesses.

b) For each surviving pairs (C^a, C^c) $((C^b, C^d))$ from the previous step, denote C^i's structure by S^{C^i} and attach to C^a (C^b) the index of S^{C^c} (S^{C^d}). After all the remaining pairs are processed, keep a hash table $H^{S_i^a}$ for each structure S_i^a ($i = 1, 2, \cdots, Y$) and insert the ciphertexts in S_i^a into $H_{S_i^a}$ according to the indexes of the structures that the ciphertext is related to. Similarly, keep the hash tables $H_{S_i^b}$ for S_i^b ($i = 1, 2, \cdots, Y$).

- This step requires one memory access for each remaining pair of the previous step and thus needs $2^{65.16}$ memory accesses.

c) For a right quartet (P^a, P^b, P^c, P^d) and the corresponding ciphertexts (C^a, C^b, C^c, C^d), it must be combined by some $P^a \in S_i^a, P^b \in S_i^b$ and $P^c \in S_j^c, P^d \in S_j^d$ where S_j^c is related to C^a and S_j^d is related to C^b and $i, j \in 1, 2, \cdots, Y$ (not necessarily

distinct). In each pair of structures (S_i^a, S_i^b) $(i = 1, \cdots, Y)$, we search for two ciphertexts C^a and C^b from S_a^i and S_b^i respectively which are attach to some other pair of structures (S_j^c, S_j^d). When we found such a pair, check whether the differences of (P^a, P^b) and (P^c, P^d) can cause α. Denote the corresponding plaintexts as $P^a = (P_0^a, P_1^a, P_2^a, P_3^a)$ and $P^b = (P_0^b, P_1^b, P_2^b, P_3^b)$. First we check whether the equation $P_3^a \oplus P_3^b = F(P_2^a) \oplus F(P_2^b)$ to be true. With this move, we can reduce the candidate quartets by 2^{-16}. Then check the same for the plaintexts to which C^a and C^b are related. For the quartets remained after this move, check whether the difference of P_2^a and P_2^b is one of the $2^{14.52}$ possible output differences may be caused by an input difference $5C00$ to the F-function, and the same for P_2^c and P_2^d.

- There are $2^{64.16}$ attachments (colliding pairs) distributed over $2^{24.82}$ structures, The first filter here can reduce the candidate quartets to $2^{21.86}$ in each structure. Out of the 2^{16} possible differences for a pair of plaintexts, only $2^{14.52}$ differences can cause the α difference into the distinguisher. Therefore, out of the $2^{46.68}$ possible quartets only $2^{43.72}$ quartets remain. Since this filtering requires one memory access for each candidate quartet, thus the algorithm requires $2^{78.68} + 2^{62.68} + 2^{46.68} + 2^{45.20}$ memory accesses.

4. Subkey Bits Guess: For each remaining quartet $((P^a, P^b), (P^c, P^d)), ((C^a, C^b), (C^c, C^d))$ perform:

(a) For each guess of the 16 bits of K_0^a, we have

$$\begin{aligned} K_0^b &= K_0^a \oplus \Delta K_0^0, \\ K_0^c &= K_0^a \oplus \Delta K_0^1, \\ K_0^d &= K_0^a \oplus \Delta K_0^0 \oplus \Delta K_0^1. \end{aligned} \tag{4}$$

(b) Increment the counter that correspond to K_0^a if

$$E_{b_{K_0^a}}(P^a) \oplus E_{b_{K_0^b}}(P^b) = E_{b_{K_0^c}}(P^c) \oplus E_{b_{K_0^d}}(P^d) = \alpha \tag{5}$$

and

$$E^{-1}_{f_{K_0^a}}(C^a) \oplus E^{-1}_{f_{K_0^c}}(C^c) = E^{-1}_{f_{K_0^b}}(C^b) \oplus E^{-1}_{f_{K_0^d}}(C^d) = \delta. \tag{6}$$

- There are $2^{14.52}$ possible input differences that lead to α difference after E_b and totally 2^{16} guesses of subkey bits of K_a, therefore, $2^{1.48}$ subkeys on average take one of these differences to α. As each pair suggests $2^{1.48}$ subkeys, a quartet agrees on $(2^{1.48})^2/2^{16} = 2^{-13.04}$ subkeys for E_b. Similarly, a quartet agrees on $2^{-13.04}$ subkeys for E_f. In total, we get a candidate list of $2^{-42.08}$ subkeys from each quartet. Thus, for the $2^{43.72}$ remaining quartets, there are total 2^{16} possible subkeys and $2^{1.64}$ hits. Averagely, the number of hits for a wrong subkey is $2^{-14.36}$, while the number of expected hits for the right one is 4. Thus, the attack can almost always succeed in recovering subkey bits. This step requires about $2^{57.72}$ encryptions and $2^{57,72}$ decryptions and $2^{59.72}$ memory accesses.

5. Output the subkey with maximal counter.

- This step requires 2^{16} memory accesses.

Thus, the overall attack has data complexity of $2^{57.82}$ chosen plaintexts, time complexity of $2^{78.68}$ memory accesses, $2^{59.77}$ encryptions, and $2^{57.72}$ decryptions. The expected number of right quartets is taken to be 4.

5 Conclusion

In this paper, we have launched an related-key rectangle attack on 16-round *Khudra*, with time complexity of $2^{78.68}$ memory accesses, $2^{59.77}$ encryptions and $2^{57.72}$ decryptions, and data complexity of $2^{57.82}$ chosen plaintexts. The attack is a generic related-key rectangle attack with a differential distinguisher deduced from an MILP based search. This is the best known cryptanalysis presented on round-reduced *Khudra*.

References

1. Biham, E.: New types of cryptanalytic attacks using related keys. Journal of Cryptology 7(4), 229–246 (1994)
2. Biham, E., Dunkelman, O., Keller, N.: The rectangle attack-rectangling the Serpent. In: Advances in Cryptology–EUROCRYPT 2001, pp. 340–357. Springer (2001)
3. Biham, E., Dunkelman, O., Keller, N.: New results on boomerang and rectangle attacks. In: Fast Software Encryption. pp. 1–16. Springer (2002)
4. Biham, E., Dunkelman, O., Keller, N.: Related-key boomerang and rectangle attacks. In: Advances in Cryptology–EUROCRYPT 2005, pp. 507–525. Springer (2005)
5. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY 4(1), 3–72 (1991)
6. Borghoff, J., Knudsen, L.R., Stolpe, M.: Bivium as a mixed-integer linear programming problem. In: Cryptography and Coding, pp. 133–152. Springer (2009)
7. Hoang, V.T., Rogaway, P.: On generalized feistel networks. In: Advances in Cryptology - CRYPTO 2010. pp. 613–630. Springer (2010)
8. Hong, S., Kim, J., Lee, S., Preneel, B.: Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In: Fast Software Encryption. pp. 368–383. Springer (2005)
9. IBMsoftware-group: User-manual cplex 12. URL:http://www-01.ibm.com (2011)
10. Kelsey, J., Kohno, T., Schneier, B.: Amplified boomerang attacks against reduced-round MARS and Serpent. In: Fast Software Encryption. pp. 75–93. Springer (2001)
11. Kelsey, J., Schneier, B., Wagner, D.: Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. Information and Communications Security pp. 233–246 (1997)
12. Kim, J., Kim, G., Hong, S., Lee, S., Hong, D.: The related-key rectangle attack–application to SHACAL-1. In: Information Security and Privacy. pp. 123–136. Springer (2004)
13. Kolay, S., Mukhopadhyay, D.: Khudra: A new lightweight block cipher for fpgas. In: Security, Privacy, and Applied Cryptography Engineering, SPACE 2014. pp. 126–145. Springer (2014)
14. Ma, X., Hu, L., Sun, S., Qiao, K., Shan, J.: Tighter security bound of MIBS block cipher against differential attack. In: Network and System Security. Springer (2014)
15. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using Mixed-Integer linear programming. In: Information Security and Cryptology. pp. 57–76. Springer (2012)
16. Qiao, K., Hu, L., Sun, S., Ma, X., Kan, H.: Improved MILP Modeling for Automatic Security Evaluation and Application to FOX. IEICE TRANSACTIONS on Cryptography and Information Security (Special) (to appear)
17. Sun, S., Hu, L., Song, L., Xie, Y., Wang, P.: Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks. In: Inscrypt 2013. Springer (2013)
18. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747 (2014), `http://eprint.iacr.org/`

19. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search : Application to SIMON, PRESENT-80/128, LBlock, DES(L) and other bit-oriented block ciphers. In: Advances in Cryptology–ASIACRYPT 2014. Springer (2014)
20. Wagner, D.: The boomerang attack. In: Fast Software Encryption. pp. 156–170. Springer (1999)
21. Wu, S., Wang, M.: Security evaluation against differential cryptanalysis for block cipher structures. Tech. rep., Cryptology ePrint Archive, Report 2011/551 (2011)

A The Difference Distribution Table (DDT) of the S-box of *Khudra*

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4