Chinese Journal of Electronics
vol. 33, no. 1, pp. 72–79, January 2024
https://doi.org/10.23919/cje.2022.00.313

RESEARCH ARTICLE

SAT-Based Automatic Searching for Differential and Linear Trails: Applying to CRAX

Yiyi HAN[1,2], Caibing WANG[1,2], Zhongfeng NIU[1,2], and Lei HU[1,2]

1. *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China*
2. *School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China*

Corresponding author: Lei HU, Email: hulei@iie.ac.cn
Manuscript Received November 1, 2022; Accepted December 9, 2022

Abstract — Boolean satisfiability problem (SAT) is now widely applied in differential cryptanalysis and linear cryptanalysis for various cipher algorithms. It generated many excellent results for some ciphers, for example, Salsa20. In this research, we study the differential and linear propagations through the operations of addition, rotation and XOR (ARX), and construct the SAT models. We apply the models to CRAX to search differential trails and linear trails automatically. In this sense, our contribution can be broadly divided into two parts. We give the bounds for differential and linear cryptanalysis of Alzette both up to 12 steps, by which we present a 3-round differential attack and a 3-round linear attack for CRAX. We construct a 4-round key-recovery attack for CRAX with time complexity 2^{89} times of 4-round encryption and data complexity 2^{25}.
Keywords — Differential cryptanalysis, Linear cryptanalysis, Boolean satisfiability problem, CRAX.
Citation — Yiyi HAN, Caibing WANG, Zhongfeng NIU, *et al.*, "SAT-Based Automatic Searching for Differential and Linear Trails: Applying to CRAX," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 72–79, 2024. doi: 10.23919/cje.2022.00.313.

I. Introduction

Addition-Rotation-XOR (ARX) design strategy is widely used in many symmetric cryptographic primitives, where only the addition operation provides non-linearity. These three kinds of operations are very comprehensible and perform efficiently in the implementation of software. Therefore, it is well suited for designing lightweight block ciphers with ARX structure. There are lots of cryptographic primitives constructed with the ARX structure, such as block ciphers HIGHT [1], SPARX [2], TEA [3], XTEA [4], CHAM [5] and SPECK [6], stream ciphers Salsa20 [7] and ChaCha [8], and hash functions Skein [9] and BLAKE [10]. The designers also adopted the strategy to construct ARX-box—Alzette [11], which can be used to design block ciphers, for instance, CRAX and TRAX.

Differential cyptanalysis [12] and linear cryptanalysis [13] are two fundamental methodologies applied in the cryptanalysis of numerous symmetric ciphers, also including ARX designs. On the basis of these two kinds of cryptanalysis, a number of cryptanalytic tools have been researched, such as zero correlation linear cryptanalysis [14], high order differential cryptanalysis [15], impossible differential cryptanalysis [16], and differential-linear cryptanalysis [17]. For an efficient differential (linear) attack, the most important thing is to search for differential (linear) characteristics with the complexity better than brute force. Thus, automated search methods for differential and linear trails with better probability or correlation are a growing area of research due to their efficiency.

At EUROCRYPT'94, the branch-and-bound algorithm [18] was put forward by Matsui to search differential characteristics (DCs) and linear characteristics (LCs) with better probability or correlation for DES block cipher. It is very strengthful and still widely in use by now. In 2011, Mouha *et al.* [19], Wu *et al.* [20] presented another automatic method derived from mixed integer linear programming (MILP) problem. It was applied in determining the least amount of active Sboxes in differential and linear attacks. Sun *et al.* [21], [22] broaden their framework and gave an MILP model suitable for bit-oriented ciphers, which can give a more precise assessment of the resilience of ciphers against various at-

Associate Editor: Prof. Debiao HE, Wuhan University.

tacks, for example, a linear analysis using MILP is presented to attack the block cipher GIFT [23]. Recently, a method grounded in Boolean satisfiability problem (SAT) [24] is proposed to assess the security of symmetric-key primitive designs. The SAT problem can conclude whether a bunch of constraints could be fulfilled by assigning values to variables. Researchers have produced some work based on SAT method, including differential and linear attacks for SIMON cipher algorithm [25], linear attacks for SPECK and Chaskey [26], and integral attacks for SHACAL-2 and LEA [27].

Our target is to formulate the searching of ideal differential and linear trails of CRAX as a SAT problem. More specifically, we need to explore the rules on the differential characteristics and linear approximations when they go through three operations of ARX structure, and use Boolean satisfiability language to represent them. Then we request the solver to output valid DCs and LCs with the probability or correlation that we input. Our work performs well in identifying differential and linear trails with better probability or correlation, thus we are able to give CRAX a more thorough security assessment resisting differential and linear cryptanalysis.

Our contributions We revisit the properties of DCs and LCs when they pass through the operations of ARX structure. Then we present the differential and linear trails of CRAX searched by SAT method. We also give a 4-round key-recovery attack for CRAX. All experiments are finished on DELL server (48 Core, Intel(R)Xeon(R) CPU E7330, 2.20 GHz).

Our main results are enumerated in the following.

1) We find a 12-step optimal single-key differential trail for Alzette with probability 2^{-59} for the first time, which takes 30.8 hours. Then we give a 3-round differential attack for CRAX using this trail.

2) We get a 12-step optimal linear trail for Alzette with correlation 2^{-30} for the first time, which cost 3.8 h, by which we present a 3-round linear attack for CRAX.

3) We give a 4-round key-recovery attack for CRAX with time complexity 2^{89} times of 4-round encryption and data complexity 2^{25}.

Table 1 displays the comparisons between our distinguished differential characteristics, linear approximations, and previous findings. Note that all the experimental probability data are provided by us since reference [11] did not give the details.

Outline In Section II, we present the notations and recall the construction of CRAX. We give the main definitions and theorems and construct SAT models for searching the differential trails and linear trails automatically in Section III. In Section IV, we present our results and give a key-recovery attack for 4-round CRAX. Section V is the conclusion of this paper.

II. Preliminaries

In this section, we introduce notation and related definitions, and give a brief description of CRAX.

Table 1 Outline of different attacks on CRAX

	Type	# Round	Probability/Correlation		Ref.
			Theoretical	Experimental	
CRAX	DC	1	2^{-6}	$2^{-5.96}$	[11]
	DC	1.25	2^{-10}	$2^{-9.94}$	[11]
	DC	1.5	2^{-18}	$2^{-17.99}$	[11]
	DC	1.75	2^{-26}	$2^{-25.69}$	This paper
	DC	2	2^{-34}	$2^{-33.98}$	This paper
	DC	2.25	2^{-39}	–	This paper
	DC	2.5	2^{-45}	–	This paper
	DC	2.75	2^{-51}	–	This paper
	DC	3	2^{-59}	–	This paper
	LC	1	2^{-2}	–	[11]
	LC	1.25	2^{-5}	–	[11]
	LC	1.5	2^{-8}	–	[11]
	LC	1.75	2^{-13}	–	[11]
	LC	2	2^{-17}	–	[11]
	LC	2.25	2^{-19}	–	This paper
	LC	2.5	2^{-21}	–	This paper
	LC	2.75	2^{-25}	–	This paper
	LC	3	2^{-30}	–	This paper

Note: DC: differential characteristic, LC: linear characteristic

1. Notation

The notations we used in this paper are listed in Table 2.

Table 2 Notations

Notation	Description
$\mathbb{F}_2$	$\{0,1\}$
$\mathbb{Z}^+$	The positive integer field
x_i	The i-th bit of $\boldsymbol{x}$
x_0/LSB	The least siginificant bit of $\boldsymbol{x}$
x_{n-1}/MSB	The most significant bit of $\boldsymbol{x}$
$\boxplus$	Modula addition operation
$\oplus$	XOR operation
$\vee$	OR operation
$\neg\boldsymbol{x}$	The NOT operation of $\boldsymbol{x}$
$\boldsymbol{x}\cdot\boldsymbol{y}$	$\bigoplus_{i=0}^{n} x_i y_i$
$\boldsymbol{x}\lll r$	The left circular rotation of $\boldsymbol{x}$ by r
$\boldsymbol{x}\ggg r$	The right circular rotation of $\boldsymbol{x}$ by r
$\boldsymbol{x}\preccurlyeq\boldsymbol{y}$	$x_i \le y_i, \forall i \in \{0,1,\ldots,n-1\}$
$1_{\mathcal{A}}$	Defined to 1 iff $\mathcal{A}$ holds, to 0 elsewhere
step	The inner iteration of Alzette
$wt(\boldsymbol{x})$	The Hamming weight of $\boldsymbol{x}$
DC	Differential characterisitc
LC	Linear characterisitc

2. Description of CRAX

CRAX is a lightweight block cipher proposed by

Beierle *et al.* at CRYPTO 2020. The block size of CRAX is 64-bit and the key size is 128-bit. CRAX is as light as SPECK and performs better for short messages. Thus, CRAX can be implemented in real-world applications like IoT devices [11]. The design of CRAX merely consists of the iterative Alzette, interleaving it with key injections. CRAX uses a very straightforward key schedule algorithm: the master key is signified by (K_0, K_1), and the 64-bit round key $k_i = K_{i \bmod 2} \oplus i$ is used at the start of round i. The round structure of CRAX is depicted in Figure 1, and the structure of ARX S-box Alzette is depicted in Figure 2.

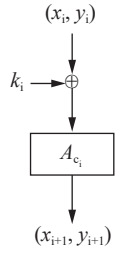


Figure 1 CRAX round function.

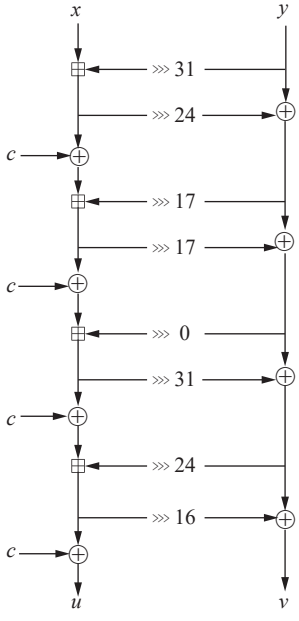


Figure 2 The Alzette instance A_c.

3. Security analysis of CRAX

We emphasize on the security of CRAX resisting differential attacks and linear attacks. Since the structure of CRAX is the iteration of Alzette with the key injection, the cryptanalysis of CRAX can directly refer to the cryptanalysis of Alzette. According to the definition of *step* in Section II.1, we regard Alzette as an ARX-box comprising of the composition of 4 steps of the form

$$\begin{cases} x_{i+1} = (x_i \boxplus (y_i \ggg p_i)) \oplus \boldsymbol{c} \\ y_{i+1} = y_i \oplus (x_{i+1} \ggg q_i) \end{cases}$$

where i-th step is characterized by the shift numbers $(p_i, q_i) \in \mathbb{Z}_{32} \times \mathbb{Z}_{32}$. Thus, an r-step attack for Alzette can be transformed into a $\frac{r}{4}$-round attack for CRAX. Here, we only give the analytical results on Alzette for simplicity.

On the differential properties, Beierle *et al.* used the Algorithm 1 in [28] to calculate the thresholds on the maximum expected differential characteristic probabilities (MEDCP) of Alzette and gave a 6-step differential characteristic with probability 2^{-18} [11]. For 7 and 8 steps, they could not get a tight bound and proved that valid differential trails with probability higher than 2^{-24} and 2^{-32} do not exist.

On the linear properties, Beierle *et al.* used the automatic method based on MILP described in [29] and the method based on SAT in [26]. They obtained an 8-step linear trail with correlation 2^{-17} [11].

On the differential-linear properties, Liu *et al.* [30] gave both theoretical and experimental correlation for a 4-step differential-linear distinguisher for Alzette, which are $2^{-0.27}$ and $2^{-0.1}$, respectively. In addition, they firstly proposed the notion of rotational differential-linear cryptanalysis and gave a distinguisher of 4-step Alzette, where the theoretical correlation is $2^{-11.37}$. At Crypto 2022, Niu *et al.* [31] improved the analysis in [30]. They gave a new 4-step rotational differential-linear distinguisher for Alzette, and the theoretical correlation is $2^{-5.57}$. For the differential-linear distinguishers, they also evaluated the correlations of the approximations for 4-, 5-, 6- and 8-step Alzette, where the theoretical correlations are 1, $-2^{-0.33}$, $2^{-4.95}$ and $-2^{-8.24}$, respectively.

On the security assessment against impossible differential attacks, Xu *et al.* [32] proposed a new automatic tool to search impossible differential characteristics for Alzette. For 4-step Alzette, they found 4096 impossible differential distinguishers with fixed input weight $wt_{\text{in}} = 1$ and output weight $wt_{\text{out}} = 1$, and 128993 ones with $wt_{\text{in}} = 2$ and $wt_{\text{out}} = 1$.

III. SAT Models for Operations of CRAX

In this section, we firstly analyze the propagations of DCs and LCs when they passing through all operations of CRAX, and introduce the SAT models in [33] for these operations. There are four operations in total, which include branching, rotation, XOR and modular addition. The former three operations are linear so the differential and linear propagations are conclusive, while modular addition is the unique non-linear one.

1. Non-probabilistic models

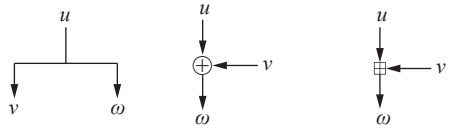
Due to the rules of differences and linear approximations passing through branching and XOR are antithetic [34], the model of XOR (resp. branching) in linear cryptanalysis matches with the model of branching (resp. XOR) in differential cryptanalysis. To avoid repetition, we only list the differential models in the following.

Differential model of branching Figure 3(a) shows an instance of branching. Suppose we represent the in-

put difference with $\boldsymbol{u}$ and use $\boldsymbol{v}$ and $\boldsymbol{\omega}$ to denote output differences of the branching operation, the constraints are presented as follows.

$$0 \le j \le n-1, \begin{cases} u_j \vee \neg v_j = 1 \\ \neg u_j \vee v_j = 1 \\ u_j \vee \neg \omega_j = 1 \\ \neg u_j \vee \omega_j = 1 \end{cases}$$

Rotation For rotational circular left (resp. right) shift, the difference and linear mask should also move the corresponding number of bits to the left (resp. right).



(a) Branching (b) XOR (c) Modular addition

Figure 3 Instances of (a) Branching, (b) XOR, (c) Modular addition.

Differential model of XOR The instance of n-bit XOR operation is shown in Figure 3(b), we denote the input differences by $\boldsymbol{u}$ and $\boldsymbol{v}$ and use $\boldsymbol{\omega}$ to represent the output difference. The constraints are listed as follows.

$$0 \le j \le n-1, \begin{cases} u_j \vee v_j \vee \neg \omega_j = 1 \\ u_j \vee \neg v_j \vee \omega_j = 1 \\ \neg u_j \vee v_j \vee \omega_j = 1 \\ \neg u_j \vee \neg v_j \vee \neg \omega_j = 1 \end{cases}$$

2. Probabilistic models

In this subsection, we analyze the differential and linear propagations of modular addition. Then we give the corresponding SAT-based models.

Differential model of modular addition Figure 3 (c) shows an instance of modular addition. To identify the differential peculiarities of modular addition, we introduce related definition and theorems at first.

Definition 1 Let $\boldsymbol{u}, \boldsymbol{v}$ denote the input XOR differences, and $\boldsymbol{\omega}$ represent the output XOR difference of ADD operation. The XOR-differential probability (xdp^+) is computed as follows, which indicates $\boldsymbol{u}$ and $\boldsymbol{v}$ spread to $\boldsymbol{\omega}$ over the addition operation.

$$xdp^+(\boldsymbol{u}, \boldsymbol{v} \to \boldsymbol{\omega}) = \frac{\#\{(\boldsymbol{a}, \boldsymbol{b}) : ((\boldsymbol{a} \oplus \boldsymbol{u}) \boxplus (\boldsymbol{b} \oplus \boldsymbol{v})) \oplus (\boldsymbol{a} \boxplus \boldsymbol{b}) = \boldsymbol{\omega}\}}{2^{2n}}$$

In [35], Lipmaa and Moriai adopted two steps to calculate $xdp^+(\boldsymbol{u}, \boldsymbol{v} \to \boldsymbol{\omega})$. First, they check if the differential characteristic is valid. Then, they computed the differential probability xdp^+. These two steps can be expressed in the following two Theorems.

Theorem 1 [35] The differential $(\boldsymbol{u}, \boldsymbol{v} \to \boldsymbol{\omega})$ is valid iff the following equations hold.

1) $u_0 \oplus v_0 \oplus \omega_0 = 0$;

2) $u_j \oplus v_j \oplus \omega_j = u_{j-1}$, for $u_{j-1} = v_{j-1} = \omega_{j-1}$, $j \in [1, n-1]$.

Theorem 2 [35] Suppose that $(\boldsymbol{u}, \boldsymbol{v} \to \boldsymbol{\omega})$ is a valid differential characteristic of addition, then we can compute $xdp^+ = 2^{-\Sigma_{j=0}^{n-2} \neg eq(u_j, v_j, \omega_j)}$ to obtain the probability, where

$$eq(u_j, v_j, \omega_j) = \begin{cases} 1, & u_j = v_i = \omega_j \\ 0, & \text{otherwise} \end{cases}$$

Suppose we denote the input differences with $\boldsymbol{u}$ and $\boldsymbol{v}$, and use $\boldsymbol{\omega}$ to represent the output difference. According to these two theorems, we can give the differential model of modular addition with CNF formulas as follows, where $0 \le j \le n-2$.

$$\begin{cases} u_j \vee v_j \vee \omega_j \vee u_{j+1} \vee v_{j+1} \vee \neg \omega_{j+1} = 1 \\ u_j \vee v_j \vee \omega_j \vee u_{j+1} \vee \neg v_{j+1} \vee \omega_{j+1} = 1 \\ u_j \vee v_j \vee \omega_j \vee \neg u_{j+1} \vee v_{j+1} \vee \omega_{i+1} = 1 \\ u_j \vee v_j \vee \omega_j \vee \neg u_{j+1} \vee \neg v_{j+1} \vee \neg \omega_{j+1} = 1 \\ \neg u_j \vee \neg v_j \vee \neg \omega_j \vee u_{j+1} \vee v_{j+1} \vee \omega_{j+1} = 1 \\ \neg u_j \vee \neg v_j \vee \neg \omega_j \vee u_{j+1} \vee \neg v_{j+1} \vee \neg \omega_{j+1} = 1 \\ \neg u_j \vee \neg v_j \vee \neg \omega_j \vee \neg u_{j+1} \vee v_{j+1} \vee \neg \omega_{j+1} = 1 \\ \neg u_j \vee \neg v_j \vee \neg \omega_j \vee \neg u_{j+1} \vee \neg v_{j+1} \vee \omega_{j+1} = 1 \end{cases}$$

$$\begin{cases} u_0 \vee v_0 \vee \neg \omega_0 = 1 \\ u_0 \vee \neg v_0 \vee \omega_0 = 1 \\ \neg u_0 \vee v_0 \vee \omega_0 = 1 \\ \neg u_0 \vee \neg v_0 \vee \neg \omega_0 = 1 \end{cases}$$

Linear model of modular addition Also, we give the definition of the correlation of LCs through addition operation at first.

Definition 2 We use $\boldsymbol{x}, \boldsymbol{y}$ to denote the input masks, and signify the output mask with $\boldsymbol{z}$ of addition modulo 2^n. The correlation $cor_{\boxplus}$ can be calculated as the following equation.

$$\begin{aligned} &cor_{\boxplus}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ &= cor(\boldsymbol{z} \cdot (\boldsymbol{a} \boxplus \boldsymbol{b}) \oplus \boldsymbol{x} \cdot \boldsymbol{a} \oplus \boldsymbol{y} \cdot \boldsymbol{b}) \\ &= 2^{-2n}(\#\{\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_2^n : \boldsymbol{z} \cdot (\boldsymbol{a} \boxplus \boldsymbol{b}) \oplus \boldsymbol{x} \cdot \boldsymbol{a} \oplus \boldsymbol{y} \cdot \boldsymbol{b} = 0\} \\ &\quad - \#\{\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_2^n : \boldsymbol{z} \cdot (\boldsymbol{a} \boxplus \boldsymbol{b}) \oplus \boldsymbol{x} \cdot \boldsymbol{a} \oplus \boldsymbol{y} \cdot \boldsymbol{b} = 1\}) \end{aligned}$$

In [36] and [37], Nyberg and Wallén proposed an algorithm based on rational series to compute the correlation of linear masks of modular addition, and utilized automaton to make the implementation efficient. In [38], Schulte-Geers developed a non-recursive algorithm to compute the correlation, which is derived from the fact that addition modulo 2^n and a vectorial quadratic boolean function are CCZ-equivalent. Since the latter is in a form of boolean function, it is more suitable for constructing SAT models. Thus we adopt the second algorithm and give the related theorem as follows.

Theorem 3 Let the n-bit vector $\boldsymbol{\omega}$ satisfies $\boldsymbol{\omega} \oplus (\boldsymbol{\omega} \gg 1) \oplus ((\boldsymbol{x} \oplus \boldsymbol{y} \oplus \boldsymbol{z}) \gg 1) = 0$, $\omega_{n-1} = 0$, where $\boldsymbol{x}, \boldsymbol{y}$ are the input masks and $\boldsymbol{z}$ is the output mask in a linear mask of addition operation. Thus we can compute the correlation by the following equation.

$$cor(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = 1_{\boldsymbol{x} \oplus \boldsymbol{z} \preccurlyeq \boldsymbol{\omega}} 1_{\boldsymbol{y} \oplus \boldsymbol{z} \preccurlyeq \boldsymbol{\omega}} (-1)^{(\boldsymbol{x} \oplus \boldsymbol{z}) \cdot (\boldsymbol{y} \oplus \boldsymbol{z})} 2^{-|\boldsymbol{\omega}|}$$

According to Theorem 3, we can know that the range of the correlation depends on the Hamming weight of $\boldsymbol{\omega}$, and the sign of correlation is determined by the value of $(\boldsymbol{x} \oplus \boldsymbol{z}) \cdot (\boldsymbol{y} \oplus \boldsymbol{z})$. To get a possible linear approximation, the vector $\boldsymbol{\omega}$ should follow the constraints below.

$$\begin{aligned}
&\omega_{n-1} = 0\\
&\omega_j = \omega_{j+1} \oplus x_{j+1} \oplus y_{j+1} \oplus z_{j+1}, 0 \le j \le n-2\\
&\left.\begin{aligned}
&x_i \vee \neg z_i \vee \omega_i = 1\\
&\neg x_i \vee z_i \vee \omega_i = 1\\
&y_i \vee \neg z_i \vee \omega_i = 1\\
&\neg y_i \vee z_i \vee \omega_i = 1
\end{aligned}\right\}, 0 \le i \le n-1
\end{aligned}$$

3. Sequential encoding method

Our target is to search for differential (linear) trails with probability (correlation) greater than 2^{-k}, and this kind of condition can always be represented as the Boolean cardinality constraint $\sum_{j=0}^{n-1} x_j \le k$, where $x_j \in \mathbb{F}_2$ and $k \in \mathbb{Z}^+$. Here, we adopt the sequential encoding approach in [39] to obtain CNF formulas.

For the Boolean cardinality constraint $\sum_{j=0}^{n-1} x_j \le k$, the SAT model is shown as follows. Among the formulas below, $t_{i,j}\,(i \in \{0,1,\ldots,n-2\}, j \in \{0,1,\ldots,k-1\})$ are some auxiliary variables we introduced.

$$\begin{aligned}
&\neg x_0 \vee t_{0,0} = 1\\
&\neg t_{0,j} = 1,\ 1 \le j \le k-1\\
&\left.\begin{aligned}
&\neg x_i \vee t_{i,0} = 1\\
&\neg t_{i-1,0} \vee t_{i,0} = 1\\
&\neg x_i \vee \neg t_{i-1,j-1} \vee t_{i,j} = 1\\
&\neg t_{i-1,j} \vee t_{i,j} = 1\\
&\neg x_i \vee \neg t_{i-1,k-1} = 1\\
&\neg x_{n-1} \vee \neg t_{n-2,k-1} = 1
\end{aligned}\right\}, 1 \le j \le k-1 \Bigg\}, 1 \le i \le n-2
\end{aligned}$$

IV. Application to CRAX

In this part, we display the DCs and LCs for CRAX, which are searched by resolving the models in Section III in an automatic manner. We adopt CaDiCaL [40] as our SAT solver, which is rooted in the conflict-driven clause learning (CDCL) algorithm [41], [42]. Then, a 4-round key recovery attack on CRAX is given at last.

1. Differential trials for CRAX

For Alzette, we obtain DCs with the MEDCP for up to 12 steps, by which we can give a 3-round differential attack for CRAX. The optimal DCs searched by us are shown in Table 3. For detail, we also list the DCs of each step of Alzette in Table 4.

Since the differential cryptanalysis of CRAX is under the premise that the inputs to the addition operation are independent and each round is also independent, we need to verify the theoretical probability by experiments. We randomly choose 100 master keys in total. In regard to every key, we calculate the experimental probability of the distinguisher with random input pairs with the fixed input differences.

Table 3 Differential characteristics for CRAX (in hex)

# Round	CRAX		
i	$\boldsymbol{\delta_L}$	$\boldsymbol{\delta_R}$	$\log_2 p_i$
0	80020100	00010080	–
1	01010000	00030101	−6
2	41030040	00808040	−32
3	a0901041	51c1b090	−21
$\sum_{i=1}^{r} \log_2 p_i$	–	–	−59

Table 4 Differential characteristics for Alzette (in hex)

# Step	Alzette		
i	$\boldsymbol{\delta_L}$	$\boldsymbol{\delta_R}$	$\log_2 p_i$
0	80020100	00010080	–
1	80000000	00010000	−2
2	00000000	00010000	0
3	00010000	00030000	−1
4	00010000	00030000	−3
5	03010202	01010302	−7
6	81800282	004043c2	−6
7	80404140	00c0c143	−9
8	41030040	00808040	−10
9	400000c0	00804000	−6
10	e0000000	00803000	−4
11	20801000	41801000	−5
12	a0901041	51c1b090	−6
$\sum_{i=1}^{r} \log_2 p_i$	–	–	−59

For 4-step Alzette (1-round CRAX), the probability we observed is $2^{-5.96}$ with a specimen size of 2^{10} plaintext pairs. For 5-step Alzette (1.25-round CRAX), we use a specimen size of 2^{16} plaintext pairs, and the experimental probability is $2^{-9.4}$ compared with the theoretical probability 2^{-10}. For 6-step Alzette (1.5-round CRAX), the experimental probability is $2^{-17.99}$ compared to the theoretical probability 2^{-18} for a specimen size of 2^{25}. For 7-step Alzette (1.75-round CRAX), we use a specimen size of 2^{30} plaintext pairs and the experimental probability is $2^{-25.69}$, compared with the theoretical probability 2^{-26}. For 8-step Alzette (2-round CRAX), the probability we observed is $2^{-33.98}$ using a specimen size of 2^{40} plaintext pairs, and the theoretical probability is 2^{-34}.

And all the comparisons between theoretical results and experimental results are listed in Table 1.

2. Linear trials for CRAX

We also obtain linear trails with the maximum expected linear characteristic correlation (MELCC) for up to 12 steps for Alzette, by which we can give a 3-round linear attack for CRAX. The best LCs found by us are

displayed in Table 5. And Table 6 shows the LCs of each step of Alzette.

Table 5 Linear characteristics for CRAX (in hex)

# Round	CRAX		
i	$\boldsymbol{\delta_L}$	$\boldsymbol{\delta_R}$	$\log_2 c_i$
0	06000201	80020107	–
1	01010001	01000101	−7
2	00000001	00000101	−15
3	85028584	84840100	−8
$\sum_{i=1}^{r} \log_2 c_i$	–	–	−30

Table 6 Linear characteristics for Alzette (in hex)

# Step	Alzette		
i	$\boldsymbol{\delta_L}$	$\boldsymbol{\delta_R}$	$\log_2 c_i$
0	06000201	80020107	–
1	00020101	02020007	−3
2	00000181	00000001	−2
3	00000181	00000100	−1
4	01010001	01000101	−1
5	0080c0c0	81c0c101	−2
6	81814041	c040c080	−4
7	01018001	010180c1	−6
8	00000001	00000101	−3
9	01800000	80000101	0
10	07030000	80000301	−1
11	c4030180	84020301	−3
12	85028584	84840100	−4
$\sum_{i=1}^{r} \log_2 c_i$	–	–	−30

Since we do not make independent assumptions in linear cryptanalysis, we don't need to do experiments to verify the theoretical correlations.

3. Key recovery attack on CRAX

In this subsection, we propose a 4-round key-recovery attack for CRAX. At first, we find a same 6-step differential-linear distinguisher of Alzette as in [31], the experimental correlation of which is $2^{-1.45}$. The input difference is

$$\Delta = 8000000000000000$$

and the output mask is

$$\boldsymbol{\lambda} = 0000004000200000$$

Then we fix the same $\boldsymbol{\lambda}$ as the input mask of CRAX and search for the 0.5-round linear approximations. The output mask with the highest correlation is

$$\boldsymbol{\lambda}_{\text{out}} = 040b10040a06020$$

The correlation computed is 2^{-4}. Therefore, we could get a 2-round differential-linear distinguisher for CRAX by combining these two distinguishers, where correlation is $2^{-1.45} \times 2^{-4} \times 2^{-4} = 2^{-9.45}$.

Let the input pair $(\boldsymbol{P}_{\text{in}}, \boldsymbol{P}'_{\text{in}})$ be generated by the fixed difference

$$\Delta \boldsymbol{P}_{\text{in}} = \boldsymbol{P}_{\text{in}} \oplus \boldsymbol{P}'_{\text{in}} = 8000000000000000$$

where $\boldsymbol{P}_{\text{in}}$ is a random plaintext. Let $(\boldsymbol{C}_{\text{out}}, \boldsymbol{C}'_{\text{out}})$ be the corresponding ciphertexts after 4-round CRAX. As Figure 4 shows, let $\boldsymbol{u}_i$ signify the input value, and $\boldsymbol{v}_i$ represent the output value of the i-th step of Alzette. According to our 2-round differential-linear distinguisher for CRAX, the following condition holds with the correlation $2^{-9.45}$:

$$\boldsymbol{\lambda}_{\text{out}} \cdot \boldsymbol{v}_2 = \boldsymbol{\lambda}_{\text{out}} \cdot \boldsymbol{v}'_2 \tag{1}$$

where $\boldsymbol{\lambda}_{\text{out}} = 040b10040a06020$.

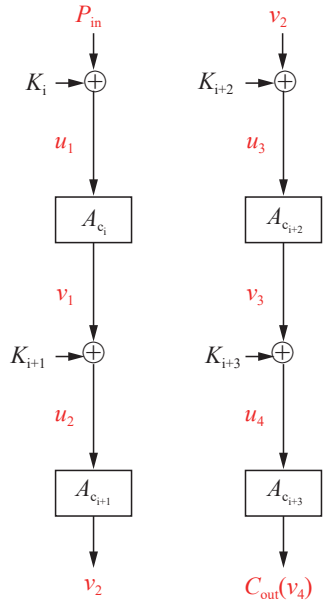


Figure 4 Four rounds of CRAX.

The goal of this attack is to decrypt $(\boldsymbol{C}_{\text{out}}, \boldsymbol{C}'_{\text{out}})$ by one total round and an Alzette. Then, for each of candidates for $(\boldsymbol{u}_3, \boldsymbol{u}'_3)$ the condition (1) will be checked to recover the last-round key used in the decryption.

In light of Matsui's rule of thumb, nearly $m = 2^3 \times (\frac{cor}{2})^{-2}$ pairs of plaintexts are needed. That is, we need to generate 2^{24} pairs of $(\boldsymbol{P}_{\text{in}}, \boldsymbol{P}'_{\text{in}})$ with the difference $\Delta \boldsymbol{P}_{\text{in}}$ randomly. The last-round key is 64 bit, thus we can decrypt $(\boldsymbol{C}_{\text{out}}, \boldsymbol{C}'_{\text{out}})$ by exhaustively searching for 2^{64} times and get $(\boldsymbol{v}_3, \boldsymbol{v}'_3)$. Then, we put an inverse Alzette permutation to $(\boldsymbol{v}_3, \boldsymbol{v}'_3)$ and get $(\boldsymbol{u}_3, \boldsymbol{u}'_3)$. Since the XOR of key doesn't change the value of $\boldsymbol{v}_2 \oplus \boldsymbol{v}'_2$, the following condition holds

$$\boldsymbol{\lambda}_{\text{out}} \cdot \boldsymbol{u}_3 = \boldsymbol{\lambda}_{\text{out}} \cdot \boldsymbol{u}'_3 \tag{2}$$

with the correlation $2^{-9.45}$ when the guessed key is the right key. We can use a counter to record how many

pairs of $(\boldsymbol{u}_3, \boldsymbol{u}'_3)$ conforming to condition (2) for each key. Then we choose the key which gets the highest scores.

Next we analyze the complexity in the key-recovery phase. We totally choose $2 \times 2^{24} = 2^{25}$ plaintexts, thus the data complexity is 2^{25}. For each pair of $(\boldsymbol{C}_{\text{out}}, \boldsymbol{C}'_{\text{out}})$, we decrypt them by 1 round and guess the last-round key, then we put an inverse Alzette to $(\boldsymbol{v}_3, \boldsymbol{v}'_3)$. It can be seen as the decryption by two rounds. Since we use the exhaustive search algorithm when guessing K_{i+3}, the time complexity is $2^{25} \times 2^{64} \times 2 \times 2^{-1} = 2^{89}$ times of 4-round encryption. When we recover the remaining 64-bit key, we can do exhaustive search for 2^{64} times. To recover the whole 128-bit of master key, the time complexity is $2^{89} + 2^{64} = 2^{89}$ times of 4-round encryption.

V. Conclusion

In this paper, we research the differential and linear properties of modular addition, and construct the SAT models to search for DCs and LCs for CRAX automatically. Since the cryptanalysis for CRAX follows that of Alzette, we can contrast our results with existing attacks for Alzette. When compared with the previous attacks for Alzette, our distinguished differential characteristics and linear masks are both extended for 4 steps. Thus, we obtain the optimal DCs and LCs for 3 rounds of CRAX for the first time. In addition, we give a 4-round key recovery attack for CRAX. The time complexity is 2^{89} times of 4-round encryption and the data complexity is 2^{25}.

Acknowledgement

This work was supported by the National Key R&D Program of China (Grant No. 2018YFA0704704), the Natural Science Foundation of China (Grant No. 61772519), and the Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ201801 02).

References

[1] D. Hong, J. Sung, S. Hong, *et al.*, "HIGHT: A new block cipher suitable for low-resource device," in *Proceedings of the 8th International Workshop on Cryptographic Hardware and Embedded Systems - CHES*, Yokohama, Japan, pp. 46–59, 2006.

[2] D. Dinu, L. Perrin, A. Udovenko, *et al.*, "Design strategies for ARX with provable bounds: SPARX and LAX," in *Proceedings of the 22nd International Conference on Advances in Cryptology*, Hanoi, Vietnam, pp. 484–513, 2016.

[3] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Proceedings of the 2nd International Workshop on Fast Software Encryption*, Leuven, Belgium, pp. 363–366, 1995.

[4] D. Wheeler and R. Needham, *TEA Extensions*. Cambridge: University of Cambridge, 1997.

[5] B. Koo, D. Roh, H. Kim, *et al.*, "CHAM: A family of lightweight block ciphers for resource-constrained devices," in *Proceedings of the 20th International Conference on Information Security and Cryptology -ICISC 2017*, Seoul, South Korea, pp. 3–25, 2018.

[6] R. Beaulieu, D. Shors, J. Smith, *et al.*, "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the 52nd Annual Design Automation Conference*, San Francisco, CA, USA, article no.175, 2015.

[7] D. J. Bernstein, "The Salsa20 family of stream ciphers," in *New Stream Cipher Designs*, M. Robshaw, O. Billet, Eds. Springer, Berlin, Germany, pp. 84–97, 2008.

[8] D. J. Bernstein, "ChaCha, a variant of Salsa20," *Workshop Record of SASC*. Available at: https://www.mendeley.com/catalogue/27343fad-085e-3a7f-a002-7541bba412d6/, 2008.

[9] N. Ferguson, S. Lucks, B. Schneier, *et al.*, *The Skein Hash Function Family*. NIST, 2010

[10] J. P. Aumasson, L. Henzen, W. Meier, *et al.*, *SHA-3 Proposal BLAKE*. ETH Zürich, 2008

[11] C. Beierle, A. Biryukov, L. C. Dos Santos, *et al.*, "Alzette: A 64-Bit ARX-box," in *Proceedings of the 40th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp. 419–448, 2020.

[12] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[13] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, Lofthus, Norway, pp. 386–397, 1994.

[14] A. Bogdanov and V. Rijmen, "Linear hulls with correlation zero and linear cryptanalysis of block ciphers," *Designs, Codes and Cryptography*, vol. 70, no. 3, pp. 369–383, 2014.

[15] L. R. Knudsen, "Truncated and higher order differentials," in *Proceedings of the Second International Workshop on Fast Software Encryption*, Leuven, Belgium, pp. 196–211, 1995.

[16] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, Prague, Czech Republic, pp. 12–23, 1999.

[17] S. K. Langford and M. E. Hellman, "Differential-linear cryptanalysis," in *Proceedings of the 14th Annual International Cryptology Conference*, Santa Barbara, CA, USA, pp. 17–25, 1994.

[18] M. Matsui, "On correlation between the order of s-boxes and the strength of DES," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, Perugia, Italy, pp. 366–375, 1995.

[19] N. Mouha, Q. J. Wang, D. W. Gu, *et al.*, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Proceedings of the 7th International Conference on Information Security and Cryptology*, Beijing, China, pp. 57–76, 2011.

[20] S. B. Wu and M. S. Wang, *Security Evaluation Against Differential Cryptanalysis for Block Cipher Structures*, in Press, 2011.

[21] S. W. Sun, L. Hu, M. Q. Wang, *et al.*, *Towards Finding the Best Characteristics of Some Bit-Oriented Block Ciphers and Automatic Enumeration of (Related-Key) Differential and Linear Characteristics with Predefined Properties*, in Press, 2014.

[22] S. W. Sun, L. Hu, P. Wang, *et al.*, "Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers," in *Proceedings of the 20th International Conference on Advances in Cryptology*, Kaoshiung, China, pp. 158–178, 2014.

[23] Y. X. Cui, H. Xu, W. F. Qi, "MILP-based linear attacks on round-reduced GIFT," *Chinese Journal of Electronics*, vol. 31, no. 1, pp. 89–98, 2022.

[24] M. Soos, K. Nohl, and C. Castelluccia, "Extending SAT solvers to cryptographic problems," in *Proceedings of the 12th International Conference on Theory and Applications*

of Satisfiability Testing, Swansea, UK, pp. 244–257, 2009.
[25] S. Kölbl, G. Leander, and T. Tiessen, "Observations on the SIMON block cipher family," in *Proceedings of the 35th Annual Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp. 161–185, 2015.
[26] Y. W. Liu, Q. J. Wang, and V. Rijmen, "Automatic search of linear trails in ARX with applications to SPECK and Chaskey," in *Proceedings of the 14th International Conference on Applied Cryptography and Network Security*, Guildford, UK, pp. 485–499, 2016.
[27] L. Sun, W. Wang, and M. Q. Wang, "Automatic search of bit-based division property for ARX ciphers and word-based division property," in *Proceedings of the 23rd International Conference on Advances in Cryptology*, Hong Kong, China, pp. 128–157, 2017.
[28] A. Biryukov, V. Velichkov, and Y. Le Corre, "Automatic search for the best trails in ARX: Application to block cipher SPECK," in *Proceedings of the 23rd International Conference on Fast Software Encryption*, Bochum, Germany, pp. 289–310, 2016.
[29] K. Fu, M. Q. Wang, Y. H. Guo, *et al.*, "MILP-based automatic search algorithms for differential and linear trails for speck," in *Proceedings of the 23rd International Conference on Fast Software Encryption*, Bochum, Germany, pp. 268–288, 2016.
[30] Y. W. Liu, S. W. Sun, and C. Li, "Rotational cryptanalysis from a differential-linear perspective," in *Proceedings of the 40th Annual International Conference on Advances in Cryptology*, Zagreb, Croatia, pp. 741–770, 2021.
[31] Z. F. Niu, S. W. Sun, Y. W. Liu, *et al.*, "Rotational differential-linear distinguishers of ARX Ciphers with arbitrary output linear masks," in *Proceedings of the 42nd Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp. 3–32, 2022.
[32] Z. Xu, Y. Q. Li, and M. S. Wang, "Security analysis of Alzette," *Journal of Cryptologic Research*, vol. 9, no. 4, pp. 698–708, 2022. (in Chinese)
[33] L. Sun, W. Wang, and M. Q. Wang, "Accelerating the search of differential and linear characteristics with the sat method," *IACR Transactions on Symmetric Cryptology*, vol. 2021, no. 1, pp. 269–315, 2021.
[34] B. Sun, Z. Q. Liu, V. Rijmen, *et al.*, "Links among impossible differential, integral and zero correlation linear cryptanalysis," in *Proceedings of the 35th Annual Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp. 95–115, 2015.
[35] H. Lipmaa and S. Moriai, "Efficient algorithms for computing differential properties of addition," in *Proceedings of the 8th International Workshop on Fast Software Encryption*, Yokohama, Japan, pp. 336–350, 2002.
[36] K. Nyberg and J. Wallén, "Improved linear distinguishers for SNOW 2.0," in *Proceedings of the 13th International Workshop on Fast Software Encryption*, Graz, Austria, pp. 144–162, 2006.
[37] J. Wallén, "Linear approximations of addition modulo 2^n," in *Proceedings of the 10th International Workshop on Fast Software Encryption*, Lund, Sweden, pp. 261–273, 2003.
[38] E. Schulte-Geers, "On CCZ-equivalence of addition mod 2^n," *Designs, Codes and Cryptography*, vol. 66, no. 1-3, pp. 111–127, 2013.
[39] N. Mouha and B. Preneel, *Towards Finding Optimal Differential Characteristics for ARX: Application to Salsa20*. 2013
[40] A. Biere, "CaDiCaL at the SAT Race 2019," in *Proceedings of the SAT Race 2019 – Solver and Benchmark Descriptions*, M. Heule, M. Järvisalo, and M. Suda, Ed. University of Helsinki, Helsinki, pp. 8–9, 2019.
[41] J. P. Marques Silva and K. A. Sakallah, "GRASP - A new search algorithm for satisfiability," in *Proceedings of the International Conference on Computer Aided Design*, San Jose, CA, USA, pp. 220–227, 1996.
[42] R. J. Bayardo and R. C. Schrag, "Using CSP look-back techniques to solve real-world SAT instances," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, Providence, RL, USA, pp. 203–208, 1997.

Yiyi HAN is an M.S. candidate of Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. Her research interests include symmetric cryptanalysis and design.
(Email: hanyiyi@iie.ac.cn)

Caibing WANG is a Ph.D. candidate of Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. Her research interests include symmetric cryptanalysis and design.
(Email: wangcaibing@iie.ac.cn)

Zhongfeng NIU is a Ph.D. candidate of Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include symmetric cryptanalysis and design.
(Email: niuzhongfeng@iie.ac.cn)

Lei HU received the Ph.D. degree from Chinese Academy of Sciences in 1994. He is a Professor in Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include basic theory of applications of pseudorandom sequences and arrays, analysis of cryptographic algorithms and theoretical cryptography.
(Email: hulei@iie.ac.cn)