Cube-Attack-Like Cryptanalysis of Round-Reduced Keccak Using MILP

Ling Song[1,2] and Jian Guo[2]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, China

[2] Nanyang Technological University, Singapore

songling@ntu.edu.sg,guojian@ntu.edu.sg

Abstract. Cube-attack-like cryptanalysis on round-reduced Keccak was proposed by Dinur *et al.* at EUROCRYPT 2015. It recovers the key through two phases: the preprocessing phase for precomputing a look-up table and online phase for querying the output and getting the cube sum with which the right key can be retrieved by looking up the precomputed table. It was shown that such attacks are efficient specifically for Keccak-based constructions with small nonce or message block size. In this paper, we provide a mixed integer linear programming (MILP) model for cube-attack-like cryptanalysis on keyed Keccak, which does not impose any unnecessary constraint on cube variables and finds almost optimal cubes by balancing the two phases of cube-attack-like cryptanalysis. Our model is applied to Ketje Jr, Ketje Sr, a Xoodoo-based authenticated encryption and Keccak-MAC-512, all of which have a relatively small nonce or message block size. As a result, time complexities of 5-round attacks on Ketje Jr and 7-round attacks on Ketje Sr can be improved significantly. Meanwhile, 6-round attacks, one more round than the previous best attack, are possible if the key size of Ketje V1 (V2) is reduced to 72 (80) bits. For Xoodoo-based AE in Ketje style, the attack reaches 6 rounds. Additionally, a 7-round attack of Keccak-MAC-512 is achieved. To verify the correctness of our attacks, a 5-round attack on Ketje V1 is implemented and tested practically. It is noted that this work does not threaten the security of any Keccak-based construction.

Keywords: Ketje, Xoodoo, Keccak-MAC, cube attack, auxiliary variable, MILP

1 Introduction

The Keccak hash function [BDPV11] was designed by Bertoni *et al.* and selected as the Secure Hash Algorithm-3 (`SHA-3`) of the National Institute of Standards and Technology of the U.S. (NIST) in 2012. The formal standardization was made public in 2015 [The15]. As a new standard, it has attracted intensive cryptanalysis from the community regarding collision, preimage, and second-preimage resistance [NRM11, MS13, DDS12, DDS13, GLS16, QSLG17, SLG17]. Up to date, practical collision (preimage) attacks on Keccak reduced up to 6 (4) out of 24 rounds were achieved.

Apart from the keyless hash function, Keccak can be used under keyed modes, such as message authentication codes (MAC), stream ciphers, etc. What's more, the Keccak permutation or its variant has been employed in other designs, such as authenticated encryptions (AE) Keyak [BDP+16b], Ketje [BDP+16a] and the pseudorandom function Kravatte [BDH+17b]. Recently, a new permutation Xoodoo similar to the Keccak permutation has been proposed [DHAK18] and one of its purposes is to construct AE in

KETJE style. In the literature, there is a line of cryptanalysis focusing on keyed KECCAK-based constructions. In [DMP+15], Dinur *et al.* analyzed keyed KECCAK with cube attacks [DS09]. Specifically, key recovery attacks and forgery attacks were mounted against KEYAK and KECCAK used as MAC and stream ciphers with reduced rounds. Particularly, a type of cube attacks (cube-attack-like cryptanalysis) which proceeds in preprocessing and online phases was proposed. In the cube-attack-like cryptanalysis, auxiliary variables which are supposed to be equal to certain key bits are used to balance the two phases such that the time complexity of the whole attack is reduced. Following [DMP+15], Dong *et al.* provided cube-attack-like cryptanalysis on round-reduced KETJE in [DLWQ17], where dynamic variables inspired by dynamic cube attacks [DS11] are used. Auxiliary variables help reduce the diffusion of key bits, whereas dynamic variables, which depend on some of the cube variables and some key bits, help reduce the diffusion of both key bits and cube variables. In [HWX+17], Huang *et al.* proposed conditional cube attacks for KECCAK where the propagation of cube variables is controlled under conditions in the first two rounds, resulting in improved attacks on round-reduced KEYAK and KECCAK used as MAC. In [HWX+17], conditional cubes were obtained through a program that has not been optimized, which allows further improvements via a mixed integer linear programming (MILP) model [LBDW17]. MILP-based methods have become popular in the search for differential/linear characteristics since Mouha *et al.*'s pioneering work [MWGP11]. However, it is the first time that MILP has been applied to cube attacks on keyed KECCAK. Later, a new MILP model for searching conditional cubes was proposed in [SGSL17], showing further improvements on attacks against most keyed KECCAK constructions, such as KECCAK used as MAC, KEYAK and KETJE except the smallest instance of KETJE, KETJE Jr. Instead of analyzing a round-reduced target in cube attacks, Fuhr *et al.* [FNR18] proposed a state-recovery attack against full KETJE Jr with increased rate size. Against KRAVATTE, algebraic attacks which utilize its structural properties were proposed in [CFG+18].

As can be seen from the previous works [DLWQ17, SGSL17], cube-attack-like cryptanalysis has an advantage over conditional cube attacks in analyzing keyed KECCAK-based constructions with small degrees of freedom, *i.e.*, small message block size or nonce size. On the other hand, MILP-based methods have shown their efficiency in conditional cube attacks with significantly improved results. In this paper, we take the advantage of this efficiency and apply it to cube-attack-like cryptanalysis on keyed KECCAK-based constructions with small degrees of freedom.

Our contributions. We develop techniques for building an MILP model for cube-attack-like cryptanalysis, which takes both auxiliary and dynamic variables into consideration and aims to find almost optimal attacks by balancing the two phases of cube-attack-like cryptanalysis. In many of previous works, cube variables are forced to be from the so-called column-parity-like (CP-like) kernel, while our model does not impose any unnecessary constraint on cube variables, and hence finds optimal cubes in terms of dimension. With regard to attack complexities, cubes found by our model are almost optimal. We apply our MILP model to keyed KECCAK constructions with small nonce or message block length, including two smaller versions of KETJE Jr, KETJE Sr, a XOODOO-based AE and KECCAK-MAC-512. The results are as follows.

- Improved 5-round attacks on KETJE Jr V1 and V2, with time complexity significantly reduced;
- 6-round attacks on KETJE V1 and V2, with key size reduced to 72 and 80 bits;
- Improved 7-round attack on KETJE Sr V2;
- 6-round attack on XOODOO-based AE in KETJE style;
- 7-round attack on KECCAK-MAC-512.

Table 1: Summary of our attacks on Ketje Jr, Ketje Sr, Xoodoo and Keccak-MAC-512 under the nonce respected setting and comparison with related works

Target	b	$\|K\|$	DF[†]	Rounds	T	M	Source	Type[‡]
Ketje Jr V1	200	96	86	5/13	$2^{36.86}$	2^{18}	Sect. 5.1	CAL_1
	200	72	110	6/13	$2^{68.04}$	2^{34}	Sect. 5.3	
Ketje Jr V2	200	96	86	5/13	$2^{34.91}$	2^{15}	Sect. 5.2	
	200	80	102	6/13	$2^{59.17}$	2^{25}	Sect. 5.4	
Ketje Sr V1	400	128	254	7/13	2^{114}	2^{48}	Sect. 5.5	
Ketje Sr V2	400	128	254	7/13	2^{99}	2^{33}		
Xoodoo	384	128	238	6/-	2^{89}	2^{55}	Sect. 5.6	
Keccak-MAC-512	1600	128	447	7/24	2^{111}	2^{46}	Sect. 5.7	
Ketje Jr V1	200	96	86	5/13	2^{56}	2^{38}	[DLWQ17]	CAL_2
Ketje Jr V2	200	96	86	5/13	$2^{50.32}$	2^{32}		
Ketje Sr V1	400	128	254	7/13	$2^{115.32}$	2^{50}		
Ketje Sr V2	400	128	254	7/13	$2^{113.58}$	2^{48}		
Lake Keyak	1600	128	1200	8/12	$2^{71.01}$	-	[SGSL17]	CC
Ketje Major	1600	128	1454	7/13	$2^{71.24}$	-		
Ketje Minor	800	128	654	7/13	$2^{73.03}$	-		
Ketje Sr V1	400	128	254	7/13	2^{91}	-		
Keccak-MAC-512	1600	128	447	6/24	2^{40}	-		
Lake Keyak	1600	128	1200	8/12	$2^{79.6}$	2^{14}	[BDL+18]	CAL_1
Ketje Major	1600	128	1454	7/13	2^{94}	2^{29}		
Ketje Minor	800	128	654	7/13	2^{113}	2^{48}		
Keccak-MAC-512	1600	128	447	7/24	$2^{112.6}$	2^{47}		

[†] DF Degrees of freedom
[‡] CC Conditional cube attacks
CAL_1 Cube-attack-like cryptanalysis with the help of MILP
CAL_2 Cube-attack-like cryptanalysis without the help of MILP

The results are summarized in Table 1. It is worth noticing that more rounds can be attacked against Ketje Jr when the key size is reduced, *i.e.*, the security is reduced. Although 72 bits or 80 bits are not the recommended key size by the designers, it is good to see how the security is affected by varying the key/nonce sizes. For Ketje Sr V1, Ketje Major and Minor which have a relatively large nonce size, cube-attack-like cryptanalysis does not outperform conditional cube attacks. In addition, our analysis shows that Xoodoo-based AE bears good resistance against cube-attack-like cryptanalysis.

Comparison with Bi *et al.*'s model. Concurrently, another model for cube-attack-like cryptanalysis on keyed Keccak was proposed by Bi *et al.* [BDL+18]. Bi *et al.*'s model utilizes auxiliary variables and finds cubes in the CP-like kernel with low complexity for the preprocessing phase. Balancing the two phases is processed independently from the model. In contrast, our model utilizes both auxiliary and dynamic variables and imposes no extra constraint on cube variables (thus covers the full set of solutions with respect to dimension). Moreover, balancing is considered inside the model. Even though both models are general to keyed Keccak constructions, our targets differentiate from those of Bi *et al.*'s. Specifically, Bi *et al.* focus on Keccak-MAC, Keyak and two larger versions of Ketje, which have relatively large degrees of freedom, while our targets are the smaller versions of Ketje, namely Ketje Jr, Ketje Sr and a Xoodoo-based AE.We also apply our model to Keccak-MAC-512 and a slightly better result is obtained than

that from [BDL+18].

Organization. The rest of the paper is organized as follows. In Section 2, a brief description of KETJE, XOODOO and KECCAK-MAC is given, followed by an introduction of related works. The MILP model is sketched in Section 4, and its application to KETJE Jr, KETJE Sr, a XOODOO-based AE and KECCAK-MAC-512 is provided in Section 5. A comparison with related works is provided in Section 6. We conclude the paper in Section 7.

2 Description of KETJE and KECCAK-MAC

2.1 Notations

$\boldsymbol{c}$	Capacity of a sponge function
$\boldsymbol{r}$	Rate of a sponge function
$\boldsymbol{b}$	Width of a KECCAK permutation in bits, $\boldsymbol{b} = \boldsymbol{r} + \boldsymbol{c}$
$\boldsymbol{n}$	Number of rounds in KECCAK-p
d	Dimension of the cube
n_i	Number of involved key bits
n_a	Number of auxiliary variables
n_k	Number of recovered key bits
$\oplus$	XOR operation
$\cdot$	AND operation

2.2 KECCAK-p

The KECCAK-p permutations, denoted by KECCAK-$p[\boldsymbol{b}, \boldsymbol{n}]$, are specified with two parameters: the width of the permutation in bits $\boldsymbol{b} = 25 \times 2^l, l = 0, \cdots, 6$, and the number of rounds $\boldsymbol{n}$. The $\boldsymbol{b}$-bit state a of the KECCAK-$p[\boldsymbol{b}, \boldsymbol{n}]$ can be seen as a three-dimensional array of bit $a[5][5][w]$ with $w = 2^l$. The expression $a[x][y][z]$ represents the bit at position (x, y, z), where expressions in the x and y coordinates are always implicitly taken modulo 5 and expressions in the z coordinate modulo w.

The two-dimensional part $a[x][*][*]$ is called a *sheet*. The one-dimensional part $a[*][y][z]$ is called a *row*, $a[x][*][z]$ a *column* and $a[x][y][*]$ a *lane*. A lane of the state is also denoted as $a[x][y]$ by omitting the z index. At lane level, the state $a[x][y]$ becomes a 5×5 array with x for the column index and y for the row index. These notations are visualized in Figure 1.

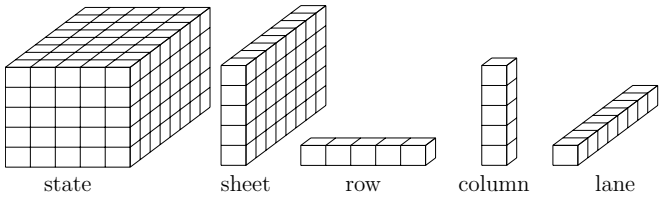


Figure 1: Notations of KECCAK-p

The round function of KECCAK-$p[\boldsymbol{b}, \boldsymbol{n}]$ consists of five steps $\mathtt{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$, with details as follows.

θ: $a[x][y][z] = a[x][y][z] \oplus \bigoplus_{y=0}^{4} a[x-1][y][z] \oplus \bigoplus_{y=0}^{4} a[x+1][y][z-1]$.

ρ: $a[x][y][z] = a[x][y][(z - T(x, y))]$, where $T(x, y)s$ are rotation constants.

π: $a[y][2x+3y][z] = a[x][y][z]$.

χ: $a[x][y][z] = a[x][y][z] \oplus ((a[x+1][y][z] \oplus 1) \cdot a[x+2][y][z])$.

ι: $a[0][0] = a[0][0] \oplus RC_{i_r}$, where RC_{i_r} is the i_r-th round constant.

In the specification of KETJE V2, the twisted permutations KECCAK-$p^\star$ are defined as

$$\text{KECCAK-}p^\star[\boldsymbol{b}, \boldsymbol{n}] = \pi \circ \text{KECCAK-}p[\boldsymbol{b}, \boldsymbol{n}] \circ \pi^{-1},$$

where π^{-1} is the inverse of the step mapping π which is expressed by

$$\pi^{-1}: \quad a[x+3y][x][z] = a[x][y][z].$$

The twist is to re-order the lanes in the state, as shown in Figure 2, so the twisted permutation is considered to apply the original permutation to the state $\pi^{-1}(a)$.

0,0	1,0	2,0	3,0	4,0
0,1	1,1	2,1	3,1	4,1
0,2	1,2	2,2	3,2	4,2
0,3	1,3	2,3	3,3	4,3
0,4	1,4	2,4	3,4	4,4

$\xrightarrow{\pi^{-1}}$

0,0	0,2	0,4	0,1	0,3
1,3	1,0	1,2	1,4	1,1
2,1	2,3	2,0	2,2	2,4
3,4	3,1	3,3	3,0	3,2
4,2	4,4	4,1	4,3	4,0

Figure 2: π^{-1}

2.3 KETJE

KETJE is a set of authenticated encryption functions built on KECCAK-p. KETJE Jr and KETJE Sr are proposed in KETJE V1, of state size 200 and 400 bits respectively. Later, two larger instances, KETJE Minor and KETJE Major with 800-bit and 1600-bit state respectively, are added to the set in KETJE V2. The major difference between KETJE V1 and V2 is that KECCAK-p is used in KETJE V1 while KECCAK-$p^\star$ is used instead in KETJE V2. In the following, we give a brief description of KETJE V2.

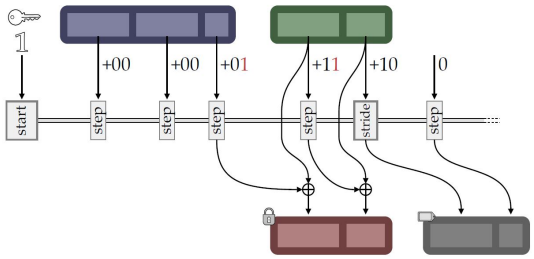


Figure 3: Wrapping a header and a body with MonkeyWrap [BDP+16a]

The structure of KETJE follows the MonkeyWrap, as illustrated in Figure 3. Like other authenticated encryption functions, KETJE proceeds in four phases.

- *Initialization* The state is initialized with the packed key, the nonce N and some paddings. Then KECCAK-$p^\star[\boldsymbol{b}, \boldsymbol{n}_{start}]$ is applied.

- *Processing associated data* The associated data is split into δ-bit blocks (except the last one). Each time an associated data block of length up to δ bits is padded to $\delta+4$ bits and XORed to the state, followed by the application of KECCAK-$p^\star[\boldsymbol{b}, \boldsymbol{n}_{step}]$. If the associated data is empty, then a single block padded from the empty string will be processed.
- *Processing message* The message is also processed in δ-bit blocks in a similar way, where the ciphertext block is generated by XORing the message block and δ bits of the internal state before the message block is absorbed.
- *Finalization* KECCAK-$p^\star[\boldsymbol{b}, \boldsymbol{n}_{stride}]$ is used to generate δ bits. If δ is greater than the required tag length, then the tag is extracted from the δ bits; otherwise, KECCAK-$p^\star[\boldsymbol{b}, \boldsymbol{n}_{step}]$ is applied until enough bits are collected for generating the tag.

The parameters of the four instances of KETJE V2 are summarized in Table 2, and for all four instances, $\boldsymbol{n}_{start} = 12$, $\boldsymbol{n}_{step} = 1$ and $\boldsymbol{n}_{stride} = 6$. As can be seen from the above phases, the first ciphertext block is generated after at least $\boldsymbol{n}_{start} + \boldsymbol{n}_{step} = 13$ rounds. Most attacks on KETJE in the literature, as well as this paper, consider versions of KETJE with this number reduced.

Table 2: Four instances of KETJE V2

Name	Key size	Permutation	δ	Confidentiality
KETJE Jr	96	KECCAK-$p^\star$[200]	16	$\min(96, \lvert K\rvert)$
KETJE Sr	128	KECCAK-$p^\star$[400]	32	$\min(128, \lvert K\rvert)$
KETJE Minor	128	KECCAK-$p^\star$[800]	128	$\min(128, \lvert K\rvert)$
KETJE Major	128	KECCAK-$p^\star$[1600]	256	$\min(128, \lvert K\rvert)$

2.4 XOODOO

XOODOO [DHAK18] is a 384-bit permutation designed by Daemen *et al.*. The design is very similar to KECCAK-p, but also inspired by the permutation Gimli [BKL+17] which also uses a 384-bit state and works efficiently on many platforms.

The 384-bit state of XOODOO can be seen as a three-dimensional array of bit $a[4][3][w]$, where $w = 32$. The round function of XOODOO has five steps as follows.

θ: $a[x][y][z] = a[x][y][z] \oplus \bigoplus_{y=0}^{2} a[x-1][y][z-5] \oplus \bigoplus_{y=0}^{2} a[x-1][y][z-14]$.

ρ_{west}: $a[x][1][z] = a[x-1][1][z], a[x][2][z] = a[x][2][z-11]$.

ι: $a[0][0] = a[0][0] \oplus RC_{i_r}$, where RC_{i_r} is the i_r-th round constant.

χ: $a[x][y][z] = a[x][y][z] \oplus ((a[x][y+1][z] \oplus 1) \cdot a[x][y+2][z])$.

ρ_{east}: $a[x][1][z] = a[x][1][z-1], a[x][2][z] = a[x-2][2][z-8]$.

It is noted in [BDH+17a] that XOODOO can be used as authenticated encryption in KETJE style.

2.5 KECCAK-MAC-512

KECCAK follows the sponge construction [BDPVA11] and uses KECCAK-p[1600, 24] as the underlying permutation. The sponge construction has two parameters, the capacity $\boldsymbol{c}$ and bit rate $\boldsymbol{r}$. At first, the state is initialized to 0. Then KECCAK takes in a message M and outputs a digest. The message M is processed by splitting it into $\boldsymbol{r}$-bit blocks which are absorbed to the first $\boldsymbol{r}$ bits of the state iteratively followed by the application of

KECCAK-$p[1600, 24]$. In [BDPA11], it is proposed that KECCAK can be used as MAC by taking $K||M$ as input. Such a MAC was called KECCAK-MAC in [HWX+17] for the first time and its round-reduced versions were analyzed in papers such as [DMP+15,LBDW17], where the key size is 128 bits no matter which instance of KECCAK is used. `KMAC` [The16] is the NIST recommendation for constructing MAC from KECCAK where the key is processed as an independent block before processing the message. In this paper, we only focus on KECCAK-MAC-512, *i.e.*, the MAC based on KECCAK-512.

3 Related Works

3.1 Cube Attacks

The cube attack, which can be seen as a variant of higher order differential attacks, was introduced by Dinur and Shamir [DS09] in 2009. It treats the output bit of a cipher as an unknown Boolean polynomial $f(k_0, ..., k_{n-1}, v_0, ..., v_{m-1})$ where $k_0, ..., k_{n-1}$ are secret input variables and $v_0, ..., v_{m-1}$ are public input variables. Given any monomial t_I which is the product of variables in $I = \{i_1, ..., i_d\}$, f can be represented as the sum of terms which are supersets of I and terms which are not supersets of I:

$$f(k_0, ..., k_{n-1}, v_0, ..., v_{m-1}) = t_I \cdot p_{S_I} + q(k_0, ..., k_{n-1}, v_0, ..., v_{m-1}),$$

where p_{S_I} is called the superpoly of I in f, and $v_{i_1}, ..., v_{i_d}$ are called *cube variables*.

The idea behind cube attacks is that the sum of the Boolean polynomial $f(k_0, ..., k_{n-1}, v_0, ..., v_{m-1})$ over the cube which contains all possible values for the cube variables is exactly p_{S_I}, while this is a random function for a random polynomial. In cube attacks, low-degree p_{S_I}s in secret variables are exploited to recovery the key, while cube testers [ADMS09] work by distinguishing p_{S_I} from a random function (e.g., $p_{S_I} = 0$).

Dynamic cube attacks [DS11] are an extension of cube testers where certain variables (called *dynamic variables*) are assigned a function that depends on some of the cube variables and some private variables (the key bits), so that the output polynomial can be simplified and the cube attack can be improved.

3.2 Cube-Attack-Like Cryptanalysis on Round-Reduced KECCAK

In [DMP+15], Dinur *et al.* proposed cube-attack-like cryptanalysis on round-reduced KECCAK-MAC and KEYAK, where the key is recovered in a divide-and-conquer manner. Specifically, the idea in the attack is to choose the cube variables in a way such that the superpoly involves only a small number of key bits, whose value can be recovered independently of the rest key using a cube attack separated into preprocessing and online phases. Once the cube is selected, then

- in the *preprocessing phase*, one is to build a look-up table that stores cube sums under all possible values of the involved key bits;
- in the *online phase*, one queries the cipher and obtains the cube sum, with which the actual value of the involved key bits can be retrieved from the look-up table.

Suppose the dimension of the cube is d and the number of involved key bits is n_i. Then the time complexities of the above two phases are 2^{d+n_i} and 2^d, respectively. As can be seen, the preprocessing phase is much more expensive than the online phase. In order to tradeoff the complexity of the preprocessing and online phases, auxiliary variables are introduced. Auxiliary variables are selected from public variables and supposed to be equal to certain key bits (the XOR of key bits in a column for KECCAK), which help reduce the diffusion of key bits, and thus reduce the number of key bits n_i the cube

sum involves. Suppose there are n_a auxiliary variables. Then in the online phase, one has to guess the key bits involved in the auxiliary variables and set the auxiliary variables accordingly. Under each setting of the auxiliary variables, one queries the cipher to obtain the cube sum. Consequently, the time complexity of the online phase is increased by a factor of 2^{n_a}. However, balanced attacks become more efficient.

Following this line, Dong *et al.* [DLWQ17] studied the cube-attack-like cryptanalysis against round-reduced initialization of KETJE, where dynamic variables were used instead. They showed that dynamic variables are more effective than auxiliary variables since dynamic variables not only reduce the diffusion of key bits, but also reduce the diffusion of cube variables, potentially leading to cubes with larger dimensions. As a demonstration, attacks on 7-round KETJE Sr and 5-round KETJE Jr can be mounted successfully using dynamic variables, while cube-attack-like cryptanalysis with auxiliary variables fails.

3.3 Conditional Cube Attacks on Round-Reduced KECCAK

In [HWX+17], Huang *et al.* proposed conditional cube testers for keyed KECCAK sponge function, in which the propagation of certain cube variables are controlled in the first few rounds if some conditions are satisfied. If the conditions involve the key information, such cube tester could be used to recover the key. Using conditional cube testers, key recovery attacks were obtained for various instances of KECCAK-MAC and KEYAK in [HWX+17]. Later, the attacks on KECCAK-MAC and KETJE attacks were improved with better conditional cubes found by an MILP model by Li *et al.* in [LBDW17]. Inspired by [LBDW17], Song *et al.* [SGSL17] provided a new MILP model for searching conditional cubes of KECCAK that fully describes the first two rounds, and the application of the new model leads to a series of better attacks against `KMAC` [The16], KEYAK, KETJE and KECCAK-MAC.

3.4 Motivations

As shown in Song *et al.*'s work, MILP widely improves conditional cube attacks on KECCAK based constructions. However, there is no MILP modeling in the literature for cube-attack-like cryptanalysis on KECCAK. Additionally, it is also noted from Song *et al.*'s work, that for KECCAK constructions with small rate (or nonce length), conditional cube attacks become less powerful whereas cube-attack-like cryptanalysis still works as shown by [DLWQ17]. So the major motivation of this work is to investigate the application of MILP in cube-attack-like cryptanalysis and access its efficiency in KECCAK constructions with relatively small rate, like KETJE Jr, KETJE Sr and KECCAK-MAC-512.

4 MILP Model for Cube-Attack-Like Cryptanalysis

Mixed integer linear programming (MILP) is a general mathematical tool for optimization, which takes an objective function and a system of linear inequalities with respect to real numbers as input, and find solutions that optimize the objective function under the constraints of all inequalities. In [MWGP11], Mouha *et al.* firstly showed that searching differential trails can be converted to an MILP problem.

In this section, ideas and techniques are introduced for searching cubes with auxiliary/dynamic variables for KECCAK-p based constructions.

4.1 Basic Idea

In cube-attack-like cryptanalysis of KECCAK-based constructions, cube variables are selected such that they do not multiply with each other in the first round, *i.e.*, the first round is linearized. Due to the fact that the algebraic degree of the round function is 2,

the algebraic degree of the output after n rounds is 2^{n-1} if the first round is linearized. Therefore, a 2^{n-1}-dimensional cube can act as a cube tester for n-round KECCAK and be used to recover the key in cube-attack-like cryptanalysis. The time complexity of such cube attacks not only depends on the dimension (d) of the cube, but also depends on the number (n_i) of key bits which the cube sum depends on, and the number (n_a) of auxiliary/dynamic variables. As introduced in Section 3, the time complexities of cube-attack-like cryptanalysis are

- Preprocessing phase: 2^{d+n_i}
- Online phase: 2^{d+n_a}

Note that, in previous papers [DMP+15, DLWQ17] either auxiliary variables or dynamic variables are used, where auxiliary variables only contain some key bits while dynamic variables contain both cube variables and key bits. In this paper, we utilize both and call them *auxiliary variables* for simplicity since their impacts on the time complexity are the same.

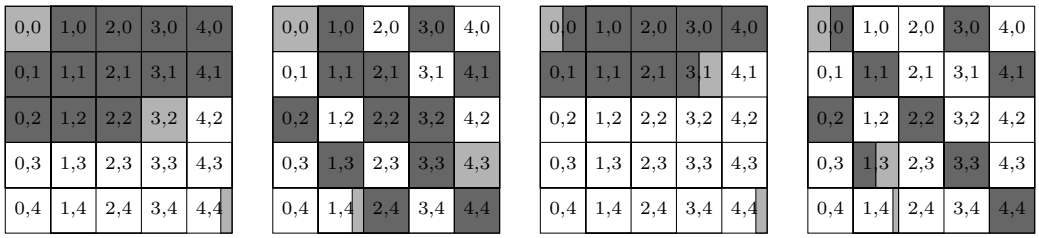
With the basics of cube-attack-like cryptanalysis in mind, the main goals of the MILP modeling are clear:

1. Find 2^{n-1}-dimensional cubes where n is as large as possible;
2. Find balanced attacks where n_i and n_a are close and as small as possible.

The model for searching cubes of KECCAK using auxiliary variables contains two lines: the propagation of cube variables through the linear layer and the propagation of key bits through the linear layer. At the nonlinear layer χ in the first round, these two lines merge and interact. In the following subsections, the model will be introduced accordingly. For the sake of clarity, we take KETJE Jr V1 as an example.

4.2 Propagation of Cube Variables and the Dimension d

Cube variables have to traverse all possible values, so they should be placed where the values are under control of the attacker, e.g., the nonce or message. For KETJE Jr V1, as shown in Figure 4 (a), cube variables can be set only in white lanes under the nonce respected setting.



(a) KETJE Jr V1 (b) KETJE Jr V2 (c) KETJE Sr V1 (d) KETJE Sr V2

Figure 4: Key pack of KETJE Jr and KETJE Sr, where gray lanes are the key, light gray lanes denote padded or encoded bits and white lanes are the nonce.

Suppose $a[x][y][z], b[x][y][z]$, $0 \le x, y < 5, 0 \le z < w$, are the input and output of θ. We introduce $A[x][y][z], B[x][y][z]$, $0 \le x, y < 5, 0 \le z < w$, where $A[x][y][z] = 1$ ($B[x][y][z] = 1$) if $a[x][y][z]$ ($b[x][y][z]$) contains a cube variable; otherwise $A[x][y][z] = 0$ ($B[x][y][z] = 0$). Each column of a may have 0, 1 or multiple cube variables. Namely, each column sum of A may be 0, 1, or more. For a column with multiple cube variables, the sum of these cube variables can be constrained to certain constant (usually 0) so that the cube

variables in this column do not diffuse to other columns. If all column sums are constant, then the state a is in the column-parity-like kernel (*CP-like kernel*). In previous cube-attack-like cryptanalysis of Keccak-based constructions [DMP+15, DLWQ17], or even conditional cube attacks [HWX+17, LBDW17], cube variables are set in CP-like kernel. This is reasonable. If cube variables are set in CP-like kernel, $A[x][y][z] = B[x][y][z]$ and θ acts as the identity. In this way, the propagation of cube variables becomes simple. However, setting cube variables in CP-like kernel limits the dimension of the cube that can be obtained, and this drawback becomes non-negligible when the nonce or message block length is short.

In our model, we do not add any additional constraint to cube variables and take all possible cases into consideration. To do this, we introduce $G[x][z], D[x][z]$, $0 \leq x < 5, 0 \leq z < w$, where $G[x][z] = 1$ if the sum of column (x, z) contains cube variables and 0 otherwise; $D[x][z] = 1$ if the column (x, z) contains cube variables and the sum of these variables is a constant and 0 otherwise. For example, each column in the first four sheet of Ketje Jr V1 has only two bit positions, say (x, y_0, z) and (x, y_1, z) that can be chosen as cube variables, so $G[x][z]$ and $D[x][z]$ depend on $A[x][y_0][z]$ and $A[x][y_1][z]$. The patterns $A[x][y_0][z], A[x][y_1][z], G[x][z], D[x][z]$ follows are listed in Table 3, as well as the inequalities describing these patterns. The inequalities that confine the 0-1 patterns into a finite set can be obtained through the *inequality_generator()* function in SageMath, as suggested in [SHW+14]. After that, an additional algorithm from [ST17], is used to select a minimal number of inequalities from the inequalities returned by *inequality_generator()*.

Table 3: Patterns of cube variables through θ and inequalities.

$A[x][y_0][z]$	$A[x][y_1][z]$	$G[x][z]$	$D[x][z]$	Inequalities
0	0	0	0	$A[x][y_0][z] + A[x][y_1][z] - G[x][z] - 2D[x][z] \geq 0,$ $-A[x][y_1][z] + G[x][z] + D[x][z] \geq 0,$ $-A[x][y_0][z] + G[x][z] + D[x][z] \geq 0.$
0	1	1	0	
1	0	1	0	
1	1	1	0	
1	1	0	1	

For columns with other numbers of bit positions that can be chosen as cube variables, inequalities can be generated in a similar way. When the full column is available for cube variables, inequalities in Table 9 in Appendix can be used.

According to the definition of θ, $B[x][y][z] = 1$ if any of $A[x][y][z]$, $G[x-1][z]$ or $G[x+1][z-1]$ is 1. This can be described by the following inequalities.

$$\begin{aligned} B[x][y][z] - A[x][y][z] &\geq 0, \\ B[x][y][z] - G[x+1][z-1] &\geq 0, \\ B[x][y][z] - G[x-1][z] &\geq 0, \\ A[x][y][z] + G[x-1][z] + G[x+1][z-1] - B[x][y][z] &\geq 0. \end{aligned} \tag{1}$$

Since ρ and π just change the bit positions of the state, we let $c = \pi \circ \rho(b)$ and $C = \pi \circ \rho(B)$. Now the propagation of cube variables through the linear layer is modeled. To linearize the first round, cube variables in c should not be adjacent, which can be constrained by

$$C[x][y][z] + C[x+1][y][z] \leq 1.$$

The dimension of the cube is determined by $A[x][y][z]$ and $D[x][y]$, and

$$d = \sum A[x][y][z] - \sum D[x][z].$$

With these inequalities, the set of solutions is exactly the set of all possible cubes that linearize the first round.

4.3 Propagation of Key Bits and n_a

This subsection presents the model for the propagation of key bits, and the number of auxiliary variables n_a is also calculated alongside.

First, let $W[x][y][z] = 1$ ($Y[x][y][z] = 1$) if $a[x][y][z]$ ($b[x][y][z]$) contains a key bit; otherwise $W[x][y][z] = 0$ ($Y[x][y][z] = 0$). Additionally, $Z = \pi \circ \rho(Y)$. For KETJE Jr, the key pack is loaded into the gray and light gray lanes as depicted in Figure 4, so $W[x][y][z] = 1$ for bits in gray and $W[x][y][z] = 0$ for bits in light gray. Among the white lanes, $W[x][y][z] = 1$ indicates an auxiliary variable at (x, y, z).

Second, let $X[x][z] = 1$ if the sum of column (x, z) contains key bits, otherwise $X[x][z] = 0$, meaning that no key bit in this column could diffuse to column $(x-1, z+1)$ and $(x+1, z)$ through θ. From Figure 4, it is known that each column of KETJE Jr contains key bits. Hence, $X[x][z] = 0$ if and only if there is an auxiliary variable in that column. So $X[x][z]$ depends on the bits that can be chosen as auxiliary variables and the sum of them should be 1. Suppose there are two bit positions (x, y_0, z) and (x, y_1, z) in column (x, z) that can be chosen as auxiliary variables, then

$$X[x][z] + W[x][y_0][z] + W[x][y_1][z] = 1.$$

The relation of $Y[x][y][z]$ and $W[x][y][z]$ can be described with $X[x][z]$. Each $Y[x][y][z]$ is calculated from $W[x][y][z]$, $X[x-1][z]$ and $X[x+1][z-1]$, and any one is 1 will lead to $Y[x][y][z] = 1$; otherwise $Y[x][y][z] = 0$. The inequalities describing this relation are the same as (1).

At last, the number of auxiliary variables is the sum of $W[x][y][z]$ in the white lanes. For KETJE Jr V1,

$$n_a = \sum_{x,z,3\le y<5} W[x][y][z] + \sum_{z} W[4][2][z].$$

4.4 Interaction of Key Bits and Cube Variables, and n_i

Recall that $Z = \pi \circ \rho(Y)$ and $Z[x][y][z] = 1$ indicates that the input of χ in the first round at (x, y, z) contains key information. If its neighbouring bits contain cube variables, *i.e.*, $C[x-1][y][z] = 1$ or $C[x+1][y][z] = 1$, then the key bit propagated to position (x, y, z) affects the cube sum and thus it is an involved key bit. In order to calculate the number of involved key bits n_i, $U[x][y][z]$ is introduced, where $U[x][y][z] = 1$ if the key bit at (x, y, z) is an involved key bit and 0 otherwise. All possible patterns of $(C[x-1][y][z], C[x][y][z], C[x+1][y][z], Z[x][y][z], U[x][y][z])$ are listed in Table 4, which can be described with 5 inequalities, three of which are new and shown in the last line of Table 4.

Table 4: Patterns of key bits and cube variables. Symbol '*' denotes arbitrary value.

$C[x-1][y][z]$	$C[x][y][z]$	$C[x+1][y][z]$	$Z[x][y][z]$	$U[x][y][z]$
0	*	0	0	0
0	*	0	1	*
1	0	*	0	0
1	0	*	1	1
0	0	1	0	0
0	0	1	1	1
			$Z[x][y][z] - U[x][y][z] \ge 0$	
			$C[x-1][y][z] + Z[x][y][z] - U[x][y][z] \le 1$	
			$C[x+1][y][z] + Z[x][y][z] - U[x][y][z] \le 1$	

To calculate the number of involved key bits n_i, we sum $U[x][y][z]$ together. Namely, $n_i = \sum U[x][y][z]$. However, the same key bit may appear in multiple positions of U.

Recall that θ adds the XORs of bits in column $(x-1,z)$ and $(x+1,z-1)$ to each bit of column (x,z). If at least two bits in column (x,z) do not contain key bits, then after θ these bits contain either the same key bits or none. So the same key bit appearing in multiple positions may be counted more than once with $n_i = \sum U[x][y][z]$, making n_i inaccurate.

To partially solve this, we introduce the key pattern $V[x][y][z]$,$0 \leq x, y < 5, 0 \leq z < w$, after θ, where $V[x][y][z]$s in each column are equal if these bits do not contain key bit before θ. For KETJE Jr V1, $V[x][0][z] = V[x][3][z] = V[x][4][z]$ for columns in the first sheet ($x = 0$), $V[x][3][z] = V[x][4][z]$ for the second and third sheet ($x = 1, 2$), and $V[x][2][z] = V[x][3][z] = V[x][4][z]$ for the fourth and fifth sheet ($x = 4, 5$). Then, we let $U = \pi \circ \rho(V)$, and n_i is set to be the sum of all distinct variables in U.

The problem remained unsolved is the impact of auxiliary variables on the key pattern $V[x][y][z]$ and the dependence of involved key bits. If $W[0][3][z] = 1$ for KETJE Jr V1, *i.e.*, $a[0][3][z]$ is an auxiliary variable, then $V[0][3][z]$ should not be equal to $V[0][0][z]$ and $V[0][4][z]$ (but $V[0][0][z] = V[0][4][z]$). In addition, the n_i involved key bits may be not fully independent and calculating the number of independent involved key bits is beyond the reach of MILP. Therefore, n_i may be still inaccurate. We leave this problem to be fixed with a postprocessing procedure.

Now, the whole model for searching cubes using auxiliary variables can be built using techniques introduced in this section. We additionally set $d = 2^{n-1}$ and the objective function to be 'Minimize n_i, n_a'. An MILP solver like Gurobi [Gur18] can then be invoked to find optimal solutions.

4.5 Postprocessing Procedure

Algorithm 1: Postprocessing procedure for recalculating n_i.

Input: A, W of the solution, the cube dimension d and the key length $|K|$
Output: n_i
$rel = \emptyset$;
BR = BooleanPolynomialRing($d + |K|, [k_0, ..., k_{|K|-1}, v_0, ..., v_{d-1}]$);
a = zeroState();
loadKey(a, $[k_0, ..., k_{|K|-1}]$);
loadCubeVar(a,A, $[v_0, ..., v_{d-1}]$);
loadAux(a, W);
$\pi \circ \rho \circ \theta(a)$;
for *All* $a[x][y][z]$ **do**
  if $a[x][y][z]$ *contains* $v_0, ..., v_{d-1}$ **then**
    if $a[x-1][y][z]$ *contains* $k_0, ..., k_{|K|-1}$ **then**
      $rel \leftarrow a[x-1][y][z]$;
    end
    if $a[x+1][y][z]$ *contains* $k_0, ..., k_{|K|-1}$ **then**
      $rel \leftarrow a[x+1][y][z]$;
    end
  end
end
return rank(rel);

From the solution returned by MILP solvers, we recalculated the number of involved key bits using symbolic computations. First, key bits are loaded to the state, and cube variables and auxiliary variables are set according to the solution. Then pass the state

through the linear layer, and collect key bits (linear expressions of key bits) that are adjacent to the cube variables. The denser the key bits are in the initial state, the more complex the relation of involved key bits will be, but only the number of independent involved key bits matters and is the actual n_i. The detailed postprocessing procedure is described in Algorithm 1. Since the number of involved key bits optimized by our model may not be equal to the actual n_i, our model does not guarantee optimal solutions with respect to attack complexities, even though the dimension of cubes can be optimized. The experiments show that in most cases the actual n_i lies in $[n_i^\star - 2, n_i^\star + 2]$, where $n_i^\star$ is the claimed number of involved key bits by the model. This means that our model still finds *almost optimal* solutions.

5 Application to Ketje Jr, Ketje Sr, Xoodoo and Keccak-MAC-512

In this section, we apply the model described in Section 4 to Ketje Jr, Ketje Sr, Xoodoo-based AE in Ketje style and Keccak-MAC-512, all of which have relatively small nonce or message block length. First, improved 5-round attacks of Ketje Jr are obtained, where the time complexity of the attack is reduced significantly. Then, we consider Ketje Jr with reduced key size. Namely, the key size is less than 96 bits, and the security goal of confidentiality becomes $|K| = \min(96, |K|)$ according to the security claims of Ketje [BDP+16a]. As a result, one more round of Ketje Jr V1 (V2) can be attacked if the key size is reduced to 72 (80) bits. Also, we give an improved 7-round attack on Ketje Sr V2. Finally, a 6-round attack on the Xoodoo-based AE and a 7-round attack of Keccak-MAC-512 are also achieved.

5.1 5-Round Attack against Ketje Jr V1 with Recommended Key Size

The attack on 5-round Ketje V1 sequentially utilizes three 16-dimensional cubes as shown in Table 7 and 8. Each cube helps to recover part of the key and these three cubes work together to make the whole time complexity low.

The first cube has $n_i = 18$ involved key bits (linear combination of key bits) and $n_a = 17$ auxiliary variables which are listed in Table 5. The two phases of the attack proceed as follows.

Preprocessing phase:

1. Set the 18 bits in light gray according to the encoding rule, as illustrated in Figure 4 (a). Set all key bits to zero except $k_2, k_{75}, k_{15}, k_{63}, k_{12}, k_{60}, k_{93}, k_{61}, k_5, k_{78}, k_{20}, k_{23}, k_3, k_{56}, k_{57}, k_{58}, k_{59}, k_{62}$[1]. Set all other state bits to an arbitrary constant except the 16 cube variables, 9 out of the 17 auxiliary variables $a[1][3][1]$, $a[1][4][4]$, $a[4][2][0]$, $a[4][2][1]$, $a[4][3][2]$, $a[4][2][5]$, $a[4][2][7]$, $a[3][4][0]$, $a[3][4][3]$, and $a[0][3][0]$, $a[0][3][1]$.

2. For the 2^{18} values of $k_2, k_{75}, k_{15}, k_{63}, k_{12}, k_{60}, k_{93}, k_{61}, k_5, k_{78}, k_{20}, k_{23}, k_3, k_{56}, k_{57}, k_{58}, k_{59}, k_{62}$:
   (a) For each of the 4 values of $a[0][3][0], a[0][3][1]$, calculate the cube sum of the 16-bit output after 5 rounds.
   (b) Store the four 16-bit cube sums in a sorted list L, next to the value of the corresponding $k_2, k_2 + k_{75}, k_{15}, k_{15} + k_{63}, k_{12}, k_{12} + k_{60}, k_{93}, k_{93} + k_{61}, k_5, k_5 + k_{78}, k_4 + k_{20} + k_{60}, k_{23} + k_{63}, k_3, k_{56}, k_{57}, k_{58}, k_{59}, k_{62}$.

[1] This is not the only way to choose free key bits in the preprocessing phase.

Table 5: Auxiliary variables and involved key bits of the first cube for KETJE Jr V1 where the gray key bits are set to be zero in the preprocessing phase.

Auxiliary variables	Involved key bits
$a[1][3][0] = k_0 + k_{40} + k_{80}$	$k_2 + k_{42} + k_{27} + k_{67} + k_{82}$
$a[1][3][1] = k_1 + k_{41} + k_{81} + v_0$	$k_2 + k_{42} + k_{27} + k_{67} + k_{82} + k_{75}$
$a[1][4][4] = k_4 + k_{44} + k_{84} + v_1$	$k_{15} + k_{30} + k_{55} + k_{70} + k_{95}$
$a[1][3][6] = k_6 + k_{46} + k_{86}$	$k_{15} + k_{30} + k_{55} + k_{70} + k_{95} + k_{63}$
$a[1][4][7] = k_7 + k_{47} + k_{87}$	$k_{12} + k_{27} + k_{52} + k_{67} + k_{92}$
$a[2][3][0] = k_8 + k_{48} + k_{88}$	$k_{12} + k_{27} + k_{52} + k_{67} + k_{92} + k_{60}$
$a[2][3][1] = k_9 + k_{49} + k_{89}$	$k_{13} + k_{28} + k_{53} + k_{68} + k_{93}$
$a[2][3][2] = k_{10} + k_{50} + k_{90}$	$k_{13} + k_{28} + k_{53} + k_{68} + k_{93} + k_{61}$
$a[2][4][3] = k_{11} + k_{51} + k_{91}$	$k_5 + k_{30} + k_{45} + k_{70} + k_{85}$
$a[2][3][6] = k_{14} + k_{54} + k_{94}$	$k_5 + k_{30} + k_{45} + k_{70} + k_{85} + k_{78}$
$a[3][4][0] = k_{16} + k_{56}$	$k_5 + k_{13} + k_{45} + k_{85} + k_{20} + k_{60}$
$a[3][4][3] = k_{19} + k_{59}$	$k_8 + k_{23} + k_{63}$
$a[4][2][0] = k_{24} + k_{64} + v_2$	$k_3 + k_{28} + k_{43} + k_{68} + k_{83}$
$a[4][2][1] = k_{25} + k_{65} + v_4$	$k_{56}, k_{57}, k_{58}, k_{59}, k_{62}$
$a[4][3][2] = k_{26} + k_{66} + v_7$	
$a[4][2][5] = k_{29} + k_{69} + v_{12}$	
$a[4][2][7] = k_{31} + k_{71} + v_{15}$	

Online phase:

1. For all possible values of the 17 linear expressions of key bits in auxiliary variables:
   (a) Set the auxiliary variables accordingly. For each of the 4 values of $a[0][3][0]$, $a[0][3][1]$, request the 16-bit outputs for the cube and calculate the cube sums (setting the same constant values in the state as in the preprocessing).
   (b) For each match in L, retrieve the 18-bit value for the involved key bits and record it and the current value of the 17 key bits in auxiliary variables as a candidate.

In the online phase, only one candidate for the 35-bit partial key will survive, since $2^{18+17} \cdot 2^{-64} < 1$. The time complexity of the preprocessing phase is $2^{18+16+2} = 2^{36}$, and the memory complexity is 2^{18}. The time complexity of the online phase is $2^{17+16+2} = 2^{35}$. In the end, $n_k = 35$ bits information of the key are obtained. However, $96 - 35 = 51$ bits of the key are still unknown. Next, we use the second and the third cube to recover more key bits. Since the two phases of the attack using other cubes work similarly to those of the first cube for KETJE Jr V1, details of the attacks are omitted afterward, and only complexities are given.

The second cube has $n_i = 22$ involved key bits and $n_a = 21$ auxiliary variables, see Table 7 in the appendix. With 35 bits of the key known from the first cube, the number of unknown involved key bits is $n_i' = 14$, and the number of unknown key bits in the auxiliary variables is $n_a' = 9$. So the complexities are as follows.

- Preprocessing phase: the time complexity is $2^{14+16+2} = 2^{32}$ and the memory complexity is 2^{14};
- Online phase: the time complexity is $2^{9+16+2} = 2^{27}$.

The accumulated number of key bits recovered from the first two cubes is $n_k = 57$.

The third cube has $n_i = 27$ involved key bits and $n_a = 26$ auxiliary variables, see Table 8. With 57 bits of the key known, the number of unknown involved key bits becomes $n_i' = 16$, and the number of unknown key bits in the auxiliary variables is $n_a' = 4$. So the complexities are as follows.

- Preprocessing phase: the time complexity is $2^{16+16+2} = 2^{34}$ and the memory complexity is 2^{16};
- Online phase: the time complexity is $2^{4+16+2} = 2^{22}$.

The number of key bits recovered from the three cubes is $n_k = 74$.

In all, all key bits can be recovered with time complexity $2^{36} + 2^{35} + 2^{32} + 2^{27} + 2^{34} + 2^{22} + 2^{96-74} = 2^{36.86}$ and the memory complexity 2^{18}.

5.2 5-Round Attack against KETJE Jr V2 with Recommended Key Size

The attack on 5-round KETJE Jr V2 also uses three 16-dimensional cubes, shown in Table 10,11. The attack on KETJE Jr V2 proceeds the same as the attack on V1. Here, we just calculate the complexities.

The first cube has $n_i = 14$ involved key bits and $n_a = 15$ auxiliary variables (see Table 10). The complexities using the first cube are as follows.

- Preprocessing phase: the time complexity is $2^{14+16+2} = 2^{32}$ and the memory complexity is 2^{14};
- Online phase: the time complexity is $2^{15+16+2} = 2^{33}$.

The number of key bits recovered from the first cube is $n_k = 29$.

The second cube has $n_i = 18$ involved key bits and $n_a = 15$ auxiliary variables (see Table 10). With 29 bits of the key known from the first cube, the number of unknown involved key bits is $n_i' = 13$, and the number of unknown key bits in the auxiliary variables is $n_a' = 9$. So the complexities are as follows.

- Preprocessing phase: the time complexity is $2^{13+16+2} = 2^{31}$ and the memory complexity is 2^{13};
- Online phase: the time complexity is $2^{9+16+2} = 2^{27}$.

The accumulated number of key bits recovered from the first two cubes is $n_k = 47$.

The third cube has $n_i = 32$ involved key bits and $n_a = 26$ auxiliary variables (see Table 11). With 47 bits of the key known, the number of unknown involved key bits becomes $n_i' = 15$, and the number of unknown key bits in the auxiliary variables is $n_a' = 1$. So the complexities are as follows.

- Preprocessing phase: the time complexity is $2^{15+16+2} = 2^{33}$ and the memory complexity is 2^{15};
- Online phase: the time complexity is $2^{1+16+2} = 2^{19}$.

The number of key bits recovered from the three cubes is $n_k = 63$.

In all, the full key can be recovered with time complexity $2^{32} + 2^{33} + 2^{31} + 2^{27} + 2^{33} + 2^{19} + 2^{96-63} = 2^{34.91}$ and the memory complexity 2^{15}.

5.3 6-Round Attack against KETJE Jr V1 with Reduced Key Size

To extend our attack on KETJE Jr V1 by one round, we need 32-dimensional cubes. However, cubes that linearize the first round have dimension of 25 at most, as demonstrated by the experiment where we only focus on cube variables and set no constraint on the number of auxiliary variables or involved key bits. Recall that our model covers all possible cubes that linearize the first round. Therefore, 32-dimensional cubes that linearize the first round of KETJE Jr V1 do not exist.

When the key size of KETJE Jr V1 is reduced to 72 bits, i.e., the nonce size increases, 32-dimensional cubes can be found. Consequently, one more round can be attacked. The 32-dimensional cube used in our attack is presented in Table 12, and has 29 auxiliary variables and 34 involved key bits. The time complexities of the 6-round attack on KETJE Jr V1 with a 72-bit key are calculated as follows.

- Preprocessing phase: the time complexity is $2^{34+32+2} = 2^{68}$ and the memory complexity is 2^{34};
- Online phase: the time complexity is $2^{29+32+2} = 2^{63}$.

With this cube, 57 bits information of the key can be recovered. The remaining $72 - 57 = 15$ key bits can be recovered by brute force. In total, the time complexity is $2^{68} + 2^{63} + 2^{15} = 2^{68.04}$, and the memory complexity is 2^{34}.

5.4 6-Round Attack against KETJE Jr V2 with Reduced Key Size

The experiment shows that 32-dimensional cubes of KETJE Jr V2 that linearize the first round do not exist and the maximal dimension of such cubes is 24. When the key size of KETJE Jr V2 is reduced to 80 bits, 32-dimensional cubes can be found using our model, and the number of rounds attacked can be increased to 6. The 32-dimensional cube used in our attack is presented in Table 13 which has 22 auxiliary variables and 25 involved key bits. The time complexities of the 6-round attack on KETJE Jr V2 with an 80-bit key are calculated as follows.

- Preprocessing phase: the time complexity is $2^{25+32+2} = 2^{59}$ and the memory complexity is 2^{25};
- Online phase: the time complexity is $2^{22+32+2} = 2^{56}$.

With this cube, 40 bits information of the key can be recovered. The remaining $80 - 40 = 40$ key bits can be recovered by brute force. In total, the time complexity is $2^{59}+2^{56}+2^{40} = 2^{59.17}$, and the memory complexity is 2^{25}.

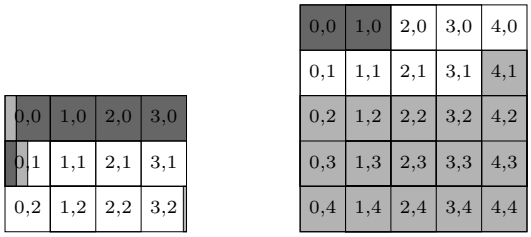
5.5 7-Round Attack against KETJE Sr

For KETJE Sr V1, the best 64-dimensional cube found by our model has 48 auxiliary variables and 48 involved key bits, leading to an attack on 7 rounds of KETJE Sr V1 with time complexity 2^{114}, which is slightly better than the attack in [DLWQ17], but worse than the conditional attack in [SGSL17].

For KETJE Sr V2, the 64-dimensional cube used in our attack is presented in Table 14 which has 33 auxiliary variables and 33 involved key bits. The time complexities of the 7-round attack on KETJE Sr V2 are calculated as follows.

- Preprocessing phase: the time complexity is $2^{33+64+1} = 2^{98}$ and the memory complexity is 2^{33};
- Online: the time complexity is $2^{33+64+1} = 2^{98}$.

With this cube, 60 bits information of the key can be recovered. The remaining $128 - 60 = 68$ key bits can be recovered by brute force. In total, the time complexity is $2^{98} + 2^{98} + 2^{68} \approx 2^{99}$, and the memory complexity is 2^{33}.



(a) Xoodoo-based AE (b) Keccak-MAC-512

Figure 5: Key pack of Xoodoo in Ketje style and Keccak-MAC-512, where gray lanes are the key, light gray lanes denote constants and white lanes are the nonce or message.

5.6 6-Round Attack against Xoodoo-based AE

Assume that the key of the Xoodoo-based AE has 128 bits and follows the Ketje's packing, as shown in Figure 5 (a). Since the operations θ and χ of Xoodoo are very similar to those of Keccak-p and ρ_{west} just reorders the state bits, the model described in Section 4 can be adapted to Xoodoo easily.

When we only focus on cube variables, the experiment shows that 64-dimensional cubes linearizing the first round do not exist and the maximal dimension of such cubes is 62. Therefore, we mount an attack on 6-round Xoodoo-based AE using a 32-dimensional cube.

The 32-dimensional cube used in our attack is presented in Table 15 which has 55 auxiliary variables and 55 involved key bits. The time complexities of the 6-round attack on the Xoodoo-based AE are calculated as follows under the assumption that the rate is 32.

- Preprocessing phase: the time complexity is $2^{55+32+1} = 2^{88}$ and the memory complexity is 2^{55};
- Online: the time complexity is $2^{55+32+1} = 2^{88}$.

With this cube, 106 bits information of the key can be recovered. The remaining $128 - 106 = 22$ key bits can be recovered by brute force. In total, the time complexity is $2^{88} + 2^{88} + 2^{22} \approx 2^{89}$, and the memory complexity is 2^{55}.

Note that such cubes with dimension 64 exist for Ketje Sr but it is not the case for Xoodoo-based AE. One reason is that Ketje Sr has a slightly larger state which provides 16 more degrees of freedom. Another important reason lies in the differences of the underlying permutation as follows

- Columns in Xoodoo are shorter than those in Keccak-p. Note long columns (specifically, columns of more free bits) are advantageous to save degrees of freedom.
- The non-linear operation (S-box) is applied to every 3-bit column in Xoodoo but to every 5-bit row in Keccak-p. More specifically, at most one bit in each column of Xoodoo contains cube variables while at most two bits in each row of Keccak-p contain cube variables.

Interestingly, if the non-linear operation is applied to every 4-bit row in Xoodoo (even though such nonlinear operations on 4-bit rows are not invertible), the dimension of cubes that linearize the first round can reach 99, allowing 64-dimensional cubes that cover one more round. Therefore, short columns and narrow S-boxes which heavily limit the dimension of the cube are helpful for Xoodoo-based AE in resisting cube-attack-like analysis.

5.7 7-Round Attack against Keccak-MAC-512

The key pack of Keccak-MAC-512 is shown in Figure 5 (b). One of 64-dimensional cubes for Keccak-MAC-512 is shown in Table 16, which has 46 auxiliary variables and 46 involved key bits. The time complexities of the 7-round attack on Keccak-MAC-512 are calculated as follows.

- Preprocessing phase: the time complexity is $2^{46+64} = 2^{110}$ and the memory complexity is 2^{46};
- Online: the time complexity is $2^{46+64} = 2^{110}$.

With this cube, 92 bits information of the key can be recovered. The remaining $128 - 92 = 36$ key bits can be recovered by brute force. In total, the time complexity is $2^{110} + 2^{110} + 2^{36} \approx 2^{111}$, and the memory complexity is 2^{46}.

5.8 Experiment and Verification

In this paper, cubes are searched by feeding the generated inequalities to Gurobi Optimizer [Gur18]. The running time for searching cubes varies from seconds to hours. Specifically, it takes seconds on a PC to search cubes for Ketje Jr and Ketje Sr, minutes for Xoodoo and hours for Keccak-MAC-512.

To verify the correctness of the attacks in this section, we implemented the attack on 5-round Ketje Jr V1 using the first cube whose details are displayed in Table 7. The 18 involved key bits and 17 auxiliary variables are also presented in Table 5 for better understanding. The experiments show that the right value of the involved key bits and the key bits in auxiliary variables can be recovered successfully[2].

6 Discussion and Comparison

Our results of Section 5 are summarized in Table 1, along with a comparison with related works. Below, the comparison will be explained in more detail.

Cube-attack-like cryptanalysis with and without MILP. In [DLWQ17], Dong *et al.* studied cube-attack-like cryptanalysis of Ketje, where cubes were constructed manually. Compared with Dong *et al.*'s work, our automated method using MILP helps to find better cubes and thus obtains better attacks. Moreover, using our model, it becomes easier to carry out cube-attack-like cryptanalysis of keyed Keccak constructions or prove that cubes of certain dimensions do not exist.

Cube-attack-like cryptanalysis and conditional cube attacks. In general, the most important factor in both types of attack is the number of degrees of freedom, *i.e.*, message block size or nonce size. Table 6 summarizes the numbers of degrees of freedom for keyed Keccak construction discussed in this paper. Recall that the first round of the Keccak permutation is linearized in both attacks, but in conditional cube attacks the propagation of some cube variables is controlled in the second round by consuming additional degrees of freedom. If there are sufficient degrees of freedom in a keyed Keccak construction, only a few conditions are required to construct conditional cubes, resulting in a lower time complexity than cube-attack-like cryptanalysis. But on keyed Keccak constructions with small degrees of freedom, *i.e.*, small message block size or nonce size, conditional cube attacks do not perform as well as cube-attack-like cryptanalysis. For example, 16-dimensional conditional cubes do not exist for Ketje Jr [SGSL17], and thus

[2]The source code is available via http://team.crypto.sg/auxCube.zip.

Table 6: Keccak-p-based constructions and their available degrees of freedom and dimensions of cubes used in attacks. 'Type' refers to the attack which is more advantageous.

		Ketje				
	Lake Keyak	Major	Minor	Sr	Jr	Keccak-MAC-512
$\boldsymbol{b}$	1600	1600	800	400	200	1600
DF	1200	1454	654	254	86	447
d	64	64	64	64	16	64
Type	CC	CC	CC	CC/CAL	CAL	CAL

[CC] Conditional cube attacks
[CAL] Cube-attack-like cryptanalysis

5-round attacks are impossible using conditional cube attacks, but both [DLWQ17] and our work show that it is not the case for cube-attack-like cryptanalysis.

Apart from the number of degrees of freedom, another important factor in both attacks is the layout of the state, especially the layout of free bits. This can be seen from the analysis of Ketje Sr and Keccak-MAC-512. Keccak-MAC-512 has 447 degrees of freedom which is much larger than that of Ketje Sr, but the dimension of conditional cubes of Keccak-MAC-512 hardly reaches 64 while 64-dimensional conditional cubes of Ketje Sr can be found easily [SGSL17]. One reason is that each column in the initial state of Keccak-MAC-512 has only one or two free bits while in that of Ketje Sr almost every column has at least three free bits. As discussed in Section 5.6, columns of more free bits are beneficial to save degrees of freedom.

Very recently, Bi *et al.* [BDL+18] also provided an MILP model for cube-attack-like cryptanalysis of keyed Keccak and applied it to Lake Keyak, Ketje Major, Ketje Minor and Keccak-MAC which were also analyzed in [SGSL17]. The comparison between the results from [SGSL17] and [BDL+18] shows that for Keyak, Ketje Major and Ketje Minor which have relatively large degrees of freedom, conditional cube attacks outperform cube-attack-like cryptanalysis. Further, our work shows that for Ketje Sr V2, Ketje Jr, and Keccak-MAC-512 with relatively small degrees of freedom, cube-attack-like cryptanalysis is more efficient. To make sure we can stop at Ketje Sr safely, we add an experiment on Ketje Minor and obtain a 7-round attack with complexity 2^{92} by finding the cube in Table 17. This attack is better than the attack of Ketje Minor in [BDL+18] which has a time complexity of 2^{113}, but worse than the conditional cube attack in [SGSL17] whose time complexity is $2^{73.03}$. Therefore, we do not apply our model to other targets with degrees of freedom larger than that of Ketje Sr, such as Ketje Major, and Keyak.

7 Conclusion

Cube-attack-like cryptanalysis using auxiliary/dynamic variables are of special interest since they are efficient for Keccak-p based constructions with a small message block size or nonce size. In this paper, we proposed a new MILP model for cube-attack-like cryptanalysis against Keccak-p based constructions, which particularly takes both auxiliary and dynamic variables into consideration and aims to find almost optimal attacks by balancing the two phases of the cube-attack-like cryptanalysis. Under the new model, the best 5-round attacks on Ketje Jr and 7-round attacks on Ketje Sr V2 were improved and 6-round attacks on Ketje Jr were achieved when the key size is reduced. The application of our model to the Xoodoo-based AE in Ketje style brought out a 6-round attack and showed that the differences between the Keccak permutation and Xoodoo do affect the resistance against cube-attack-like cryptanalysis. Finally, a 7-round attack on Keccak-MAC-512 was also proposed.

Acknowledgements

The first author is partially supported by the Fundamental Theory and Cutting Edge Technology Research Program of Institute of Information Engineering, CAS (Grant No. Y7Z0341103), Youth Innovation Promotion Association CAS, the National Natural Science Foundation of China (Grants No. 61802399, 61802400, 61732021, 61772519 and 61472415) and Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20180102).

References

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube Testers and Key Recovery Attacks On Reduced-Round MD6 and Trivium. In Helena Handschuh, Stefan Lucks, Bart Preneel, and Phillip Rogaway, editors, *Symmetric Cryptography 2009*, volume 09031 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.

[BDH+17a] Guido Bertoni, Joan Daemen, Seth Hoffert, Johan De Meulder, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Innovations in permutation-based crypto. 21st Workshop on Elliptic Curve Cryptography, 2017. https://ecc2017.cs.ru.nl/slides/ecc2017-daemen.pdf.

[BDH+17b] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

[BDL+18] Wenquan Bi, Xiaoyang Dong, Zheng Li, Rui Zong, and Xiaoyun Wang. Milp-aided cube-attack-like cryptanalysis on keccak keyed modes. *Designs, Codes and Cryptography*, Aug 2018.

[BDP+16a] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: Ketje v2. Candidate of CAESAR Competition, September 2016.

[BDP+16b] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: Keyak v2. Candidate of CAESAR Competition, September 2016.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, 2011.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak Reference. http://keccak.noekeon.org, January 2011. Version 3.0.

[BDPVA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic Sponge functions. *Submission to NIST (Round 3)*, 2011.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 299–320. Springer, 2017.

[CFG+18] Colin Chaigneau, Thomas Fuhr, Henri Gilbert, Jian Guo, Jérémy Jean, Jean-René Reinhard, and Ling Song. Key-recovery attacks on full kravatte. *IACR Trans. Symmetric Cryptol.*, 2018(1):5–28, 2018.

[DDS12] Itai Dinur, Orr Dunkelman, and Adi Shamir. New Attacks on Keccak-224 and Keccak-256. In Anne Canteaut, editor, *FSE 2012*, volume 7549 of *LNCS*, pages 442–461. Springer, 2012.

[DDS13] Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision Attacks on Up to 5 Rounds of SHA-3 Using Generalized Internal Differentials. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 219–240. Springer, 2013.

[DHAK18] Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. Xoodoo cookbook. Cryptology ePrint Archive: Report 2018/767, 2018. https://eprint.iacr.org/2018/767.

[DLWQ17] Xiaoyang Dong, Zheng Li, Xiaoyun Wang, and Ling Qin. Cube-like Attack on Round-Reduced Initialization of Ketje Sr. *IACR Trans. Symmetric Cryptol.*, 2017(1):259–280, 2017.

[DMP+15] Itai Dinur, Pawel Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michal Straus. Cube Attacks and Cube-Attack-Like Cryptanalysis on the Round-Reduced Keccak Sponge Function. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 733–761. Springer, 2015.

[DS09] Itai Dinur and Adi Shamir. Cube Attacks on Tweakable Black Box Polynomials. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 278–299. Springer, 2009.

[DS11] Itai Dinur and Adi Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 167–187. Springer, 2011.

[FNR18] Thomas Fuhr, María Naya-Plasencia, and Yann Rotella. State-recovery attacks on modified ketje jr. *IACR Trans. Symmetric Cryptol.*, 2018(1):29–56, 2018.

[GLS16] Jian Guo, Meicheng Liu, and Ling Song. Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 249–274, 2016.

[Gur18] Gurobi. Gurobi. http://www.gurobi.com/, 2018.

[HWX+17] Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao. Conditional Cube Attack on Reduced-Round Keccak Sponge Function. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 259–288, 2017.

[LBDW17] Zheng Li, Wenquan Bi, Xiaoyang Dong, and Xiaoyun Wang. Improved Conditional Cube Attacks on Keccak Keyed Modes with MILP Method. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 99–127. Springer, 2017.

[MS13] Pawel Morawiecki and Marian Srebrny. A SAT-based preimage analysis of reduced Keccak hash functions. *Inf. Process. Lett.*, 113(10-11):392–397, 2013.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76. Springer, 2011.

[NRM11] María Naya-Plasencia, Andrea Röck, and Willi Meier. Practical Analysis of Reduced-Round Keccak. In Daniel J. Bernstein and Sanjit Chatterjee, editors, *INDOCRYPT 2011*, volume 7107 of *LNCS*, pages 236–254. Springer, 2011.

[QSLG17] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New Collision Attacks on Round-Reduced Keccak. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017 Part III*, volume 10212 of *LNCS*, pages 216–243, 2017.

[SGSL17] Ling Song, Jian Guo, Danping Shi, and San Ling. New MILP Modeling: Improved Conditional Cube Attacks on Keccak-based Constructions. to appear in ASIACRYPT 2018, Cryptology ePrint Archive, Report 2017/1030, 2017. https://eprint.iacr.org/2017/1030.

[SHW+14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 158–178. Springer, 2014.

[SLG17] Ling Song, Guohong Liao, and Jian Guo. Non-full Sbox Linearization: Applications to Collision Attacks on Round-Reduced Keccak. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 428–451. Springer, 2017.

[ST17] Yu Sasaki and Yosuke Todo. New Algorithm for Modeling S-box in MILP Based Differential and Division Trail Search. In Pooya Farshim and Emil Simion, editors, *SecITC 2017*, volume 10543 of *LNCS*, pages 150–165. Springer, 2017.

[The15] The U.S. National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions . Federal Information Processing Standard, FIPS 202, 5th August 2015.

[The16] The U.S. National Institute of Standards and Technology. SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash. NIST Special Publication 800-185, 21st December 2016.

A Appendix

Table 7: The first two 16-dimensional cubes for KETJE Jr V1, where the corresponding (n_a, n_i) are $(17, 18)$ and $(21, 22)$, and (n'_a, n'_i) are $(17, 18)$ and $(9, 14)$, respectively. The accumulated numbers of key bits recovered are 35 and 57.

Cube variables
$a[1][3][1] = a[1][4][1] = v_0, a[1][3][4] = a[1][4][4] = v_1 a[4][2][0] = v_2, a[4][3][0] = v_3,$ $a[4][4][0] = v_2 + v_3, a[4][2][1] = v_4, a[4][3][1] = v_5, a[4][4][1] = v_4 + v_5, a[4][2][2] = v_6,$ $a[4][3][2] = v_7, a[4][4][2] = v_6 + v_7, a[4][2][3] = v_8, a[4][3][3] = v_9, a[4][4][3] = v_8 + v_9,$ $a[4][2][4] = v_{10}, a[4][3][4] = v_{11}, a[4][4][4] = v_{10} + v_{11}, a[4][2][5] = v_{12}, a[4][3][5] = v_{13},$ $a[4][4][5] = v_{12} + v_{13}, a[4][2][6] = a[4][3][6] = v_{14}, a[4][2][7] = a[4][3][7] = v_{15}.$
Auxiliary variables
$a[1][3][0] = k_0 + k_{40} + k_{80}, a[1][3][1] = k_1 + k_{41} + k_{81} + v_1, a[1][4][7] = k_7 + k_{47} + k_{87},$ $a[2][3][1] = k_9 + k_{49} + k_{89}, a[2][3][2] = k_{10} + k_{50} + k_{90}, a[2][4][3] = k_{11} + k_{51} + k_{91},$ $a[4][2][0] = k_{24} + k_{64} + v_2, a[4][2][1] = k_{25} + k_{65} + v_4, a[4][3][2] = k_{26} + k_{66} + v_7,$ $a[1][4][4] = k_4 + k_{44} + k_{84} + v_0, a[1][3][6] = k_6 + k_{46} + k_{86}, a[2][3][0] = k_8 + k_{48} + k_{88},$ $a[2][3][6] = k_{14} + k_{54} + k_{94}, a[3][4][0] = k_{16} + k_{56}, a[3][4][3] = k_{19} + k_{59},$ $a[4][2][5] = k_{29} + k_{69} + v_{12}, a[4][2][7] = k_{31} + k_{71} + v_{15}.$
Involved key bits
$k_{57}, k_{58}, k_{59}, k_{56}, k_{62}, k_8 + k_{23} + k_{63}, k_2 + k_{27} + k_{42} + k_{67} + k_{82}, k_2 + k_{27} + k_{42} + k_{67} + k_{82}$ $+ k_{75}, k_{15} + k_{30} + k_{55} + k_{70} + k_{95}, k_{15} + k_{30} + k_{55} + k_{70} + k_{95} + k_{63}, k_{12} + k_{27} + k_{52} + k_{67}$ $+ k_{92}, k_{12} + k_{27} + k_{52} + k_{67} + k_{92} + k_{60}, k_{13} + k_{28} + k_{53} + k_{68} + k_{93}, k_{13} + k_{28} + k_{53} + k_{68}$ $+ k_{93} + k_{61}, k_5 + k_{30} + k_{45} + k_{70} + k_{85}, k_5 + k_{30} + k_{45} + k_{70} + k_{85} + k_{78}, k_3 + k_{28} + k_{43}$ $+ k_{68} + k_{83}, k_5 + k_{13} + k_{20} + k_{45} + k_{60} + k_{85}.$

Cube variables
$a[0][3][5] = a[0][4][5] = v_0, a[2][3][4] = a[2][4][4] = v_1, a[2][3][5] = a[2][4][5] = v_2,$ $a[4][2][0] = v_3, a[4][3][0] = v_4, a[4][4][0] = v_3 + v_4, a[4][2][1] = v_5, a[4][3][1] = v_6,$ $a[4][4][1] = v_5 + v_6, a[4][2][2] = v_7, a[4][3][2] = v_8, a[4][4][2] = v_7 + v_8, a[4][2][3] = v_9,$ $a[4][3][3] = v_{10}, a[4][4][3] = v_9 + v_{10}, a[4][2][4] = v_{11}, a[4][3][4] = v_{12}, a[4][4][4] = v_{11} + v_{12},$ $a[4][2][5] = v_{13}, a[4][3][5] = v_{14}, a[4][4][5] = v_{13} + v_{14}, a[4][2][6] = a[4][3][6] = v_{15}.$
Auxiliary variables
$a[0][4][4] = k_{36} + k_{76}, a[0][3][5] = k_{37} + k_{77} + v_0, a[1][4][0] = k_0 + k_{40} + k_{80},$ $a[1][3][1] = k_1 + k_{41} + k_{81}, a[1][3][2] = k_2 + k_{42} + k_{82}, a[1][4][5] = k_5 + k_{45} + k_{85},$ $a[1][4][6] = k_6 + k_{46} + k_{86}, a[1][3][7] = k_7 + k_{47} + k_{87}, a[2][4][1] = k_9 + k_{49} + k_{89},$ $a[2][4][2] = k_{10} + k_{50} + k_{90}, a[2][4][3] = k_{11} + k_{51} + k_{91}, a[2][4][4] = k_{12} + k_{52} + k_{92} + v_1,$ $a[4][2][0] = k_{24} + k_{64} + v_3, a[4][3][1] = k_{25} + k_{65} + v_6, a[4][4][2] = k_{26} + k_{66} + v_7 + v_8,$ $a[4][4][3] = k_{27} + k_{67} + v_9 + v_{10}, a[4][2][6] = k_{30} + k_{70} + v_{15}, a[4][3][7] = k_{31} + k_{71}.$
Involved key bits
$k_{60}, k_{59}, k_{57}, k_{58}, k_{23} + k_{38} + k_{63} + k_{78}, k_{23} + k_{38} + k_{63} + k_{78} + k_{31} + k_{71}, k_{15} + k_{55} + k_{95}$ $k_3 + k_{28} + k_{68} + k_{43} + k_{83}, k_3 + k_{28} + k_{68} + k_{43} + k_{83} + k_{36} + k_{76}, k_8 + k_{48} + k_{88},$ $k_8 + k_{48} + k_{88} + k_{56}, k_8 + k_{48} + k_{88} + k_{33} + k_{73} + k_{81}, k_{13} + k_{53} + k_{93} + k_{28} + k_{68},$ $k_{13} + k_{53} + k_{93} + k_{28} + k_{68} + k_{21}, k_{13} + k_{53} + k_{93} + k_{28} + k_{68} + k_{61},$ $k_{14} + k_{54} + k_{94} + k_{29} + k_{69}, k_{14} + k_{54} + k_{94} + k_{29} + k_{69} + k_{22},$ $k_{14} + k_{54} + k_{94} + k_{29} + k_{69} + k_{62}, k_{34} + k_{74} + k_{82}, k_6 + k_{13} + k_{38} + k_{53} + k_{78} + k_{93}.$

Table 8: The third 16-dimensional cubes for KETJE Jr V1, where the corresponding (n_a, n_i) is $(26, 27)$ and (n'_a, n'_i) is $(4, 16)$. The accumulated number of key bits recovered is 74.

Cube variables
$a[0][3][0] = a[0][4][0] = v_0, a[0][3][3] = a[0][4][3] = v_1, a[0][3][4] = a[0][4][4] = v_2,$ $a[0][3][5] = a[0][4][5] = v_3, a[0][3][6] = a[0][4][6] = v_4, a[0][3][7] = a[0][4][7] = v_5,$ $a[1][3][0] = a[1][4][0] = v_6, a[1][3][1] = a[1][4][1] = v_7, a[2][3][3] = a[2][4][3] = v_8,$ $a[2][3][4] = a[2][4][4] = v_9, a[2][3][5] = a[2][4][5] = v_{10}, a[2][3][6] = a[2][4][6] = v_{11},$ $a[2][3][7] = a[2][4][7] = v_{12}, a[3][3][0] = a[3][4][0] = v_{13}, a[3][3][1] = a[3][4][1] = v_{14}.$ $a[3][3][7] = a[3][4][7] = v_{15}.$
Auxiliary variables
$a[0][3][0] = k_{32} + k_{72} + v_0, a[0][4][1] = k_{33} + k_{73}, a[0][3][3] = k_{35} + k_{75} + v_1,$ $a[0][3][4] = k_{36} + k_{76} + v_2, a[0][3][5] = k_{37} + k_{77} + v_3, a[0][4][6] = k_{38} + k_{78} + v_4,$ $a[0][3][7] = k_{39} + k_{79} + v_5, a[1][3][0] = k_0 + k_{40} + k_{80} + v_6, a[1][3][1] = k_1 + k_{41} + k_{81} + v_7,$ $a[1][3][2] = k_2 + k_{42} + k_{82}, a[2][4][2] = k_{10} + k_{50} + k_{90}, a[2][4][3] = k_{11} + k_{51} + k_{91} + v_8,$ $a[2][4][4] = k_{12} + k_{52} + k_{92} + v_9, a[2][4][5] = k_{13} + k_{53} + k_{93} + v_{10}, a[4][4][5] = k_{29} + k_{69}$ $a[2][4][6] = k_{14} + k_{54} + k_{94} + v_{11}, a[3][3][0] = k_{16} + k_{56} + v_{13}, a[3][3][1] = k_{17} + k_{57} + v_{14},$ $a[3][3][2] = k_{18} + k_{58}, a[3][3][5] = k_{21} + k_{61}, a[3][3][6] = k_{22} + k_{62}, a[3][3][7] = k_{23} + k_{63} + v_{15},$ $a[4][4][1] = k_{25} + k_{65}, a[4][4][2] = k_{26} + k_{66}, a[4][2][3] = k_{27} + k_{67}, a[4][3][4] = k_{28} + k_{68}.$
Involved key bits
$k_{30}, k_{29}, k_{22}, k_{21}, k_7, k_4, k_6, k_5, k_{74}, k_{75}, k_{20}, k_{89}, k_{90}, k_{84}, k_{83},$ $k_3 + k_{43} + k_{83} + k_{91}, k_{15} + k_{55} + k_{95} + k_0, k_{15} + k_{55} + k_{95} + k_{80},$ $k_{15} + k_{55} + k_{95} + k_{23} + k_{30} + k_{70}, k_4 + k_{12} + k_{19} + k_{44} + k_{59} + k_{84},$ $k_9 + k_{49} + k_{89} + k_{34} + k_{74}, k_9 + k_{49} + k_{89} + k_{34} + k_{74} + k_{82}, k_{20} + k_{28} + k_{60},$ $k_5 + k_{13} + k_{20} + k_{45} + k_{60} + k_{85}, k_8 + k_{48} + k_{88} + k_1, k_8 + k_{48} + k_{88} + k_{81},$ $k_8 + k_{48} + k_{88} + k_{71} + k_{16} + k_{31}.$

Table 9: Inequalities describing the column parity if the whole column can be placed with cube variables.

$$
\begin{aligned}
-D[x][z] - G[x][z] &\geq -1,\\
-A[x][0][z] + D[x][z] + G[x][z] &\geq 0,\\
-A[x][1][z] + D[x][z] + G[x][z] &\geq 0,\\
-A[x][2][z] + D[x][z] + G[x][z] &\geq 0,\\
-A[x][3][z] + D[x][z] + G[x][z] &\geq 0,\\
-A[x][4][z] + D[x][z] + G[x][z] &\geq 0,\\
A[x][0][z] + A[x][1][z] + A[x][2][z] + A[x][3][z] + A[x][4][z] - 2D[x][z] - G[x][z] &\geq 0.
\end{aligned}
$$

Table 10: The first two 16-dimensional cubes for KETJE Jr V2, where the corresponding (n_a, n_i) are $(15, 14)$ and $(15, 18)$, and (n'_a, n'_i) are $(15, 14)$ and $(9, 13)$, respectively. The accumulated numbers of key bits recovered are 29 and 47.

Cube variables
$a[0][1][0] = v_0, a[0][3][0] = v_1, a[0][4][0] = v_0 + v_1, a[0][1][1] = v_2, a[0][3][1] = v_3,$ $a[0][4][1] = v_2 + v_3, a[0][1][2] = v_4, a[0][3][2] = v_5, a[0][4][2] = v_4 + v_5, a[0][1][3] = v_6,$ $a[0][3][3] = v_7, a[0][4][3] = v_6 + v_7, a[0][1][4] = v_8, a[0][3][4] = v_9, a[0][4][4] = v_8 + v_9,$ $a[0][1][5] = v_{10}, a[0][3][5] = v_{11}, a[0][4][5] = v_{10} + v_{11}, a[0][1][6] = v_{12}, a[0][3][6] = v_{13},$ $a[0][4][6] = v_{12} + v_{13}, a[0][1][7] = v_{14}, a[0][3][7] = v_{15}, a[0][4][7] = v_{14} + v_{15}.$
Auxiliary variables
$a[0][4][3] = k_{51} + v_6 + v_7, a[0][4][4] = k_{52} + v_8 + v_9, a[0][4][5] = k_{53} + v_{10} + v_{11},$ $a[0][4][6] = k_{54} + v_{12} + v_{13}, a[0][4][7] = k_{55} + v_{14} + v_{15}, a[2][3][2] = k_{10} + k_{66} + k_{82},$ $a[2][3][3] = k_{11} + k_{67} + k_{83}, a[2][3][4] = k_{12} + k_{68} + k_{84}, a[2][3][5] = k_{13} + k_{69} + k_{85},$ $a[2][3][6] = k_{14} + k_{70} + k_{86}, a[3][4][0] = k_{16} + k_{32} + k_{88}, a[3][4][4] = k_{20} + k_{36} + k_{92},$ $a[3][4][5] = k_{21} + k_{37} + k_{93}, a[3][4][6] = k_{22} + k_{38} + k_{94}, a[3][4][7] = k_{23} + k_{39} + k_{95}.$
Involved key bits
$k_{75}, k_{76}, k_{77}, k_{78}, k_{79}, k_8 + k_{49} + k_{64} + k_{80}, k_8 + k_{49} + k_{64} + k_{73} + k_{80}, k_9 + k_{50} + k_{65} + k_{81},$ $k_9 + k_{50} + k_{65} + k_{74} + k_{81}, k_{15} + k_{48} + k_{71} + k_{87}, k_{15} + k_{48} + k_{71} + k_{72} + k_{87},$ $k_{17} + k_{33} + k_{48} + k_{89}, k_{18} + k_{34} + k_{49} + k_{90}, k_{19} + k_{35} + k_{50} + k_{91}.$

Cube variables
$a[2][0][0] = a[2][3][0] = v_0, a[2][0][1] = a[2][3][1] = v_1, a[2][0][2] = a[2][3][2] = v_2,$ $a[2][0][3] = a[2][3][3] = v_3, a[2][0][4] = a[2][3][4] = v_4, a[2][0][5] = a[2][3][5] = v_5,$ $a[2][0][6] = a[2][3][6] = v_6, a[2][0][7] = a[2][3][7] = v_7, a[4][0][0] = a[4][2][0] = v_8,$ $a[4][0][1] = a[4][2][1] = v_9, a[4][0][2] = a[4][2][2] = v_{10}, a[4][0][3] = a[4][2][3] = v_{11},$ $a[4][0][4] = a[4][2][4] = v_{12}, a[4][0][5] = a[4][2][5] = v_{13}, a[4][0][6] = a[4][2][6] = v_{14},$ $a[4][0][7] = a[4][2][7] = v_{15}.$
Auxiliary variables
$a[0][1][2] = k_{50}, a[0][4][6] = k_{54}, a[0][4][7] = k_{55}, a[1][2][5] = k_5 + k_{61} + k_{77},$ $a[1][4][0] = k_0 + k_{56} + k_{72}, a[1][4][3] = k_3 + k_{59} + k_{75}, a[2][3][1] = k_9 + k_{65} + k_{81} + v_1,$ $a[2][3][2] = k_{10} + k_{66} + k_{82} + v_2, a[2][3][5] = k_{13} + k_{69} + k_{85} + v_5,$ $a[2][3][6] = k_{14} + k_{70} + k_{86} + v_6, a[4][2][0] = k_{24} + k_{40} + v_8, a[4][2][1] = k_{25} + k_{41} + v_9,$ $a[4][2][4] = k_{28} + k_{44} + v_{12}, a[4][2][5] = k_{29} + k_{45} + v_{13}, a[4][2][6] = k_{30} + k_{46} + v_{14}.$
Involved key bits
$k_{51}, k_0 + k_{15} + k_{48} + k_{56} + k_{71} + k_{72} + k_{87}, k_1 + k_{26} + k_{42} + k_{57} + k_{73},$ $k_1 + k_{26} + k_{42} + k_{57} + k_{73} + k_{50}, k_2 + k_{27} + k_{43} + k_{58} + k_{74}, k_3 + k_{51} + k_{59} + k_{75},$ $k_4 + k_{60} + k_{76}, k_5 + k_{12} + k_{53} + k_{61} + k_{68} + k_{77} + k_{84}, k_6 + k_{31} + k_{47} + k_{62} + k_{78},$ $k_7 + k_{63} + k_{79}, k_8 + k_{64} + k_{80} + k_{49}, k_8 + k_{64} + k_{80} + k_{31} + k_{47},$ $k_{11} + k_{67} + k_{83} + k_{52}, k_{11} + k_{67} + k_{83} + k_{26} + k_{42}, k_{12} + k_{68} + k_{84} + k_{53},$ $k_{12} + k_{68} + k_{84} + k_{27} + k_{43}, k_{15} + k_{71} + k_{87}, k_{15} + k_{71} + k_{87} + k_{48}.$

Table 11: The third 16-dimensional cubes for KETJE Jr V2, where the corresponding (n_a, n_i) is $(26, 32)$ and (n'_a, n'_i) is $(1, 15)$. The accumulated number of key bits recovered is 63.

Cube variables
$a[0][1][0] = v_0, a[0][3][0] = v_1, a[0][4][0] = v_0 + v_1, a[0][1][1] = v_2, a[0][3][1] = v_3,$ $a[0][4][1] = v_2 + v_3, a[0][1][2] = v_4, a[0][3][2] = v_5, a[0][4][2] = v_4 + v_5, a[0][1][3] = v_6,$ $a[0][3][3] = v_7, a[0][4][3] = v_6 + v_7, a[0][1][4] = v_8, a[0][3][4] = v_9, a[0][4][4] = v_8 + v_9,$ $a[0][1][5] = v_{10}, a[0][3][5] = v_{11}, a[0][4][5] = v_{10} + v_{11}, a[0][3][6] = a[0][4][6] = v_{12},$ $a[0][3][7] = a[0][4][7] = v_{13}, a[1][2][0] = v_{14}, a[4][0][7] = v_{15}.$
Auxiliary variables
$a[0][3][0] = k_{48} + v_1, a[0][3][2] = k_{50} + v_5, a[0][3][5] = k_{53} + v_{11}, a[0][4][1] = k_{49} + v_2 + v_3,$ $a[0][4][3] = k_{51} + v_6 + v_7, a[0][4][4] = k_{52} + v_8 + v_9, a[0][4][6] = k_{54} + v_{12},$ $a[0][4][7] = k_{55} + v_{13}, a[1][2][0] = k_0 + k_{56} + k_{72} + v_{14}, a[2][0][2] = k_{10} + k_{66} + k_{82},$ $a[2][0][3] = k_{11} + k_{67} + k_{83}, a[2][0][4] = k_{12} + k_{68} + k_{84}, a[2][0][5] = k_{13} + k_{69} + k_{85},$ $a[2][3][0] = k_8 + k_{64} + k_{80}, a[2][3][1] = k_9 + k_{65} + k_{81}, a[2][3][6] = k_{14} + k_{70} + k_{86},$ $a[2][3][7] = k_{15} + k_{71} + k_{87}, a[3][1][2] = k_{18} + k_{34} + k_{90}, a[3][1][3] = k_{19} + k_{35} + k_{91},$ $a[3][1][6] = k_{22} + k_{38} + k_{94}, a[3][1][7] = k_{23} + k_{39} + k_{95}, a[3][4][0] = k_{16} + k_{32} + k_{88},$ $a[3][4][4] = k_{20} + k_{36} + k_{92}, a[3][4][5] = k_{21} + k_{37} + k_{93}, a[4][0][7] = k_{31} + k_{47} + v_{15},$ $a[4][2][1] = k_{25} + k_{41}.$
Involved key bits
$k_3, k_5, k_7, k_{27}, k_{31}, k_{40}, k_{46}, k_{56}, k_{61}, k_{63}, k_{72}, k_{73}, k_{74}, k_{75}, k_{76}, k_{77}, k_{78}, k_{79},$ $k_1 + k_{57} + k_{73}, k_1 + k_{57} + k_{73} + k_9 + k_{65} + k_{81}, k_2 + k_{10} + k_{17} + k_{33} + k_{58} + k_{74} + k_{89},$ $k_3 + k_{59} + k_{75}, k_3 + k_{59} + k_{75} + k_{83}, k_5 + k_{30} + k_{46} + k_{61} + k_{77}, k_7 + k_{63} + k_{71} + k_{79},$ $k_{17} + k_{33} + k_{89}, k_{17} + k_{33} + k_{89} + k_{25}, k_{17} + k_{33} + k_{89} + k_{25} + k_{41}, k_{22} + k_{29} + k_{45},$ $k_{23} + k_{30} + k_{39} + k_{46} + k_{95}, k_{24} + k_{33} + k_{40}, k_{28} + k_{44} + k_{93}, k_{30} + k_{46}.$

Table 12: A 32-dimensional cube for KETJE Jr V1 with 72-bit keys, where $(n_a, n_i) = (29, 34)$. In total, there are 57 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[1][2][0] = v_0, a[1][3][0] = v_1, a[1][4][0] = v_0 + v_1, a[1][2][1] = v_2, a[1][3][1] = v_3,$ $a[1][4][1] = v_2 + v_3, a[1][2][2] = v_4, a[1][3][2] = v_5, a[1][4][2] = v_4 + v_5, a[1][2][3] = v_6,$ $a[1][3][3] = v_7, a[1][4][3] = v_6 + v_7, a[1][2][4] = v_8, a[1][3][4] = v_9, a[1][4][4] = v_8 + v_9,$ $a[1][2][5] = v_{10}, a[1][3][5] = v_{11}, a[1][4][5] = v_{10} + v_{11}, a[1][2][6] = v_{12}, a[1][3][6] = v_{13},$ $a[1][4][6] = v_{12} + v_{13}, a[1][2][7] = v_{14}, a[1][3][7] = v_{15}, a[1][4][7] = v_{14} + v_{15},$ $a[3][3][0] = a[3][4][0] = v_{16}, a[3][2][7] = v_{17}, a[3][3][7] = v_{18}, a[3][4][7] = v_{17} + v_{18},$ $a[4][2][0] = a[4][4][0] = v_{19}, a[4][2][1] = v_{20}, a[4][3][1] = v_{21}, a[4][4][1] = v_{20} + v_{21},$ $a[4][2][2] = v_{22}, a[4][3][2] = v_{23}, a[4][4][2] = v_{22} + v_{23}, a[4][2][3] = v_{24}, a[4][3][3] = v_{25},$ $a[4][4][3] = v_{24} + v_{25}, a[4][2][4] = v_{26}, a[4][3][4] = v_{27}, a[4][4][4] = v_{26} + v_{27}, a[4][2][5] = v_{28},$ $a[4][3][5] = v_{29}, a[4][4][5] = v_{28} + v_{29}, a[4][2][6] = a[4][3][6] = v_{30}, a[4][2][7] = a[4][3][7] = v_{31}.$
Auxiliary variables
$a[0][3][3] = k_{35}, a[3][4][5] = k_{21} + k_{61}, a[2][2][0] = k_8 + k_{48}, a[2][2][2] = k_{10} + k_{50},$ $a[2][4][4] = k_{12} + k_{52}, a[0][4][5] = k_{37}, a[2][4][1] = k_9 + k_{49}, a[2][2][3] = k_{11} + k_{51},$ $a[2][2][5] = k_{13} + k_{53}, a[2][2][6] = k_{14} + k_{54}, a[3][4][1] = k_{17} + k_{57}, a[3][4][2] = k_{18} + k_{58},$ $a[3][4][3] = k_{19} + k_{59}, a[3][2][4] = k_{20} + k_{60}, a[3][4][6] = k_{22} + k_{62}, a[1][2][1] = k_1 + k_{41} + v_2,$ $a[1][4][0] = k_0 + k_{40} + v_0 + v_1, a[1][2][3] = k_3 + k_{43} + v_6, a[1][4][5] = k_5 + k_{45} + v_{10} + v_{11},$ $a[1][3][6] = k_6 + k_{46} + v_{13}, a[1][4][7] = k_7 + k_{47} + v_{14} + v_{15}, a[4][2][0] = k_{24} + k_{64} + v_{19},$ $a[3][4][7] = k_{23} + k_{63} + v_{17} + v_{18}, a[4][3][1] = k_{25} + k_{65} + v_{21}, a[4][3][5] = k_{29} + k_{69} + v_{29},$ $a[4][4][2] = k_{26} + k_{66} + v_{22} + v_{23}, a[4][4][4] = k_{28} + k_{68} + v_{26} + v_{27},$ $a[4][2][6] = k_{30} + k_{70} + v_{30}, a[4][3][7] = k_{31} + k_{71} + v_{31}.$
Involved key bits
$k_8, k_{11}, k_{13}, k_{14}, k_{15}, k_{28}, k_{32}, k_{33}, k_{34}, k_{36}, k_{38}, k_{39}, k_{56}, k_{57}, k_{58}, k_{59}, k_{61},$ $k_{62}, k_{16} + k_{56}, k_{16} + k_{56} + k_{39}, k_{16} + k_{56} + k_9, k_{16} + k_{56} + k_9 + k_{49}, k_4 + k_{37} + k_{44},$ $k_2 + k_{42}, k_2 + k_{42} + k_{10}, k_2 + k_{42} + k_{50}, k_2 + k_{42} + k_{10} + k_{50}, k_2 + k_{42} + k_{27} + k_{67},$ $k_2 + k_{42} + k_{27} + k_{67} + k_{35}, k_{27} + k_{67}, k_{27} + k_{67} + k_{60}, k_{27} + k_{67} + k_{60} + k_{20},$ $k_4 + k_{44}, k_4 + k_{44} + k_{12}, k_4 + k_{44} + k_{12} + k_{52}, k_{29} + k_{36}, k_{15} + k_{55} + k_{63}.$

Table 13: A 32-dimensional cube for KETJE Jr V2 with 80-bit keys, where $(n_a, n_i) = (22, 25)$. In total, there are 40 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[0][1][0] = v_0, a[0][1][1] = v_2, a[0][1][2] = v_4, a[0][1][3] = v_6,$ $a[0][3][0] = v_1, a[0][3][1] = v_3, a[0][3][2] = v_5, a[0][3][3] = v_7,$ $a[0][4][0] = v_0 + v_1, a[0][4][1] = v_2 + v_3, a[0][4][2] = v_4 + v_5, a[0][4][3] = v_6 + v_7,$ $a[0][1][4] = v_8, a[0][1][5] = v_{10}, a[0][1][6] = v_{12}, a[0][1][7] = v_{14},$ $a[0][3][4] = v_9, a[0][3][5] = v_{11}, a[0][3][6] = v_{13}, a[0][3][7] = v_{15},$ $a[0][4][4] = v_8 + v_9, a[0][4][5] = v_{10} + v_{11}, a[0][4][6] = v_{12} + v_{13}, a[0][4][7] = v_{14} + v_{15},$ $a[3][1][0] = v_{16}, a[3][1][1] = v_{18}, a[3][1][2] = v_{20}, a[3][1][3] = v_{22},$ $a[3][2][0] = v_{17}, a[3][2][1] = v_{19}, a[3][2][2] = v_{21}, a[3][2][3] = v_{23},$ $a[3][4][0] = v_{16} + v_{17}, a[3][4][1] = v_{18} + v_{19}, a[3][4][2] = v_{20} + v_{21}, a[3][4][3] = v_{22} + v_{23},$ $a[3][1][4] = v_{24}, a[3][1][5] = v_{26}, a[3][1][6] = v_{28}, a[3][1][7] = v_{30},$ $a[3][2][4] = v_{25}, a[3][2][5] = v_{27}, a[3][2][6] = v_{29}, a[3][2][7] = v_{31},$ $a[3][4][4] = v_{24} + v_{25}, a[3][4][5] = v_{26} + v_{27}, a[3][4][6] = v_{28} + v_{29}, a[3][4][7] = v_{30} + v_{31}.$
Auxiliary variables
$a[0][1][1] = k_{49} + v_2, a[0][3][7] = k_{55} + v_{15}, a[0][4][4] = k_{52} + v_8 + v_9,$ $a[0][1][3] = k_{51} + v_6, a[0][3][2] = k_{50} + v_5, a[0][4][0] = k_{48} + v_0 + v_1, a[0][1][6] = k_{54} + v_{12},$ $a[1][2][0] = k_0 + k_{56} + k_{72}, a[1][2][1] = k_1 + k_{57} + k_{73}, a[1][2][2] = k_2 + k_{58} + k_{74},$ $a[1][2][6] = k_6 + k_{62} + k_{78}, a[1][2][7] = k_7 + k_{63} + k_{79}, a[1][4][4] = k_4 + k_{60} + k_{76},$ $a[3][1][0] = k_{16} + k_{32} + v_{16}, a[3][1][4] = k_{20} + k_{36} + v_{24}, a[3][1][7] = k_{23} + k_{39} + v_{30},$ $a[3][2][1] = k_{17} + k_{33} + v_{19}, a[3][2][2] = k_{18} + k_{34} + v_{21}, a[3][2][5] = k_{21} + k_{37} + v_{27},$ $a[3][4][3] = k_{19} + k_{35} + v_{22} + v_{23}, a[0][4][5] = k_{53} + v_{10} + v_{11}, a[2][3][5] = k_{13} + k_{69}.$
Involved key bits
$k_{78}, k_{10} + k_{66}, k_{10} + k_{66} + k_{75}, k_3 + k_{59} + k_{75}, k_9 + k_{65}, k_9 + k_{65} + k_{74},$ $k_9 + k_{65} + k_{74} + k_2 + k_{58}, k_{15} + k_{71}, k_{15} + k_{71} + k_{72}, k_{15} + k_{71} + k_{72} + k_0 + k_{56},$ $k_{14} + k_{70}, k_{14} + k_{70} + k_{79}, k_{14} + k_{70} + k_{79} + k_7 + k_{63}, k_{11} + k_{67}, k_{11} + k_{67} + k_{76},$ $k_{11} + k_{67} + k_{76} + k_{60} + k_4, k_{12} + k_{68}, k_{12} + k_{68} + k_{77}, k_8 + k_{64}, k_8 + k_{64} + k_{73},$ $k_8 + k_{64} + k_{73} + k_{57} + k_1, k_5 + k_{61} + k_{77}, k_5 + k_{61} + k_{77} + k_{13} + k_{69}, k_{22} + k_{38},$ $k_6 + k_{54} + k_{62} + k_{78}.$

Table 14: A 64-dimensional cube for KETJE Sr V2, where $(n_a, n_i) = (33, 33)$. In total, there are 60 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[0][1][0] = v_0, a[0][1][8] = v_{16}, a[3][1][0] = v_{32}, a[3][1][8] = v_{48},$ $a[0][3][0] = v_1, a[0][3][8] = v_{17}, a[3][2][0] = v_{33}, a[3][2][8] = v_{49},$ $a[0][4][0] = v_0 + v_1, a[0][4][8] = v_{16} + v_{17}, a[3][4][0] = v_{32} + v_{33}, a[3][4][8] = v_{48} + v_{49},$ $a[0][1][1] = v_2, a[0][1][9] = v_{18}, a[3][1][1] = v_{34}, a[3][1][9] = v_{50},$ $a[0][3][1] = v_3, a[0][3][9] = v_{19}, a[3][2][1] = v_{35}, a[3][2][9] = v_{51},$ $a[0][4][1] = v_2 + v_3, a[0][4][9] = v_{18} + v_{19}, a[3][4][1] = v_{34} + v_{35}, a[3][4][9] = v_{50} + v_{51},$ $a[0][1][2] = v_4, a[0][1][10] = v_{20}, a[3][1][2] = v_{36}, a[3][1][10] = v_{52},$ $a[0][3][2] = v_5, a[0][3][10] = v_{21}, a[3][2][2] = v_{37}, a[3][2][10] = v_{53},$ $a[0][4][2] = v_4 + v_5, a[0][4][10] = v_{20} + v_{21}, a[3][4][2] = v_{36} + v_{37}, a[3][4][10] = v_{52} + v_{53},$ $a[0][1][3] = v_6, a[0][1][11] = v_{22}, a[3][1][3] = v_{38}, a[3][1][11] = v_{54},$ $a[0][3][3] = v_7, a[0][3][11] = v_{23}, a[3][2][3] = v_{39}, a[3][2][11] = v_{55},$ $a[0][4][3] = v_6 + v_7, a[0][4][11] = v_{22} + v_{23}, a[3][4][3] = v_{38} + v_{39}, a[3][4][11] = v_{54} + v_{55},$ $a[0][1][4] = v_8, a[0][1][12] = v_{24}, a[3][1][4] = v_{40}, a[3][1][12] = v_{56},$ $a[0][3][4] = v_9, a[0][3][12] = v_{25}, a[3][2][4] = v_{41}, a[3][2][12] = v_{57},$ $a[0][4][4] = v_8 + v_9, a[0][4][12] = v_{24} + v_{25}, a[3][4][4] = v_{40} + v_{41}, a[3][4][12] = v_{56} + v_{57},$ $a[0][1][5] = v_{10}, a[0][1][13] = v_{26}, a[3][1][5] = v_{42}, a[3][1][13] = v_{58},$ $a[0][3][5] = v_{11}, a[0][3][13] = v_{27}, a[3][2][5] = v_{43}, a[3][2][13] = v_{59},$ $a[0][4][5] = v_{10} + v_{11}, a[0][4][13] = v_{26} + v_{27}, a[3][4][5] = v_{42} + v_{43}, a[3][4][13] = v_{58} + v_{59},$ $a[0][1][6] = v_{12}, a[0][1][14] = v_{28}, a[3][1][6] = v_{44}, a[3][1][14] = v_{60},$ $a[0][3][6] = v_{13}, a[0][3][14] = v_{29}, a[3][2][6] = v_{45}, a[3][2][14] = v_{61},$ $a[0][4][6] = v_{12} + v_{13}, a[0][4][14] = v_{28} + v_{29}, a[3][4][6] = v_{44} + v_{45}, a[3][4][14] = v_{60} + v_{61},$ $a[0][1][7] = v_{14}, a[0][1][15] = v_{30}, a[3][1][7] = v_{46}, a[3][1][15] = v_{62},$ $a[0][3][7] = v_{15}, a[0][3][15] = v_{31}, a[3][2][7] = v_{47}, a[3][2][15] = v_{63},$ $a[0][4][7] = v_{14} + v_{15}, a[0][4][15] = v_{30} + v_{31}, a[3][4][7] = v_{46} + v_{47}, a[3][4][15] = v_{62} + v_{63}.$
Auxiliary variables
$a[0][1][2] = k_{106} + v_4, a[0][1][5] = k_{109} + v_{10}, a[0][1][8] = k_0 + k_{112} + v_{16}, a[0][1][12] = k_4 +$ $k_{116} + v_{24}, a[0][1][14] = k_6 + k_{118} + v_{28}, a[0][1][15] = k_7 + k_{119} + v_{30}, a[0][3][3] = k_{107} +$ $v_7, a[0][3][7] = k_{111} + v_{15}, a[0][3][9] = k_1 + k_{113} + v_{19}, a[0][3][10] = k_2 + k_{114} + v_{21},$ $a[0][4][4] = k_{108} + v_8 + v_9, a[0][4][11] = k_3 + k_{115} + v_{22} + v_{23}, a[1][0][0] = k_8 + k_{120},$ $a[1][0][1] = k_9 + k_{121}, a[1][0][5] = k_{13} + k_{125}, a[1][0][6] = k_{14} + k_{126}, a[1][2][7] = k_{15} + k_{127},$ $a[1][4][4] = k_{12} + k_{124}, a[2][4][1] = k_{25}, a[2][4][7] = k_{31}, a[2][4][13] = k_{37}, a[3][1][4] = k_{44} + k_{76}$ $+ v_{40}, a[3][1][5] = k_{45} + k_{77} + v_{42}, a[3][1][6] = k_{46} + k_{78} + v_{44}, a[3][1][15] = k_{55} + k_{87} +$ $v_{62}, a[3][2][8] = k_{48} + k_{80} + v_{49}, a[3][2][9] = k_{49} + k_{81} + v_{51}, a[3][2][11] = k_{51} + k_{83} +$ $v_{55}, a[3][2][12] = k_{52} + k_{84} + v_{57}, a[3][4][0] = k_{40} + k_{72} + v_{32} + v_{33}, a[3][4][3] = k_{43} +$ $k_{75} + v_{38} + v_{39}, a[3][4][10] = k_{50} + k_{82} + v_{52} + v_{53}, a[3][4][13] = k_{53} + k_{85} + v_{58} + v_{59}.$
Involved key bits
$k_{17}, k_{18}, k_{19}, k_{20}, k_{21}, k_{22}, k_{26}, k_{27}, k_{28}, k_{30}, k_{32}, k_{33}, k_{34}, k_{35}, k_{38},$ $k_5 + k_{36} + k_{117}, k_5 + k_{54} + k_{86} + k_{117}, k_8 + k_{39} + k_{104} + k_{120}, k_9 + k_{24} + k_{105} + k_{121},$ $k_{10} + k_{41} + k_{73} + k_{122}, k_{11} + k_{42} + k_{74} + k_{123}, k_{12} + k_{27} + k_{124}, k_{13} + k_{28} + k_{125},$ $k_{14} + k_{29} + k_{110} + k_{126}, k_{15} + k_{30} + k_{127}, k_{16} + k_{47} + k_{79}, k_{24} + k_{105}, k_{29} + k_{110},$ $k_{39} + k_{104}, k_{41} + k_{73} + k_{104}, k_{42} + k_{74} + k_{105}, k_{47} + k_{79} + k_{110}, k_{23} + k_{54} + k_{86}.$

Table 15: A 32-dimensional cube for the Xoodoo AE, where $(n_a, n_i) = (55, 55)$. In total, there are 106 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[0][1][24] = a[0][2][24] = v_0, a[1][1][1] = a[1][2][1] = v_1, a[1][1][2] = a[1][2][2] = v_2,$ $a[1][1][3] = a[1][2][3] = v_3, a[1][1][5] = a[1][2][5] = v_4, a[1][1][10] = a[1][2][10] = v_5,$ $a[1][1][14] = a[1][2][14] = v_6, a[1][1][15] = a[1][2][15] = v_7, a[1][1][16] = a[1][2][16] = v_8,$ $a[1][1][17] = a[1][2][17] = v_9, a[2][1][0] = v_{10}, a[2][2][0] = v_{11}, a[2][1][9] = v_{12}, a[2][2][9] = v_{13},$ $a[3][1][0] = a[3][2][0] = v_{14}, a[3][1][2] = a[3][2][2] = v_{15}, a[3][1][4] = a[3][2][4] = v_{16},$ $a[3][1][5] = a[3][2][5] = v_{17}, a[3][1][6] = a[3][2][6] = v_{18}, a[3][1][7] = a[3][2][7] = v_{19},$ $a[3][1][9] = a[3][2][9] = v_{20}, a[3][1][11] = a[3][2][11] = v_{21}, a[3][1][13] = a[3][2][13] = v_{22},$ $a[3][1][14] = a[3][2][14] = v_{23}, a[3][1][15] = a[3][2][15] = v_{24}, a[3][1][16] = a[3][2][16] = v_{25},$ $a[3][1][17] = a[3][2][17] = v_{26}, a[3][1][18] = a[3][2][18] = v_{27}, a[3][1][23] = a[3][2][23] = v_{28},$ $a[3][1][24] = a[3][2][24] = v_{29}, a[3][1][25] = a[3][2][25] = v_{30}, a[3][1][27] = a[3][2][27] = v_{31}.$
Auxiliary variables
$a[0][1][31] = k_{23}, a[0][2][6] = k_{126}, a[0][2][8] = k_0, a[0][2][15] = k_7, a[0][2][29] = k_{21},$ $a[1][1][1] = k_{25} + v_1, a[1][1][2] = k_{26} + v_2, a[1][1][6] = k_{30}, a[1][1][9] = k_{33}, a[1][1][12] = k_{36},$ $a[1][1][14] = k_{38} + v_6, a[1][1][15] = k_{39} + v_7, a[1][1][18] = k_{42}, a[1][1][19] = k_{43},$ $a[1][1][21] = k_{45}, a[1][1][22] = k_{46}, a[1][1][24] = k_{48}, a[1][1][26] = k_{50}, a[1][1][29] = k_{53},$ $a[1][2][0] = k_{24}, a[1][2][3] = k_{27} + v_3, a[1][2][4] = k_{28}, a[1][2][8] = k_{32}, a[1][2][10] = k_{34} + v_5,$ $a[1][2][11] = k_{35}, a[1][2][17] = k_{41} + v_9, a[1][2][20] = k_{44}, a[1][2][23] = k_{47}, a[1][2][30] = k_{54},$ $a[1][2][31] = k_{55}, a[2][1][7] = k_{63}, a[2][2][12] = k_{68}, a[2][2][21] = k_{77}, a[2][2][25] = k_{81},$ $a[2][2][30] = k_{86}, a[3][1][0] = k_{88} + v_{14}, a[3][1][2] = k_{90} + v_{15}, a[3][1][7] = k_{95} + v_{19},$ $a[3][1][8] = k_{96}, a[3][1][11] = k_{99} + v_{21}, a[3][1][12] = k_{100}, a[3][1][14] = k_{102} + v_{23},$ $a[3][1][16] = k_{104} + v_{25}, a[3][1][21] = k_{109}, a[3][1][22] = k_{110}, a[3][1][23] = k_{111} + v_{28},$ $a[3][1][25] = k_{113} + v_{30}, a[3][1][27] = k_{115} + v_{31}, a[3][1][30] = k_{118}, a[3][1][31] = k_{119},$ $a[3][2][9] = k_{97} + v_{20}, a[3][2][13] = k_{101} + v_{22}, a[3][2][18] = k_{106} + v_{27}, a[3][2][20] = k_{108},$ $a[3][2][29] = k_{117}.$
Involved key bits
$k_{64} + k_{73} + k_{110}, k_{52} + k_{57}, k_6 + k_{15} + k_{52}, k_{71} + k_{80} + k_{117}, k_1 + k_{92}, k_8 + k_{45} + k_{127}, k_8,$ $k_{89} + k_{112}, k_{67} + k_{76} + k_{113}, k_1 + k_{38} + k_{120}, k_{76} + k_{85} + k_{90}, k_{71}, k_{73}, k_{62} + k_{85} + k_{99},$ $k_{60} + k_{69} + k_{106}, k_{29} + k_{52} + k_{66}, k_3 + k_{40} + k_{122}, k_{58}, k_{70}, k_{66} + k_{75} + k_{112}, k_{59} + k_{105},$ $k_{89} + k_{112} + k_{126}, k_{10} + k_{92}, k_{91}, k_5, k_{64} + k_{87} + k_{101}, k_{17}, k_5 + k_{14} + k_{51}, k_{15}, k_9 + k_{91},$ $k_{19}, k_{114}, k_{58} + k_{67} + k_{104}, k_{22} + k_{36} + k_{127}, k_{37}, k_3 + k_{12} + k_{49}, k_3 + k_{94}, k_{72}, k_{76},$ $k_{69} + k_{78} + k_{115}, k_{60} + k_{83} + k_{97}, k_{59}, k_{61}, k_4 + k_{13} + k_{50}, k_7 + k_{89} + k_{98}, k_{67},$ $k_{70} + k_{79} + k_{116}, k_{78} + k_{87} + k_{92}, k_{74} + k_{83} + k_{88}, k_{57} + k_{66} + k_{103}, k_6, k_{62} + k_{71} + k_{108},$ $k_2 + k_{11} + k_{48}, k_{57} + k_{80} + k_{94}, k_{16} + k_{98} + k_{107}.$

Table 16: A 64-dimensional cube for Keccak-MAC-512, where $(n_a, n_i) = (46, 46)$. In total, there are 92 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[1][1][29] = v_0, a[1][1][35] = v_1, a[1][1][42] = v_2, a[1][1][45] = v_3, a[1][1][58] = v_4,$ $a[2][0][0] = a[2][1][0] = v_5, a[2][0][2] = a[2][1][2] = v_6, a[2][0][3] = a[2][1][3] = v_7,$ $a[2][0][5] = a[2][1][5] = v_8, a[2][0][6] = a[2][1][6] = v_9, a[2][0][7] = a[2][1][7] = v_{10},$ $a[2][0][9] = a[2][1][9] = v_{11}, a[2][0][12] = a[2][1][12] = v_{12}, a[2][0][13] = a[2][1][13] = v_{13},$ $a[2][0][15] = a[2][1][15] = v_{14}, a[2][0][16] = a[2][1][16] = v_{15}, a[2][0][18] = v_{16}, a[2][1][18] = v_{17},$ $a[2][0][19] = a[2][1][19] = v_{18}, a[2][0][21] = a[2][1][21] = v_{19}, a[2][0][22] = a[2][1][22] = v_{20},$ $a[2][0][25] = a[2][1][25] = v_{21}, a[2][0][26] = a[2][1][26] = v_{22}, a[2][0][28] = a[2][1][28] = v_{23},$ $a[2][0][29] = a[2][1][29] = v_{24}, a[2][0][32] = a[2][1][32] = v_{25}, a[2][0][34] = a[2][1][34] = v_{26},$ $a[2][0][35] = a[2][1][35] = v_{27}, a[2][0][38] = a[2][1][38] = v_{28}, a[2][0][41] = a[2][1][41] = v_{29},$ $a[2][0][42] = a[2][1][42] = v_{30}, a[2][0][44] = a[2][1][44] = v_{31}, a[2][0][45] = a[2][1][45] = v_{32},$ $a[2][0][48] = a[2][1][48] = v_{33}, a[2][0][50] = a[2][1][50] = v_{34}, a[2][0][51] = a[2][1][51] = v_{35},$ $a[2][0][54] = a[2][1][54] = v_{36}, a[2][0][55] = a[2][1][55] = v_{37}, a[2][0][56] = a[2][1][56] = v_{38},$ $a[2][0][57] = a[2][1][57] = v_{39}, a[2][0][58] = a[2][1][58] = v_{40}, a[2][0][60] = a[2][1][60] = v_{41},$ $a[2][0][61] = a[2][1][61] = v_{42}, a[2][0][62] = a[2][1][62] = v_{43}, a[2][0][63] = a[2][1][63] = v_{44},$ $a[3][0][2] = a[3][1][2] = v_{45}, a[3][0][8] = a[3][1][8] = v_{46}, a[3][0][15] = a[3][1][15] = v_{47},$ $a[3][0][18] = a[3][1][18] = v_{48}, a[3][0][24] = a[3][1][24] = v_{49}, a[3][0][30] = a[3][1][30] = v_{50},$ $a[3][0][31] = a[3][1][31] = v_{51}, a[3][0][34] = a[3][1][34] = v_{52}, a[3][0][37] = a[3][1][37] = v_{53},$ $a[3][0][38] = a[3][1][38] = v_{54}, a[3][0][40] = a[3][1][40] = v_{55}, a[3][0][43] = a[3][1][43] = v_{56},$ $a[3][0][44] = a[3][1][44] = v_{57}, a[3][0][47] = a[3][1][47] = v_{58}, a[3][0][50] = a[3][1][50] = v_{59},$ $a[3][0][53] = a[3][1][53] = v_{60}, a[3][0][54] = a[3][1][54] = v_{61}, a[3][0][56] = a[3][1][56] = v_{62},$ $a[3][0][60] = a[3][1][60] = v_{63}.$
Auxiliary variables
$a[0][1][1) = k_1, a[0][1][2] = k_2, a[0][1][3] = k_3, a[0][1][4] = k_4, a[0][1][5] = k_5, a[0][1][7] = k_7,$ $a[0][1][8] = k_8, a[0][1][10] = k_{10}, a[0][1][11] = k_{11}, a[0][1][12] = k_{12}, a[0][1][14] = k_{14},$ $a[0][1][17] = k_{17}, a[0][1][18] = k_{18}, a[0][1][20] = k_{20}, a[0][1][21] = k_{21}, a[0][1][23] = k_{23},$ $a[0][1][24] = k_{24}, a[0][1][28] = k_{28}, a[0][1][30] = k_{30}, a[0][1][31] = k_{31}, a[0][1][33] = k_{33},$ $a[0][1][34] = k_{34}, a[0][1][37] = k_{37}, a[0][1][39] = k_{39}, a[0][1][40] = k_{40}, a[0][1][41] = k_{41},$ $a[0][1][43] = k_{43}, a[0][1][44] = k_{44}, a[0][1][45] = k_{45}, a[0][1][46] = k_{46}, a[0][1][47] = k_{47},$ $a[0][1][49] = k_{49}, a[0][1][50] = k_{50}, a[0][1][51] = k_{51}, a[0][1][52] = k_{52}, a[0][1][53] = k_{53},$ $a[0][1][54] = k_{54}, a[0][1][55] = k_{55}, a[0][1][56] = k_{56}, a[0][1][58] = k_{58}, a[0][1][59] = k_{59},$ $a[0][1][60] = k_{60}, a[0][1][61] = k_{61}, a[0][1][62] = k_{62}, a[1][1][27] = k_{91}, a[1][1][36] = k_{100}.$
Involved key bits
$k_{81}, k_{98}, k_{85}, k_{71}, k_{27} + k_{91}, k_{75}, k_{101}, k_{97}, k_{95}, k_{68}, k_{87}, k_9 + k_{72}, k_{123}, k_{125}, k_{119}, k_{26} + k_{90},$ $k_{103}, k_{63} + k_{126}, k_{107}, k_{113}, k_{111}, k_{117}, k_{82}, k_{84}, k_{72}, k_{74}, k_{76}, k_{57} + k_{120}, k_{78}, k_{114}, k_{94}, k_{88}, k_{69},$ $k_{124}, k_{126}, k_{38} + k_{101}, k_{120}, k_{25} + k_{88}, k_{66}, k_{104}, k_{63} + k_{127}, k_{110}, k_{22} + k_{85}, k_{15} + k_{78}, k_{67}, k_{65}.$

Table 17: A 64-dimensional cube for KETJE Minor, where $(n_a, n_i) = (27, 27)$. In total, there are 52 bits key information involved in both auxiliary variables and involved key bits.

Cube varaibles
$a[2][1][1] = a[2][3][1] = v_0, a[2][0][2] = v_1, a[2][1][2] = v_2, a[2][3][2] = v_3,$ $a[2][4][2] = v_1 + v_2 + v_3, a[2][0][4] = v_4, a[2][1][4] = v_5, a[2][3][4] = v_4 + v_5,$ $a[2][1][7] = a[2][3][7] = v_6, a[2][0][8] = a[2][4][8] = v_7, a[2][0][10] = a[2][1][10] = v_8,$ $a[2][0][11] = a[2][4][11] = v_9, a[2][0][13] = v_{10}, a[2][1][13] = v_{11}, a[2][3][13] = v_{10} + v_{11},$ $a[2][0][16] = a[2][4][16] = v_{12}, a[2][0][19] = v_{13}, a[2][1][19] = v_{14}, a[2][3][19] = v_{13} + v_{14},$ $a[2][0][22] = v_{15}, a[2][1][22] = v_{16}, a[2][3][22] = v_{17}, a[2][4][22] = v_{15} + v_{16} + v_{17},$ $a[2][1][25] = v_{18}, a[2][3][25] = v_{19}, a[2][4][25] = v_{18} + v_{19}, a[2][0][28] = v_{20},$ $a[2][1][28] = v_{21}, a[2][4][28] = v_{20} + v_{21}, a[2][0][31] = v_{22}, a[2][1][31] = v_{23},$ $a[2][3][31] = v_{24}, a[2][4][31] = v_{22} + v_{23} + v_{24}, a[4][0][0] = v_{25}, a[4][1][0] = v_{26},$ $a[4][2][0] = v_{25} + v_{26}, a[4][0][1] = v_{27}, a[4][1][1] = v_{28}, a[4][2][1] = v_{29},$ $a[4][3][1] = v_{27} + v_{28} + v_{29}, a[4][2][2] = a[4][3][2] = v_{30}, a[4][0][6] = v_{31}, a[4][1][6] = v_{32},$ $a[4][2][6] = v_{31} + v_{32}, a[4][0][7] = v_{33}, a[4][1][7] = v_{34}, a[4][2][7] = v_{35},$ $a[4][3][7] = v_{33} + v_{34} + v_{35}, a[4][0][8] = v_{36}, a[4][2][8] = v_{37}, a[4][3][8] = v_{36} + v_{37},$ $a[4][0][9] = v_{38}, a[4][2][9] = v_{39}, a[4][3][9] = v_{38} + v_{39}, a[4][2][10] = a[4][3][10] = v_{40},$ $a[4][1][12] = a[4][3][12] = v_{41}, a[4][0][13] = a[4][2][13] = v_{42}, a[4][0][14] = v_{43}, a[4][1][14] = v_{44},$ $a[4][3][14] = v_{43} + v_{44}, a[4][0][15] = v_{45}, a[4][2][15] = v_{46}, a[4][3][15] = v_{45} + v_{46},$ $a[4][0][17] = v_{47}, a[4][1][17] = v_{48}, a[4][3][17] = v_{47} + v_{48}, a[4][0][19] = v_{49}, a[4][1][19] = v_{50},$ $a[4][2][19] = v_{49} + v_{50}, a[4][0][20] = v_{51}, a[4][1][20] = v_{52}, a[4][2][20] = v_{53},$ $a[4][3][20] = v_{51} + v_{52} + v_{53}, a[4][1][21] = a[4][3][21] = v_{54}, a[4][0][22] = a[4][1][22] = v_{55},$ $a[4][1][23] = a[4][3][23] = v_{56}, a[4][1][25] = a[4][2][25] = v_{57}, a[4][0][26] = v_{58}, a[4][1][26] = v_{59},$ $a[4][2][26] = v_{60}, a[4][3][26] = v_{58} + v_{59} + v_{60}, a[4][0][27] = a[4][3][27] = v_{61},$ $a[4][1][28] = a[4][3][28] = v_{62}, a[4][1][31] = a[4][2][31] = v_{63}.$
Auxiliary variables
$a[0][1][15] = k_7, a[1][0][17] = k_{41}, a[1][0][28] = k_{52}, a[1][2][16] = k_{40}, a[1][3][1] = k_{25},$ $a[1][3][7] = k_{31}, a[1][3][30] = k_{54}, a[1][4][9] = k_{33}, a[1][4][12] = k_{36}, a[2][0][0] = k_{56},$ $a[2][0][3] = k_{59}, a[2][0][22] = k_{78} + v_{15}, a[2][0][24] = k_{80}, a[2][1][10] = k_{66} + v_8,$ $a[2][1][14] = k_{70}, a[2][1][20] = k_{76}, a[2][1][25] = k_{81} + v_{18}, a[2][1][27] = k_{83}, a[2][3][6] = k_{62},$ $a[2][3][9] = k_{65}, a[2][3][13] = k_{69} + v_{10} + v_{11}, a[2][4][11] = k_{67} + v_9, a[2][4][15] = k_{71},$ $a[2][4][18] = k_{74}, a[2][4][23] = k_{79}, a[2][4][31] = k_{87} + v_{22} + v_{23} + v_{24}, a[4][0][2] = k_{122}.$
Involved key bits
$k_1 + k_{33} + k_{64}, k_4 + k_{36}, k_4, k_{73}, k_{19} + k_{82}, k_{34}, k_1 + k_{64}, k_{16}, k_{55} + k_{120}, k_{72}, k_{68}, k_{125},$ $k_{22} + k_{85}, k_{30} + k_{127}, k_{82}, k_{61}, k_{86}, k_{85}, k_{28} + k_{125}, k_{29} + k_{126}, k_{47}, k_{60} + k_{123}, k_{10} + k_{73},$ $k_{53}, k_{39}, k_{27} + k_{124}, k_{46}.$