MILP-aided Cube-attack-like Cryptanalysis on Keccak Keyed Modes

Wenquan Bi[1], Xiaoyang Dong[2], Zheng Li[1], Rui Zong[1], and Xiaoyun Wang[1,2]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education,Shandong University,Jinan,250100,China,
[2] Institute of Advanced Study, Tsinghua University, Beijing, 100084, China
biwenquan@mail.sdu.edu.cn

Abstract. Cube-attack-like cryptanalysis was proposed by Dinur et al. at EUROCRYPT 2015, which recovers the key of Keccak keyed modes in a divide-and-conquer manner. In their attack, one selects cube variables that are not multiplied with each other (denoted as *linear-cube*) with the method of construction. The chosen cube variables are consecutive bits in one lane and the key bits they multiply with are still too many, which leads to that one can attack less rounds sometimes. In this paper, we introduce a new MILP model to solve the above problem. Using this new MILP tool, we find the optimal *linear-cube*s for Keccak-MAC and Ketje that multiply with a minimum number of key bits in the first round. For example, when the capacity is 256, we find a new 32-dimension *linear-cube* for Keccak-MAC that only multiply with 18 key bits instead of Dinur et al.'s 64 bits and the complexity of the 6-round attack is reduced to 2^{42} from 2^{66}. More impressively, using this new tool, we give the very first 7-round key-recovery attack on Keccak-MAC-512. In addition, we get the best attacks on Ketje Major/Minor. For Ketje Major, when nonce is 9 lanes, we could improve the best previous 6-round attack to 7-round. We give the first 7-round attacks on Ketje Minor when the nonce is reduced to 9 lanes. When comparing with Huang et al.'s conditional cube attack, the MILP-aided cube-attack-like cryptanalysis have larger effective range and get the best results on the Keccak keyed variants with relatively smaller degrees of freedom.

Keywords: Keccak-MAC, Ketje, MILP, Cube attack

1 Introduction

As a countermeasure of the collision attacks on MD5 and SHA-1 by Wang et al. [27, 28], the U.S. National Institute of Standards and Technology (NIST) announced a public contest in 2007 aiming at the selection of a new standard for a cryptographic hash function (SHA-3). After 5 years of intensive scrutiny, in 2012 NIST selected Keccak as the winner of the SHA-3 competition. As one of the most important cryptographic standards, Keccak attracts lots of attention from the world wide researchers and engineers. Till now, many cryptanalysis results [7,8,15,16,20] and evaluation tools [6,11,19] have been proposed, including

the recent impressive collision attacks [22,25]. Since the robust design of Keccak, the cryptanalysis progress of Keccak is still limited.

At EUROCRYPT 2015, Dinur et al. [9] introduced a new cube-attack-like cryptanalysis technique and gave the security evaluations of the Keccak keyed modes for the first time. At CT-RSA 2015, Dobraunig et al. [12] evaluated the security of Ascon [13] against the cube-attack-like cryptanalysis. Later, Dong et al. [14] applied the cube-like method to Ketje Sr [3]. At EUROCRYPT 2017, Huang et al. [17] introduced the *conditional cube attack*, which takes advantage of the large state freedom of Keccak to find a so-called *conditional cube variable* that do not multiply with all the other *cube variables* (called *ordinary cube variables*) in the first round and second round of Keccak.

Recently, cryptographic communities found many classical cryptanalysis methods could be converted to mathematical optimization problems which aim to achieve the minimal or maximal value of an objective function under certain constraints. Mixed-integer Linear Programming (MILP) is the most widely studied technique to solve these optimization problems. One of the most successful applications of MILP is to search differential and linear trails. Mouha et al. [21] first applied MILP method to count active Sboxes of word-based block ciphers. Then, at Asiacrypt 2014, by deriving some linear inequalities through the H-Representation of the convex hull of all differential patterns of Sbox, Sun et al. [26] extended this technique to search differential and linear trails. Another two important applications are to search integral distinguisher [29] and impossible differentials [23] [5].

At Asiacrypt 2017, Li et al. [18] introduced a new MILP tool to improve conditional cube attacks on Keccak keyed modes. They found that when the conditional cube variable is given, to find enough ordinary cube variables is a mathematical optimization problem. They gave the MILP model and improved Huang et al.'s conditional cube attacks. And most recently, Song et al. [24] introduced a new MILP model to find better/optimal choices of conditional cubes. These works seem to exhibit a new way to research Keccak sponge function. Since Keccak's robust design, many classic cryptanalysis techniques become very complex to implement on Keccak. However, by using MILP model, one may send these tedious works to a MILP solver.

1.1 Our Contributions

In this paper, we find Dinur et al.'s [9] cube-attack-like cryptanalysis technique could also be converted to and improved by a MILP model. In Dinur et al.'s attack, the key point is to select the public variables of the cube in such a way that the superpolys depend only on a (relatively) small number of key bits. In detail, at the first round of Keccak, the attacker finds a set of cube variables that are not multiplied with each other (we denoted it as *linear-cube*), meanwhile, these cube variables are not multiplied with some key bits. By taking advantage of the CP-kernel, Dinur et al. find 32/64-dimension *linear-cube*s that are not multiplied with 64 key bits in Keccak-MAC with capacity 256.

In this paper, we propose a novel MILP model to search optimal *linear-cube*s that multiply with a minimum number of key bits in the first round. We model the so-called *CP-like-kernel*, model the way that the cube variables are not multiplied with each other in the first round and model the way that the cube variables are not multiplied with key bits, etc. We construct a linear inequality system. The target object is the minimum number of key bits which are multiplied with cube variables. Based on this new MILP tool, we find the optimal cubes that are multiplied with fewest key bits for Keccak-MAC and Ketje. All the results improve Dinur et al.'s attacks.

When comparing with Huang et al.'s conditional cube attack, the advantage of the MILP-aided cube-attack-like cryptanalysis is that it has larger effective range. The conditional cube attack becomes much weaker or invalid when the of degrees of freedom are small[3]. Hence, the conditional cube attack can only applied to 6-round on Keccak-MAC-512. However, MILP-aided cube-attack-like cryptanalysis could not only attack the same rounds with conditional cube attack with the same degree of freedom, but also get best results on the Keccak keyed variants with relatively smaller degree of freedom. The results are summarized in Table 1. In addition, we list the source code of the new MILP tools and the verification programs in a public domain[4] to help researchers study Keccak. Our main results achieved by the MILP tools are listed below.

1. When the capacity is 256, we find the optimal 32-dimension *linear-cube* for Keccak-MAC that only multiply with 18 key bits instead of Dinur et al.'s 64 bits. By divide-and-conquer manner, the complexity of the 6-round attack is only 2^{42} instead of 2^{66}. we find a new 64-dimension *linear-cube* that only multiply with 30 key bits instead of Dinur et al.'s 64 bits. Based on it, the complexity of 7-round cube-attack-like cryptanalysis is reduced by a factor of 2^{17}. We could also improve Dinur et al.'s 5-round attack on Keccak-MAC with capacity 576 to 7-round, and get 7-round cube-attack-like on Keccak-MAC-384, but we omit these results due to page limit.
2. In Keccak sponge function, when the capacity reaches 1024, the degree of freedom is so small that the cryptanalysis becomes quite hard. Actually, the rounds of the collision attack and preimage attack on Keccak-512 are only 3 and 4, respectively. For Keccak-MAC-512, the cryptanalysis results are also the weakest. In fact, at EUROCRYPT 2015, Dinur et al. only give cube-attack-like cryptanalysis when the capacity is smaller than 576. At EUROCRYPT 2017, Huang et al. give the first 5-round key-recovery attack on Keccak-MAC-512 using conditional cube attack. Then at Asiacrypt 2017, Li et al. give a 6-round conditional cube attack. In this paper, using our new MILP tool, we give the first 7-round key-recovery attack on Keccak-MAC-512.
3. In addition, we also get the best attacks on Ketje Major/Minor with nonce reduced settings. For Ketje Major, when nonce is 9-lane, we improve the

[3] In Keccak-MAC, the capacity is larger, the degrees of freedom are smaller; in Ketje, the nonce or size of state are smaller, the degrees of freedom are smaller.

[4] `https://github.com/biwenquan/MILP-aided-Cube-attack-like-cryptanalysis/`

best previous 6-round attack to 7-round. We give the first 7-round attacks on Ketje Minor when the nonce is reduced to 288 bits, while the best previous 7-round attacks need 654-bit nonce.

Table 1: Summary of Key Recovery Attacks on Keccak Keyed Modes

Variant	Capacity	Type	Attacked Rounds	Time	Memory	Source
Keccak-MAC	256	Cube-attack-like	6	2^{66}	2^{32}	[9]
		Conditional Cube Attack	6	2^{40}	–	[17]
		Balanced Divide-and-Conquer	6	2^{45}	2^{13}	[30]
		MILP-aided Cube-attack-like	6	2^{42}	2^{9}	Sect. 6
		Cube-attack-like	7	2^{97}	2^{32}	[9]
		Conditional Cube Attack	7	2^{72}	–	[17]
		Balanced Divide-and-Conquer	7	2^{84}	2^{67}	[30]
		MILP-aided Cube-attack-like	7	2^{80}	2^{15}	Sect. 6
	1024	Conditional Cube Attack	5	2^{24}	–	[17]
		Conditional Cube Attack	6	$2^{58.3}$	–	[18]
		Conditional Cube Attack	6	2^{41}	–	[24]
		MILP-aided Cube-attack-like	6	2^{59}	2^{26}	Sect. 6
		MILP-aided Cube-attack-like	7	$2^{112.6}$	2^{47}	Sect. 6
Variant	Nonce	Type	Attacked Rounds	Time	Memory	Source
Ketje Minor	654	Cube-attack-like	7	2^{96}	2^{32}	[14]
	654	Conditional Cube Attack	7	2^{81}	–	[18]
	288	MILP-aided Cube-attack-like	7	2^{113}	2^{48}	Sect .7.1
Ketje Major	576	Conditional Cube Attack	6	2^{41}	–	[18]
	576	MILP-aided Cube-attack-like	7	2^{94}	2^{29}	Sect .7.2

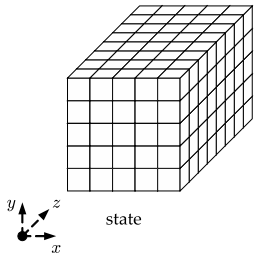
1.2 Organization of the Paper

Sect. 2 gives some notations, and a brief description on Keccak-permutations, Keccak-MAC and Ketje. Some related works are introduced in Sect. 3. Sect. 4 introduces the idea of improvement of Dinur et al.'s attack. Sect. 5 describes the MILP search model for cube-like-attack. Round-reduced key-recovery attacks on Keccak-MAC-512 are introduced in Sect. 6. Sect. 7 gives the applications to Ketje. Sect. 8 concludes this paper.

2 Preliminaries

2.1 Notations

S_i	the intermediate state after i-round of Keccak-p, for example $S_{0.5}$ means the intermediate state before χ in 1st round of Keccak-p
A	used in tables: for Keccak-MAC, the initial state; for Ketje, the state after π^{-1} of Keccak-p^*
$A[i][j]$	the 32/64-bit word indexed by $[i, j, *]$ of state A, $0 \leqslant i \leqslant 4$, $0 \leqslant j \leqslant 4$
$A[i][j][k]$	the bit indexed by $[i, j, k]$ of state A
v_i	the ith cube variable
a_i	the ith auxiliary variable used in the attack procedure
K	128-bit key, for Keccak-MAC, $K = k_0 \| k_1$, both k_0 and k_1 are 64-bit; for Ketje Major, $K = k_0 \| k_1 \| k_2$, k_0 is 56-bit, k_1 is 64-bit, k_2 is 8-bit; for Ketje Minor, $K = k_0 \| k_1 \| k_2 \| k_3 \| k_4$, k_0 is 24-bit, k_1,k_2 and k_3 are 32-bit, k_4 is 8-bit
$k_i[j]$	the jth bit of k_i



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

Fig. 1: (a) The Keccak State [2], (b) State A In 2-dimension

2.2 The Keccak-p permutations

The Keccak-p permutations are derived from the Keccak-f permutations [2] and have a tunable number of rounds. A Keccak-p permutation is defined by its width $b = 25 \times 2^l$, with $b \in \{25, 50, 100, 200, 400, 800, 1600\}$, and its number of rounds n_r, denoted as Keccak-$p[b]$. The round function $\mathtt{R}$ consists of five operations:

$$\mathtt{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

$\theta : A[x][y] = A[x][y] \oplus \sum_{j=0}^{4} (A[x-1][j] \oplus (A[x+1][j] \lll 1))$.
$\rho : A[x][y] = A[x][y] \lll r[x, y]$.
$\pi : A[y][2x + 3y] = A[x][y]$.
$\chi : A[x][y] = A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y]$.
$\iota : A[0][0] = A[0][0] \oplus RC$.

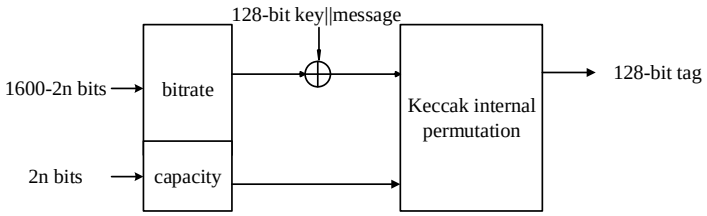


Fig. 2: Construction of Keccak-MAC-n

Keccak-$p[b]$ works on a state A of size b, which can be represented as 5×5 $\frac{b}{25}$-bit lanes, as depicted in Fig. 1, $A[i][j]$ with i for the index of column and j for the index of row. In what follows, indexes of i and j are in set $\{0, 1, 2, 3, 4\}$ and they are working in modulo 5 without other specification.

Table 2: Rotation constants $r[x, y]$ in Keccak ρ operation.

	x=0	x = 1	x = 2	x = 3	x = 4
y = 0	0	1	62	28	27
y = 1	36	44	6	55	20
y = 2	3	10	43	25	39
y = 3	41	45	15	21	8
y = 4	18	2	61	56	14

In Ketje v2, *the twisted permutations*, Keccak-$p^*[b]$=$\pi\circ$ Keccak-$p[b]\circ\pi^{-1}$, are introduced to effectively re-order the bits in the state. π^{-1} is the inverse of π: $\pi^{-1} : A[x + 3y][x] = A[x][y]$.

2.3 Keccak-MAC

A MAC mode of Keccak can be obtained by adding key as the prefix of message/nonce. As depicted in Fig. 2, the input of Keccak-MAC-n is concatenation of key and message and n is half of the capacity length.

2.4 Ketje

Ketje [3] is also one of the 16 candidates in the 3rd round CAESAR competition. It is a sponge-like construction.

The structure of Ketje is an authenticated encryption mode MonkeyWrap, shown Fig. 3, which is based on MonkeyDuplex [4]. It consists of four parts: the initialization phase, Processing associated data, Processing the plaintext and Finalization phase. The initialization takes the secret key K, the public nonce N and some paddings as the initial state. Then $n_{start} = 12$ rounds Keccak-p^* is applied. Our attack is applied to the initialization phase of Ketje.

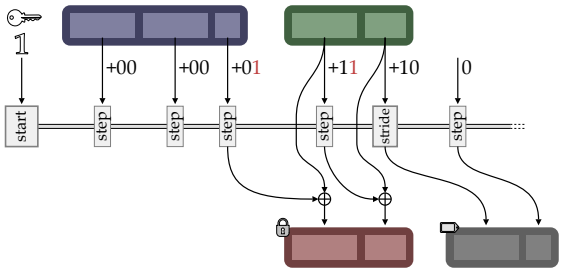


Fig. 3: Wrapping a Header and a Body with MonkeyWrap [3]

In Ketje v2, four concrete instances are proposed, shown in Table 3. $n_{start} = 12$, $n_{step} = 1$ and $n_{stride} = 6$. For Ketje Minor and Major, the recommended key length is 128-bit, so the maximal length of nonce is (800-128-18=)654 and (1600-128-18=)1454 bits.

Table 3: Four Instances in Ketje v2

Name	f	ρ	Main use case
Ketje Jr	Keccak-p^*[200]	16	lightweight
Ketje Sr	Keccak-p^*[400]	32	lightweight
Ketje Minor	Keccak-p^*[800]	128	lightweight
Ketje Major	Keccak-p^*[1600]	256	high performance

3 Related Work

3.1 Cube Attack

At EUROCRYPT 2009, Dinur and Shamir introduced the cube attack [10], in which the output bit of a symmetric cryptographic scheme can be regarded as a polynomial $f(k_0, ..., k_{n-1}, v_0, ..., v_{m-1})$ over $GF(2)$, $k_0, ..., k_{n-1}$ are the secret variables (the key bits), $v_0, ..., v_{m-1}$ are the public variables (e.g. IV or nonce bits).

Theorem 1. *(Dinur, Shamir [10]) Given a polynomial f: $X^n \to \{0,1\}$ of degree d, suppose that $0 < k < d$ and t is the monomial $x_0...x_{k-1}$. Write f as*

$$f(x) = t \cdot P_t(x) + Q_t(x) \tag{1}$$

where none of the terms in Q_t is divisible by t. Note that deg $P_t \leq d - k$. Then the sum of f over all values of the cube C_t (defined by t) is

$$\sum_{x'=(x_0,...,x_{k-1}) \in C_t} f(x', x) = P_t(1, 1, ...1, x_k, ...x_{n-1}) \tag{2}$$

whose degree is at most d-k (or 1 if $k = d - 1$),where C_t contains all binary vectors of the length k.

The basic idea is to find good C_t whose P is linear and not a constant. This enables the key recovery through solving a system of linear equations.

3.2 Dinur et al.'s Cube-attack-like Attack

At EUROCRYPT 2015, Dinur et al. [9] launched a cube-attack-like cryptanalysis on Keccak keyed modes. In the attack on 6-round reduced Keccak-MAC with capacity 256, the 128-bit key is placed in the lane $A[0,0]$ and $A[1,0]$. They find if the cube variables are in $A[2,2]$ and $A[2,3]$ which are equal in the same column, shown in Fig. 4, after θ, ρ and π, the cube variables are only multiplied with 64-bit key in $A[0,0]$ after the first round. The cube sums after 6-round are independent of the key bits in $A[1,0]$.

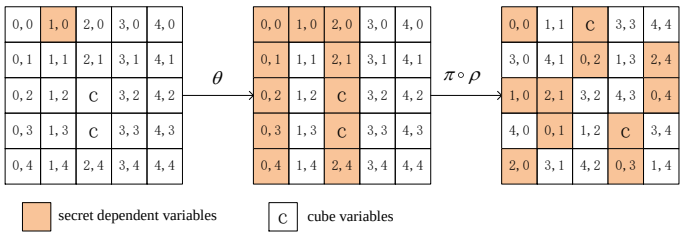


Fig. 4: Dinur et al.'s Work

In addition, Dinur et al. introduce 32 bits auxiliary variables which are assumed to be equal to key bits in $A[0,0]$ in the same column. Hence, half of $A[0,0]$ (32-bit key) and auxiliary variables are in CP-kernel, which makes that cube variables do not multiply with those key bits and auxiliary variables in the first round. So only 32 key bits will multiply with the cube variables after the first round, which means only 32 key bits will affect the cube sums of the output after 6-round.

The whole 6-round attack is as follows. In preprocessing phase, the attacker calculates the cube sums for each of 32-bit keys which multiply with cube variables and store them in a list L. In the online phase, for 2^{32} values of auxiliary variables, the attacker calculates the cube sums for the output bits and search them in L, for each match in L return the corresponding 32-bit key as a candidate. Similar attack is applied to 7-round Keccak-MAC. For more details, please refer to [9].

4 An Improvement of Dinur et al.'s Idea

In Dinur et al.'s divide-and-conquer strategy, the cube was chosen by hand and not optimal. If we choose the cube variables more precisely, the number of secret key bits, which multiply with the cube variables in the first round, will decrease and the complexity would be reduced as well so that we could attack more rounds even. We call these secret key bits *related key bits* (and other secret key bits which do not multiply with cube variables in the first round called *unrelated key bits*). This problem has been studied by Ye et al. [30], now we describe this idea as follows.

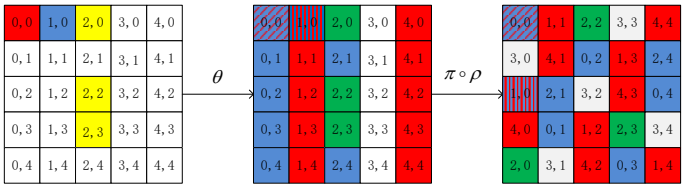


Fig. 5: The diffusion of key bits and cube variables in one round

As shown in Fig. 5, for example, we set the 128-bit secret key in $A[0][0] = k_0$ and $A[1][0] = k_1$ for 1600-bit-state Keccak-MAC. If we select 32-dimension cube variables as follows:

$$\begin{cases} A[2][0][3 \cdot i] = v_i, \\ A[2][2][3 \cdot i] = v_{i+16}, \\ A[2][3][3 \cdot i] = v_i + v_{i+16} \end{cases} \quad for\ i = 0, 1...15 \tag{3}$$

Firstly, we explore how the cube variables multiply with key bits in the lane-size. After θ operation, these lanes in red are diffused by k_0 and the lanes in blue are diffused by k_1, while the cube variables in green is just added by the key k_1, not multiplied with them. After ρ and π operation, we could see that only three lanes diffused by k_0 will multiply with cube variables (in green), other lanes especially the k_1 would do not multiply with cube variables in the first round.

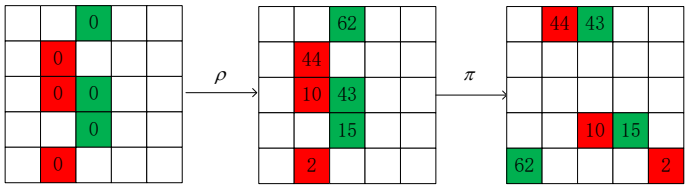


Fig. 6: The offset of key bits and cube variables in one round

Secondly, we explore how many bits in k_0 (only in 3 lanes) would multiply with these cube variables. As the ρ operation rotates a different offset (Table 2) for different lane, we use the number in each lane denote the rotated offset compared with the initial state, as shown in Fig. 6. In the χ operation, the cube variables in $A[2][0]$ (in the initial state $A[2][0]$) would multiply with the key bits $k_0[3 \cdot i + 62 - 2] \pmod{64}$ for $0 \leq i \leq 15$, the cube variables in $A[2][2]$ would multiply with the key bits $k_0[3 \cdot i + 43 - 44] \pmod{64}$ for $0 \leq i \leq 15$, and the cube variables in $A[2][3]$ would multiply with the key bits $k_0[3 \cdot i + 15 - 10] \pmod{64}$ for $0 \leq i \leq 15$, we list the key bits they multiplied in Table 4 for each lane. We can see that the key bits are repeated greatly with each other for different lane.

Table 4: The key bits multiplied with the 32-dimension cube variable

Lane	Key bits
$A[2][0]$	$k_0[2], k_0[5], k_0[8], k_0[11], k_0[14], k_0[17], k_0[20], k_0[23],$ $k_0[26], k_0[29], k_0[32], k_0[35], k_0[38], k_0[41], k_0[60], k_0[63]$
$A[2][2]$	$k_0[2], k_0[5], k_0[8], k_0[11], k_0[14], k_0[17], k_0[20], k_0[23],$ $k_0[26], k_0[29], k_0[32], k_0[35], k_0[38], k_0[41], k_0[44], k_0[63]$
$A[2][3]$	$k_0[5], k_0[8], k_0[11], k_0[14], k_0[17], k_0[20], k_0[23], k_0[26],$ $k_0[29], k_0[32], k_0[35], k_0[38], k_0[41], k_0[44], k_0[47], k_0[50]$

As a result, the new 32-dimension *linear cube* just multiplies with only 19-bit key bits instead of Dinur et al.'s 64 key bits. However, this *linear cube* is found by hand and is not an optimal cube that multiplied with minimum key bits. Obviously, it is hard to find such optimal 32 or 64 dimension *linear cube* by hand. In this paper, we introduce the novel MILP method to solve the above optimization problem and then improve cube-attack-like cryptanalysis on Keccak keyed modes a lot.

5 MILP Modeling Search Strategy

In this section, we present how to model our search strategy using the MILP method. For any bit $A[x][y][z]$ in the Keccak-p initial state, we define $A[x][y][z] = 1$ when it is a *cube variable* or a *related key bit*.

Since we need *linear cube*s in the first round, we need constraints to make the cube variables do not multiply in the first round, and the following inequalities are sufficient to model this:

$$A[x_1][y_1][z_1] + A[x_2][y_2][z_2] \leq 1 \tag{4}$$

which means if there are two bits $A[x_1][y_1][z_1]$ and $A[x_2][y_2][z_2]$ multiply with each other, we only keep at most one of them as cube variables.

In order to control the diffusion of the cube variables and make the most of freedom variables, we make use of the *CP-like-kernel* which was formalized by

Guo et al. [16] and studied by the related work [18]. We keep the sum of the variables within the same column is constant (usually zero) which makes the following θ be identity, hence the diffusion of the variables is reduced largely. As the number of cube variables in a column is at least 2, the following inequalities are sufficient to model the CP-like-kernel:

$$\begin{cases} \sum_{y=0}^{4} A[x][y][z] - 2d[x][z] \geq 0 \\ d[x][z] - A[x][i][z] \geq 0 \quad 0 \leq i \leq 4 \end{cases} \tag{5}$$

where the auxiliary variable $d[x][z]$ records whether the column $[x][z]$ contain cube variables as illustrated above. The $[x][z]$ column can provide $\sum_{y=0}^{4} A[x][y][z] - d[x][z]$ independent cube variables. As we need enough cube variables for our attack, for example, 64 cube variables for 7-round Keccak-MAC, we sum up the the state with freedom for cube variables and make it equal to 64. That is,

$$\sum_{x,y,z} A[x][y][z] - \sum_{x,z} d[x][z] = 2^{n-1}, \tag{6}$$

i.e. the number of cube variables in the freedom state.

When a cube variable multiplies with a key bit $A[x][y][z]$, we name this key bit as a *related key bit*, and set $A[x][y][z] = 1$. If one key bit $A[x_1][y_1][z_1]$ multiplies with a *cube variable* $A[x_2][y_2][z_2]$, we need the following inequality to constraint a related key bit:

$$A[x_1][y_1][z_1] - A[x_2][y_2][z_2] \geq 0 \tag{7}$$

Since we would like to get the minimum of the related key bits when the round number is given, We set the objective function as:

$$Min \sum_{x,y,z} A[x][y][z]. \tag{8}$$

Now we have the objective function and all the inequalities above as constrains, thus we get a complete MILP model which could be solved by the openly available software Gurobi [1].

6 Applications to round-reduced Keccak-MAC

6.1 Attack on 6/7-round Keccak-MAC-128

For Keccak-MAC-128 with 1600-bit state, rate occupies 1344 bits, and capacity 256 bits. As Fig. 7 shows us, 128-bit key (k_0, k_1) locates at the first two yellow lanes, then the white bits represent nonce or message bits, all of which can be selected as *cube variables*, while the grey ones are initialized with all zero.

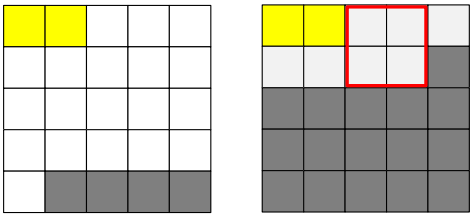


Fig. 7: The Initial State of Keccak-MAC-128(left) and Keccak-MAC-512(right)

According to the modeling search strategy illustrated in Section 5, we search for the minimize number of related key bits. The objective function is

$$\sum_{y=0,x\in\{0,1\},z\in\{0,1...63\}} A[x][y][z],$$

To model the CP-like-kernel, apply the equations (5) in Section **??** to all the freedom initial state to get the corresponding constrains. The input state is initialized with key k, possible *cube variables* v_i (placed in bit position $[x_i][y_i][z_i]$) and zero padding. After the first round, the state is in the algebraic symbolic form of the initial state bits. If any v_iv_j exists, the bit corresponding to i, j, constraint $A[x_i][y_i][z_i] + A[x_j][y_j][z_j] \leq 1$ is added. The above constraints avoid any multiplication in the first round among cube variables. Additionally, if any k_iv_j exists, the bit corresponding to i, j, constraint $A[x_i][y_i][z_i] - A[x_j][y_j][z_j] \geq 0$ is added which makes sure that the key bit $A[x_i][y_i][z_i]$ is a *related key bit* if $A[x_j][y_j][z_j]$ is chosen as a *cube variable*.

With the help of Gurobi [1], the objective function is optimized under all the above constraints, the minimum of related key bits is 18 for 32-dimension *linear cube* and 30 for 64-dimension *linear cube*. The cube variables and related bits are listed in Table 5 and Table 6.

Attack on 6-round Keccak-MAC-128. The attack includes preprocessing phase and online phase. The related key bits in the Table 5 are multiplied with the cube variables in the first round, and the cube variables are not multiplies with other secret key bits, which means that the other secret key bits have no influence on the cube sums of the 2^{32} different messages. Among these 18 related key bits, we guess the 9 *guessed key bits* in Table 5 in the preprocessing phase, and for the other related key bits except the guessed key bits, we set auxiliary variables in the same column for each key bit. When the auxiliary variables are equal to the key bits in the same column, these bits act as CP-kernel and the diffusion of these key bits is reduced and not multiplied with cube variables so that the cube sums do not depend on these key bits. We choose the auxiliary variables precisely either and examine that their related key bits matched (in the same column) do not multiply with the cube variables in the first round when the auxiliary variables equal to their matched related key bits [5]. We present the attack procedure as follows:

[5] `https://github.com/biwenquan/MILP-aided-Cube-attack-like-cryptanalysis/`

Table 5: Parameters set for attack on 6-round Keccak-MAC-128

Cube Variables
$A[2][2][9]=A[2][3][9]=v_0$,$A[2][0][12]=v_1$,$A[2][2][12]=v_2$,$A[2][3][12]=v_1+v_2$, $A[2][0][15]=v_3$,$A[2][2][15]=v_4$,$A[2][3][15]=v_3+v_4$,$A[2][0][18]=v_5$,$A[2][2][18]=v_6$, $A[2][3][18]=v_5+v_6$,$A[2][0][21]=v_7$,$A[2][2][21]=v_8$,$A[2][3][21]=v_7+v_8$,$A[2][0][24]=v_9$, $A[2][2][24]=v_{10}$,$A[2][3][24]=v_9+v_{10}$,$A[2][0][27]=v_{11}$,$A[2][2][27]=v_{12}$, $A[2][3][27]=v_{11}+v_{12}$,$A[2][0][30]=v_{13}$,$A[2][2][30]=v_{14}$,$A[2][3][30]=v_{13}+v_{14}$, $A[2][0][33]=v_{15}$,$A[2][2][33]=v_{16}$,$A[2][3][33]=v_{15}+v_{16}$,$A[2][0][36]=v_{17}$,$A[2][2][36]=v_{18}$, $A[2][3][36]=v_{17}+v_{18}$,$A[2][0][39]=v_{19}$,$A[2][2][39]=v_{20}$,$A[2][3][39]=v_{19}+v_{20}$, $A[2][0][42]=v_{21}$,$A[2][2][42]=v_{22}$,$A[2][3][42]=v_{21}+v_{22}$,$A[2][0][45]=v_{23}$,$A[2][2][45]=v_{24}$, $A[2][3][45]=v_{23}+v_{24}$,$A[2][0][48]=v_{25}$,$A[2][2][48]=v_{26}$,$A[2][3][48]=v_{25}+v_{26}$, $A[2][0][51]=v_{27}$,$A[2][2][51]=v_{28}$,$A[2][3][51]=v_{27}+v_{28}$,$A[2][0][54]=A[2][2][54]=v_{29}$, $A[2][0][57]=v_{30}$,$A[2][2][57]=v_{31}$,$A[2][3][57]=v_{30}+v_{31}$
Related Key Bits
$k_0[8]$,$k_0[11]$,$k_0[14]$,$k_0[17]$,$k_0[20]$, $k_0[23]$,$k_0[26]$,$k_0[29]$,$k_0[32]$,$k_0[35]$,$k_0[38]$,$k_0[41]$,$k_0[44]$, $k_0[47]$,$k_0[50]$,$k_0[53]$,$k_0[56]$,$k_0[62]$
Guessed Key bits
$k_0[35]$,$k_0[38]$,$k_0[41]$,$k_0[44]$, $k_0[47]$,$k_0[50]$,$k_0[53]$,$k_0[56]$,$k_0[62]$
Auxiliary Variables
$A[0][1][8]=a_0$,$A[0][1][11]=a_1$,$A[0][1][14]=a_2$,$A[0][1][17]=a_3$,$A[0][1][20]=a_4$, $A[0][1][23]=a_5$,$A[0][1][26]=a_6$,$A[0][1][29]=a_7$,$A[0][1][32]=a_8$

Preprocessing Phase.

- Set all the state bits except the cube variables to zero (or any other arbitrary constant).
- For each possible value of 9 guessed key bits in Table 5, calculate the 2^{32} cube sums after 6 rounds for all the output bits according to the 32-dimension cube variables in Table 5. Store the cube sums in a sorted list L with the value of 9-bit guessed key.

In preprocessing phase we calculate 2^9 cube sums for 32-dimension cube variables, so there needs $2^9 \times 2^{32} = 2^{41}$ 6-round Keccak-MAC, and the memory complexity is 2^9.

Online Phase.

- For each possible value of 9-bit auxiliary variables list in Table 5, request the outputs of 2^{32} messages that make up the 32-dimension cube variables.
- Calculate the cube sums for the output bits and search them in list L. For each match in L, regard the 9-bit guessed key and 9-bit auxiliary variables as the candidate for the 18-bit related key in the Table 5.

Once the value of the 9-bit auxiliary variables equal to the 9-bit other related key bits except the 9-bit guessed key bits, these 9-bit related key with auxiliary variables would have no influence on the cube sums as they are not multiply with the 32-dimension cube variables any more in the first round. Then only the

9-bit guessed key affects on the cube sums. The memory complexity is 2^9, the data complexity is 2^{32}, and total computation of this attack is $2^9 \times 2^{32} \times 2 = 2^{42}$, which is less than Dinur et al.'s 2^{66}.

Table 6: Parameters set for attack on 7-round Keccak-MAC-128

Cube Variables
$A[0][1][4]=A[0][3][4]=v_0$, $A[0][1][13]A[0][3][13]=v_1$,$A[0][1][17]=A[0][2][17]=v_2$, $A[0][2][18]=A[0][3][18]=v_3$, $A[0][1][22]A[0][3][22]=v_4$,$A[0][2][24]=A[0][3][24]=v_5$, $A[0][1][26]=A[0][2][26]=v_6$, $A[0][1][32]A[0][2][32]=v_7$,$A[0][1][35]=A[0][2][35]=v_8$, $A[0][1][38]=A[0][2][38]=v_9$,$A[0][2][39]=A[0][3][39]=v_{10}$,$A[0][1][41]=A[0][2][41]=v_{11}$, $A[0][1][44]=A[0][3][44]=v_{12}$,$A[0][2][48]=A[0][3][48]=v_{13}$,$A[0][1][50]=A[0][3][50]=v_{14}$, $A[0][1][59]=A[0][3][59]=v_{15}$,$A[0][1][61]=A[0][2][61]=v_{16}$,$A[2][0][0]=A[2][2][0]=v_{18}$, $A[2][3][0]=v_{17}+v_{18}$,$A[2][0][1]=v_{19}$,$A[2][2][1]=v_{20}$,$A[2][3][1]=v_{19}+v_{20}$, $A[2][0][3]=A[2][3][3]=v_{21}$,$A[2][0][4]=A[2][2][4]=v_{22}$,$A[2][0][7]=A[2][2][7]=v_{24}$, $A[2][3][7]=v_{23}+v_{24}$,$A[2][0][9]=A[2][2][9]=v_{26}$,$A[2][3][9]=v_{25}+v_{26}$, $A[2][0][12]=A[2][3][12]=v_{27}$,$A[2][2][15]=A[2][3][15]=v_{28}$,$A[2][0][16]=A[2][3][16]=v_{29}$, $A[2][0][18]=v_{30}$,$A[2][2][18]=v_{31}$,$A[2][3][18]=v_{30}+v_{31}$,$A[2][2][20]=A[2][3][20]=v_{32}$, $A[2][0][21]=A[2][2][21]=v_{33}$,$A[2][0][24]=A[2][2][24]=v_{34}$,$A[2][0][25]=A[2][3][25]=v_{35}$, $A[2][0][29]=A[2][3][29]=v_{36}$,$A[2][0][34]=A[2][3][34]=v_{37}$,$A[2][2][35]=A[2][3][35]=v_{38}$, $A[2][0][38]=A[2][3][38]=v_{39}$,$A[2][2][40]=A[2][3][40]=v_{40}$,$A[2][2][41]=A[2][3][41]=v_{41}$, $A[2][0][43]=A[2][3][43]=v_{42}$,$A[2][0][44]=v_{43}$,$A[2][2][44]=v_{44}$,$A[2][3][44]=v_{43}+v_{44}$, $A[2][2][46]=A[2][3][46]=v_{45}$,$A[2][0][47]=v_{46}$,$A[2][2][47]=v_{47}$,$A[2][3][47]=v_{46}+v_{47}$, $A[2][0][49]=v_{48}$,$A[2][2][49]=v_{49}$,$A[2][3][49]=v_{48}+v_{49}$,$A[2][0][50]=A[2][2][50]=v_{50}$, $A[2][0][52]=v_{51}$,$A[2][2][52]=v_{52}$,$A[2][3][52]=v_{51}+v_{52}$,$A[2][0][53]=v_{53}$, $A[2][2][53]=v_{54}$, $A[2][3][53]=v_{53}+v_{54}$,$A[2][0][55]=v_{55}$,$A[2][2][55]=v_{56}$, $A[2][3][55]=v_{55}+v_{56}$, $A[2][0][56]=A[2][3][56]=v_{57}$,$A[2][0][58]=v_{58}$,$A[2][2][58]=v_{59}$, $A[2][3][58]=v_{58}+v_{59}$, $A[2][2][59]=A[2][3][59]=v_{60}$,$A[2][0][61]=A[2][2][61]=v_{61}$, $A[2][0][62]=v_{62}$,$A[2][2][62]=v_{63}$, $A[2][3][62]=v_{62}+v_{63}$,
Related Key Bits
$k_0[0]$,$k_0[3]$,$k_0[5]$,$k_0[6]$,$k_0[8]$,$k_0[12]$,$k_0[14]$,$k_0[17]$,$k_0[19]$, $k_0[20]$,$k_0[21]$,$k_0[23]$,$k_0[25]$, $k_0[30]$,$k_0[34]$,$k_0[39]$,$k_0[40]$, $k_0[43]$,$k_0[45]$,$k_0[46]$,$k_0[48]$,$k_0[49]$,$k_0[51]$,$k_0[52]$,$k_0[54]$, $k_0[57]$,$k_0[58]$,$k_0[60]$,$k_0[61]$,$k_0[63]$
Guessed Key bits
$k_0[17]$,$k_0[40]$,$k_0[43]$,$k_0[45]$,$k_0[46]$,$k_0[48]$,$k_0[49]$,$k_0[51]$,$k_0[52]$, $k_0[54]$,$k_0[57]$,$k_0[58]$, $k_0[60]$,$k_0[61]$,$k_0[63]$,
Auxiliary Variables
$A[0][1][0]=a_0$, $A[0][1][3]=a_1$,$A[0][1][5]=a_2$,$A[0][1][6]=a_3$,$A[0][1][8]=a_4$, $A[0][1][12]=a_5$,$A[0][1][14]=a_6$,$A[0][1][19]=a_7$,$A[0][1][20]=a_8$,$A[0][1][21]=a_9$, $A[0][1][23]=a_{10}$,$A[0][1][25]=a_{11}$,$A[0][1][30]=a_{12}$,$A[0][1][34]=a_{13}$,$A[0][1][39]=a_{14}$

For 7-round Keccak-MAC-128, we find 30 related key bits with 64-dimension cube variables which are listed in Table 6 as well as the guessed key bits and auxiliary variables. The attack procedure is just like the 6-round attack. In the preprocessing phase, we compute 2^{64} cube sums for each value of 15-bit guessed key bits and store them in list L. In the online phase, we compute the

cube sums for each of the 15-bit auxiliary variables. The total time complexity for 7-round attack is $2^{15} \times 2^{64} \times 2 = 2^{80}$ and the memory complexity is 2^{15}.

6.2 Attack on 6/7-round Keccak-MAC-512

For Keccak-MAC-512 with 1600-bit state, rate occupies 576 bits, and capacity 1024 bits. As Fig. 7 shows us, 128-bit key (k_0, k_1) locates at the first two yellow lanes, then the white bits represent nonce bits, but only white ones highlighted by red thick lines can be selected as *cube variables* because the other white lanes do not satisfy the 'CP-like-kernel' and diffuse badly. In fact, we could found enough cube variables in those lanes highlighted by red trick lines. the grey ones are initialized with all zero.

Table 7: Parameters set for attack on 6-round Keccak-MAC-512

Cube Variables
$A[2][0][2]=A[2][1][2]=v_0$, $A[2][0][3]=A[2][1][3]=v_1$,$A[2][0][9]=A[2][1][9]=v_2$, $A[2][0][10]=A[2][1][10]=v_3$,$A[2][0][11]=A[2][1][11]=v_4$,$A[2][0][17]=A[2][1][17]=v_5$, $A[2][0][19]=A[2][1][19]=v_6$,$A[2][0][20]=A[2][1][20]=v_7$,$A[2][0][26]=A[2][1][26]=v_8$, $A[2][0][28]=A[2][1][28]=v_9$,$A[2][0][29]=A[2][1][29]=v_{10}$,$A[2][0][35]=A[2][1][35]=v_{11}$, $A[2][0][36]=A[2][1][36]=v_{12}$,$A[2][0][37]=A[2][1][37]=v_{13}$,$A[2][0][43]=A[2][1][43]=v_{14}$, $A[2][0][45]=A[2][1][45]=v_{15}$,$A[2][0][51]=A[2][1][51]=v_{16}$,$A[2][0][52]=A[2][1][52]=v_{17}$, $A[2][0][58]=A[2][1][58]=v_{18}$,$A[2][0][59]=A[2][1][59]=v_{19}$,$A[3][0][0]=A[3][1][0]=v_{20}$, $A[3][0][7]=A[3][1][7]=v_{21}$,$A[3][0][25]=A[3][1][25]=v_{22}$,$A[3][0][32]=A[3][1][32]=v_{23}$, $A[3][0][40]=A[3][1][40]=v_{24}$,$A[3][0][41]=A[3][1][41]=v_{25}$,$A[3][0][47]=A[3][1][47]=v_{26}$, $A[3][0][48]=A[3][1][48]=v_{27}$,$A[3][0][49]=A[3][1][49]=v_{28}$,$A[3][0][55]=A[3][1][55]=v_{29}$, $A[3][0][57]=A[3][1][57]=v_{30}$,$A[3][0][63]=A[3][1][63]=v_{31}$
Related Key Bits
$k_0[0]$,$k_0[5]$,$k_0[6]$,$k_0[7]$,$k_0[8]$,$k_0[13]$,$k_0[14]$,$k_0[15]$, $k_0[16]$,$k_0[22]$,$k_0[24]$,$k_0[25]$,$k_0[31]$,$k_0[32]$, $k_0[33]$,$k_0[34]$,$k_0[39]$, $k_0[40]$,$k_0[41]$,$k_0[42]$,$k_0[47]$,$k_0[48]$,$k_0[50]$,$k_0[54]$,$k_0[55]$,$k_0[56]$, $k_0[57]$, $k_0[62]$,$k_0[63]$,$k_1[0]$,$k_1[7]$,$k_1[8]$,$k_1[14]$,$k_1[15]$,$k_1[16]$, $k_1[18]$,$k_1[22]$,$k_1[24]$,$k_1[25]$,$k_1[30]$, $k_1[31]$,$k_1[33]$,$k_1[34]$,$k_1[38]$, $k_1[40]$,$k_1[41]$,$k_1[42]$,$k_1[48]$,$k_1[50]$,$k_1[56]$,$k_1[57]$,$k_1[63]$
Guessed Key bits
$k_0[57]$,$k_0[62]$,$k_0[63]$,$k_1[0]$,$k_1[7]$,$k_1[8]$,$k_1[14]$,$k_1[15]$,$k_1[16]$, $k_1[18]$,$k_1[22]$,$k_1[24]$,$k_1[25]$, $k_1[30]$,$k_1[31]$,$k_1[33]$,$k_1[34]$,$k_1[38]$, $k_1[40]$,$k_1[41]$,$k_1[42]$,$k_1[48]$,$k_1[50]$,$k_1[56]$,$k_1[57]$,$k_1[63]$
Auxiliary Variables
$A[0][2][0]=a_0$,$A[0][2][5]=a_1$,$A[0][2][6]=a_2$,$A[0][2][7]=a_3$,$A[0][2][8]=a_4$,$A[0][2][13]=a_5$, $A[0][2][14]=a_6$,$A[0][2][15]=a_7$,$A[0][2][16]=a_8$,$A[0][2][22]=a_9$, $A[0][2][24]=a_{10}$, $A[0][2][25]=a_{11}$,$A[0][2][31]=a_{12}$,$A[0][2][32]=a_{13}$,$A[0][2][33]=a_{14}$, $A[0][2][34]=a_{15}$, $A[0][2][39]=a_{16}$,$A[0][2][40]=a_{17}$,$A[0][2][41]=a_{18}$,$A[0][2][42]=a_{19}$, $A[0][2][47]=a_{20}$, $A[0][2][48]=a_{21}$,$A[0][2][50]=a_{22}$,$A[0][2][54]=a_{23}$,$A[0][2][55]=a_{24}$, $A[0][2][56]=a_{25}$

We use our MILP tools and find 32-dimension cube variables with 52 related key bits and 64-dimension cube variables with 95 related key bits respectively. The cube variables, related key bits, guessed key bits (for 7-round attack are

Table 8: Parameters set for attack on 7-round Keccak-MAC-512

Cube Variables
$A[2][0][1]=A[2][1][1]=v_0$,$A[2][0][2]=A[2][1][2]=v_1$,$A[2][0][8]=A[2][1][8]=v_2$, $A[2][0][9]=A[2][1][9]=v_3$,$A[2][0][10]=A[2][1][10]=v_4$,$A[2][0][11]=A[2][1][11]=v_5$, $A[2][0][12]=A[2][1][12]=v_6$,$A[2][0][13]=A[2][1][13]=v_7$,$A[2][0][14]=A[2][1][14]=v_8$, $A[2][0][20]=A[2][1][20]=v_9$,$A[2][0][21]=A[2][1][21]=v_{10}$,$A[2][0][22]=A[2][1][22]=v_{11}$, $A[2][0][23]=A[2][1][23]=v_{12}$,$A[2][0][24]=A[2][1][24]=v_{13}$,$A[2][0][25]=A[2][1][25]=v_{14}$, $A[2][0][26]=A[2][1][26]=v_{15}$,$A[2][0][33]=A[2][1][33]=v_{16}$,$A[2][0][34]=A[2][1][34]=v_{17}$, $A[2][0][35]=A[2][1][35]=v_{18}$,$A[2][0][36]=A[2][1][36]=v_{19}$,$A[2][0][37]=A[2][1][37]=v_{20}$, $A[2][0][38]=A[2][1][38]=v_{21}$,$A[2][0][41]=A[2][1][41]=v_{22}$,$A[2][0][46]=A[2][1][46]=v_{23}$, $A[2][0][47]=A[2][1][47]=v_{24}$,$A[2][0][48]=A[2][1][48]=v_{25}$,$A[2][0][49]=A[2][1][49]=v_{26}$, $A[2][0][50]=A[2][1][50]=v_{27}$,$A[2][0][53]=A[2][1][53]=v_{28}$,$A[2][0][54]=A[2][1][54]=v_{29}$, $A[2][0][59]=A[2][1][59]=v_{30}$,$A[2][0][60]=A[2][1][60]=v_{31}$,$A[2][0][61]=A[2][1][61]=v_{32}$, $A[2][0][62]=A[2][1][62]=v_{33}$,$A[2][0][63]=A[2][1][63]=v_{34}$,$A[3][0][0]=A[3][1][0]=v_{35}$, $A[3][0][1]=A[3][1][1]=v_{36}$,$A[3][0][7]=A[3][1][7]=v_{37}$,$A[3][0][10]=A[3][1][10]=v_{38}$, $A[3][0][11]=A[3][1][11]=v_{39}$,$A[3][0][12]=A[3][1][12]=v_{40}$,$A[3][0][13]=A[3][1][13]=v_{41}$, $A[3][0][14]=A[3][1][14]=v_{42}$,$A[3][0][22]=A[3][1][22]=v_{43}$,$A[3][0][23]=A[3][1][23]=v_{44}$, $A[3][0][24]=A[3][1][24]=v_{45}$,$A[3][0][25]=A[3][1][25]=v_{46}$,$A[3][0][26]=A[3][1][26]=v_{47}$, $A[3][0][34]=A[3][1][34]=v_{48}$,$A[3][0][35]=A[3][1][35]=v_{49}$,$A[3][0][36]=A[3][1][36]=v_{50}$, $A[3][0][37]=A[3][1][37]=v_{51}$,$A[3][0][38]=A[3][1][38]=v_{52}$,$A[3][0][39]=A[3][1][39]=v_{53}$, $A[3][0][46]=A[3][1][46]=v_{54}$,$A[3][0][47]=A[3][1][47]=v_{55}$,$A[3][0][49]=A[3][1][49]=v_{56}$, $A[3][0][50]=A[3][1][50]=v_{57}$,$A[3][0][51]=A[3][1][51]=v_{58}$,$A[3][0][52]=A[3][1][52]=v_{59}$, $A[3][0][58]=A[3][1][58]=v_{60}$,$A[3][0][59]=A[3][1][59]=v_{61}$,$A[3][0][62]=A[3][1][62]=v_{62}$, $A[3][0][63]=A[3][1][63]=v_{63}$
Related Key Bits
$k_0[0]$,$k_0[1]$,$k_0[2]$,$k_0[3]$,$k_0[4]$,$k_0[5]$,$k_0[6]$,$k_0[7]$, $k_0[8]$,$k_0[9]$,$k_0[10]$,$k_0[13]$,$k_0[14]$,$k_0[15]$, $k_0[16]$,$k_0[17]$,$k_0[18]$, $k_0[19]$,$k_0[20]$,$k_0[21]$,$k_0[22]$,$k_0[25]$,$k_0[26]$,$k_0[27]$,$k_0[28]$,$k_0[29]$, $k_0[30]$,$k_0[31]$,$k_0[32]$,$k_0[33]$,$k_0[34]$,$k_0[37]$,$k_0[38]$,$k_0[39]$,$k_0[40]$, $k_0[41]$,$k_0[42]$,$k_0[43]$, $k_0[44]$,$k_0[45]$,$k_0[46]$,$k_0[49]$,$k_0[50]$,$k_0[51]$, $k_0[52]$,$k_0[53]$,$k_0[54]$,$k_0[55]$,$k_0[56]$,$k_0[57]$, $k_0[58]$,$k_0[59]$,$k_0[61]$, $k_0[62]$,$k_1[0]$,$k_1[1]$,$k_1[2]$,$k_1[3]$,$k_1[4]$,$k_1[5]$,$k_1[6]$,$k_1[7]$,$k_1[13]$,$k_1[14]$, $k_1[15]$,$k_1[16]$,$k_1[17]$,$k_1[18]$,$k_1[19]$,$k_1[25]$,$k_1[26]$,$k_1[27]$,$k_1[28]$, $k_1[29]$,$k_1[30]$,$k_1[31]$, $k_1[32]$,$k_1[38]$,$k_1[39]$,$k_1[40]$,$k_1[41]$,$k_1[42]$, $k_1[43]$,$k_1[44]$,$k_1[45]$,$k_1[46]$,$k_1[51]$,$k_1[52]$, $k_1[53]$,$k_1[54]$,$k_1[55]$, $k_1[56]$,$k_1[57]$,$k_1[58]$,$k_1[59]$
Auxiliary Variables
$A[0][1][0]=a_0$,$A[0][1][1]=a_1$,$A[0][1][2]=a_2$,$A[0][1][3]=a_3$,$A[0][1][4]=a_4$,$A[0][1][5]=a_5$, $A[0][1][6]=a_6$,$A[0][1][7]=a_7$,$A[0][1][8]=a_8$,$A[0][1][9]=a_9$,$A[0][1][10]=a_{10}$, $A[0][1][13]=a_{11}$,$A[0][1][14]=a_{12}$,$A[0][1][15]=a_{13}$,$A[0][1][16]=a_{14}$,$A[0][1][17]=a_{15}$, $A[0][1][18]=a_{16}$,$A[0][1][19]=a_{17}$,$A[0][1][20]=a_{18}$,$A[0][1][21]=a_{19}$,$A[0][1][22]=a_{20}$, $A[0][1][25]=a_{21}$,$A[0][1][26]=a_{22}$,$A[0][1][27]=a_{23}$,$A[0][1][28]=a_{24}$,$A[0][1][29]=a_{25}$, $A[0][1][30]=a_{26}$,$A[0][1][31]=a_{27}$,$A[0][1][32]=a_{28}$,$A[0][1][33]=a_{29}$,$A[0][1][34]=a_{30}$, $A[0][1][37]=a_{31}$,$A[0][1][38]=a_{32}$,$A[0][1][39]=a_{33}$,$A[0][1][40]=a_{34}$,$A[0][1][41]=a_{35}$, $A[0][1][42]=a_{36}$,$A[0][1][43]=a_{37}$,$A[0][1][44]=a_{38}$,$A[0][1][45]=a_{39}$,$A[0][1][46]=a_{40}$, $A[0][1][49]=a_{41}$,$A[0][1][50]=a_{42}$,$A[0][1][51]=a_{43}$,$A[0][1][52]=a_{44}$,$A[0][1][53]=a_{45}$, $A[0][1][54]=a_{46}$,$A[0][1][55]=a_{47}$

listed in Table 9) and the auxiliary variables for 6/7-round attack are list in Table 7 and 8 respectively.

The attack procedure is the same as the Keccak-MAC-128. We just discuss the complexity here.

For 6-round Keccak-MAC-512, there are 52 related key bits in total and 26 guessed key bits guessed in the preprocessing phase. The auxiliary variables are also 26-bit, then the time complexity is both $2^{26} \times 2^{32} = 2^{58}$ for preprocessing phase and online phase, so the time complexity is 2^{59} totally and the memory complexity is 2^{26}.

For 7-round Keccak-MAC-512, there are 95 related key bits. We choose 47-bit as guessed key bits and find auxiliary variables for the other 48 bits. The time complexity of 7-round attack on Keccak-MAC-512 is $2^{47} \times 2^{64} + 2^{48} \times 2^{64} = 2^{112.6}$, and the memory complexity is 2^{47}.

Table 9: Guessed Key Bits for attack on 7-round Keccak-MAC-512

Guessed Key Bits
$k_0[56]$,$k_0[57]$,$k_0[58]$,$k_0[59]$,$k_0[61]$, $k_0[62]$,$k_1[0]$,$k_1[1]$,$k_1[2]$,$k_1[3]$,$k_1[4]$,$k_1[5]$,$k_1[6]$, $k_1[7]$,$k_1[13]$,$k_1[14]$, $k_1[15]$,$k_1[16]$,$k_1[17]$,$k_1[18]$,$k_1[19]$,$k_1[25]$,$k_1[26]$,$k_1[27]$,$k_1[28]$, $k_1[29]$,$k_1[30]$,$k_1[31]$,$k_1[32]$,$k_1[38]$,$k_1[39]$,$k_1[40]$,$k_1[41]$,$k_1[42]$, $k_1[43]$,$k_1[44]$, $k_1[45]$,$k_1[46]$,$k_1[51]$,$k_1[52]$,$k_1[53]$,$k_1[54]$,$k_1[55]$, $k_1[56]$,$k_1[57]$,$k_1[58]$,$k_1[59]$

7 Applications to round-reduced Initialization of Ketje

At 6 March 2017, the Keccak team announces the Ketje cryptanalysis prize to encourage the cryptanalysis. In [18], Li et al. present the conditional cube attacks on Ketje. Besides, they explore the resistance of Ketje against conditional cube attack according to different lengths of nonce. For Ketje Major, they study how short the length of nonce is, the conditional cube attacks work still. As a result, they point out that one could attack 7-round Ketje Major when its nonce is larger than 704 bits. While for Ketje Minor, it's necessary for adversaries to utilize (nearly) full length of nonce. In this section, we use MILP-aided cube-like-attack to explore the degrees of freedom for Ketje. We should point out that the MILP-aided cube-like-attack could work in the condition of smaller degrees of freedom. We present our attacks on Ketje as follows.

7.1 Attacks on round-reduced Initialization of Ketje Minor

Since we would like to explore how smaller the degree of freedom could be when the MILP-aided cube-like-attack works, we need to search for enough cube variables (64 for 7-round attack) and minimize the related key bits at the same time. The number of related key bits should be smaller than 128, on the other

Table 10: Parameters set for attack on 7-round KETJE MINOR

Cube Variables
$A[1][0][0]=A[1][3][0]=v_0$,$A[1][0][1]=A[1][3][1]=v_1$,$A[1][0][2]=A[1][3][2]=v_2$, $A[1][0][3]=A[1][3][3]=v_3$,$A[1][0][4]=A[1][3][4]=v_4$,$A[1][0][5]=A[1][3][5]=v_5$, $A[1][0][6]=A[1][3][6]=v_6$,$A[1][0][7]=A[1][3][7]=v_7$,$A[1][0][8]=A[1][3][8]=v_8$, $A[1][0][9]=A[1][3][9]=v_9$,$A[1][0][10]=A[1][3][10]=v_{10}$,$A[1][0][11]=A[1][3][11]=v_{11}$, $A[1][0][12]=A[1][3][12]=v_{12}$,$A[1][0][13]=A[1][3][13]=v_{13}$,$A[1][0][14]=A[1][3][14]=v_{14}$, $A[1][0][15]=A[1][3][15]=v_{15}$,$A[1][0][16]=A[1][3][16]=v_{16}$,$A[1][0][17]=A[1][3][17]=v_{17}$, $A[1][0][18]=A[1][3][18]=v_{18}$,$A[1][0][19]=A[1][3][19]=v_{19}$,$A[1][0][20]=A[1][3][20]=v_{20}$, $A[1][0][21]=A[1][3][21]=v_{21}$,$A[1][0][22]=A[1][3][22]=v_{22}$,$A[1][0][23]=A[1][3][23]=v_{23}$, $A[1][0][24]=A[1][3][24]=v_{24}$,$A[1][0][25]=A[1][3][25]=v_{25}$,$A[1][0][26]=A[1][3][26]=v_{26}$, $A[1][0][27]=A[1][3][27]=v_{27}$,$A[1][0][28]=A[1][3][28]=v_{28}$,$A[1][0][29]=A[1][3][29]=v_{29}$, $A[1][0][30]=A[1][3][30]=v_{30}$,$A[1][0][31]=A[1][3][31]=v_{31}$,$A[3][0][0]=A[3][2][0]=v_{32}$, $A[3][0][1]=A[3][2][1]=v_{33}$,$A[3][0][2]=A[3][2][2]=v_{34}$,$A[3][0][3]=A[3][2][3]=v_{35}$, $A[3][0][4]=A[3][2][4]=v_{36}$,$A[3][0][5]=A[3][2][5]=v_{37}$,$A[3][0][6]=A[3][2][6]=v_{38}$, $A[3][0][7]=A[3][2][7]=v_{39}$,$A[3][0][8]=A[3][2][8]=v_{40}$,$A[3][0][9]=A[3][2][9]=v_{41}$, $A[3][0][10]=A[3][2][10]=v_{42}$,$A[3][0][11]=A[3][2][11]=v_{43}$,$A[3][0][12]=A[3][2][12]=v_{44}$, $A[3][0][13]=A[3][2][13]=v_{45}$,$A[3][0][14]=A[3][2][14]=v_{46}$,$A[3][0][15]=A[3][2][15]=v_{47}$, $A[3][0][16]=A[3][2][16]=v_{48}$,$A[3][0][17]=A[3][2][17]=v_{49}$,$A[3][0][18]=A[3][2][18]=v_{50}$, $A[3][0][19]=A[3][2][19]=v_{51}$,$A[3][0][20]=A[3][2][20]=v_{52}$,$A[3][0][21]=A[3][2][21]=v_{53}$, $A[3][0][22]=A[3][2][22]=v_{54}$,$A[3][0][23]=A[3][2][23]=v_{55}$,$A[3][0][24]=A[3][2][24]=v_{56}$, $A[3][0][25]=A[3][2][25]=v_{57}$,$A[3][0][26]=A[3][2][26]=v_{58}$,$A[3][0][27]=A[3][2][27]=v_{59}$, $A[3][0][28]=A[3][2][28]=v_{60}$,$A[3][0][29]=A[3][2][29]=v_{61}$,$A[3][0][30]=A[3][2][30]=v_{62}$, $A[3][0][31]=A[3][2][31]=v_{63}$
Related Key Bits
$k_0[8]$,$k_0[9]$,$k_0[10]$,$k_0[11]$,$k_0[12]$,$k_0[13]$, $k_0[14]$,$k_0[15]$,$k_0[16]$,$k_0[17]$,$k_0[18]$,$k_0[19]$,$k_0[20]$, $k_0[21]$,$k_0[22]$,$k_0[23]$,$k_0[24]$,$k_0[25]$,$k_0[26]$,$k_0[27]$, $k_0[28]$,$k_0[29]$,$k_0[30]$,$k_0[31]$,$k_1[0]$,$k_1[1]$, $k_1[2]$,$k_1[3]$,$k_1[4]$,$k_1[5]$,$k_1[6]$,$k_1[7]$,$k_1[8]$, $k_1[9]$,$k_1[10]$,$k_1[11]$,$k_1[12]$,$k_1[13]$,$k_1[14]$,$k_1[15]$, $k_1[16]$,$k_1[17]$,$k_1[18]$,$k_1[19]$,$k_1[20]$,$k_1[21]$,$k_1[22]$, $k_1[23]$,$k_1[24]$,$k_1[25]$,$k_1[26]$,$k_1[27]$, $k_1[28]$, $k_1[29]$,$k_1[30]$,$k_1[31]$,$k_3[0]$,$k_3[1]$,$k_3[2]$,$k_3[3]$, $k_3[4]$,$k_3[5]$,$k_3[6]$,$k_3[7]$,$k_3[8]$,$k_3[9]$, $k_3[10]$, $k_3[11]$,$k_3[12]$,$k_3[13]$,$k_3[14]$,$k_3[15]$,$k_3[16]$,$k_3[17]$, $k_3[18]$,$k_3[19]$,$k_3[20]$,$k_3[21]$, $k_3[22]$,$k_3[23]$, $k_3[24]$,$k_3[25]$,$k_3[26]$,$k_3[27]$,$k_3[28]$,$k_3[29]$,$k_3[30]$, $k_3[31]$,$k_4[0]$,$k_4[1]$,$k_4[2]$, $k_4[3]$,$k_4[4]$,$k_4[5]$,$k_4[6]$,$k_4[7]$,
Guessed Key bits
$k_0[8]$,$k_0[9]$,$k_0[10]$,$k_0[11]$,$k_0[12]$,$k_0[13]$, $k_0[14]$,$k_0[15]$,$k_0[16]$,$k_0[17]$,$k_0[18]$,$k_0[19]$,$k_0[20]$, $k_0[21]$,$k_0[22]$,$k_0[23]$,$k_0[24]$,$k_0[25]$,$k_0[26]$,$k_0[27]$, $k_0[28]$,$k_0[29]$,$k_0[30]$,$k_0[31]$,$k_3[16]$, $k_3[17]$, $k_3[18]$,$k_3[19]$,$k_3[20]$,$k_3[21]$,$k_3[22]$,$k_3[23]$, $k_3[24]$,$k_3[25]$,$k_3[26]$,$k_3[27]$,$k_3[28]$, $k_3[29]$,$k_3[30]$, $k_3[31]$,$k_4[0]$,$k_4[1]$,$k_4[2]$,$k_4[3]$,$k_4[4]$,$k_4[5]$,$k_4[6]$,$k_4[7]$,

side, if the number of related key bits is smaller than 128, we would utilize the smallest length of nonce as far as we can. As a result, for Ketje Minor, when the length of nonce is reduced to 288-bit, we find 64-dimension *linear cubes* with 96 related key bits which are listed in Table 10 as well as the auxiliary variables and guessed key bits (listed in Table 11). With these variables, we could perform cube-attack-like cryptanalysis just as before. In preprocessing phase, we compute the cube sums over the 64 cube variables for each possible value of 48 guessed key bits and store them, while in the online phase, we compute the cube sums for each possible of 48 auxiliary variables, if the values of cube sums for these two phase equal, we regard the combination of 96-bit related key as the right key. The time complexity of this attack is $2^{48} \times 2^{64} \times 2 = 2^{113}$, and the memory complexity is 2^{48} for our 7-round key-recovery attack.

Table 11: Auxiliary Variables for attack on 7-round Ketje Minor

Auxiliary Variables
$A[1][3][0]=a_0+v_0$,$A[1][3][1]=a_1+v_1$,$A[1][3][2]=a_2+v_2$,$A[1][3][3]=a_3+v_3$,
$A[1][3][4]=a_4+v_4$,$A[1][3][5]=a_5+v_5$,$A[1][3][6]=a_6+v_6$,$A[1][3][7]=a_7+v_7$,
$A[1][3][8]=a_8+v_8$,$A[1][3][9]=a_9+v_9$,$A[1][3][10]=a_{10}+v_{10}$,$A[1][3][11]=a_{11}+v_{11}$,
$A[1][3][12]=a_{12}+v_{12}$,$A[1][3][13]=a_{13}+v_{13}$,$A[1][3][14]=a_{14}+v_{14}$,$A[1][3][15]=a_{15}+v_{15}$,
$A[1][3][16]=a_{16}+v_{16}$,$A[1][3][17]=a_{17}+v_{17}$,$A[1][3][18]=a_{18}+v_{18}$,$A[1][3][19]=a_{19}+v_{19}$,
$A[1][3][20]=a_{20}+v_{20}$,$A[1][3][21]=a_{21}+v_{21}$,$A[1][3][22]=a_{22}+v_{22}$,$A[1][3][23]=a_{23}+v_{23}$,
$A[1][3][24]=a_{24}+v_{24}$,$A[1][3][25]=a_{25}+v_{25}$,$A[1][3][26]=a_{26}+v_{26}$,$A[1][3][27]=a_{27}+v_{27}$,
$A[1][3][28]=a_{28}+v_{28}$,$A[1][3][29]=a_{29}+v_{29}$,$A[1][3][30]=a_{30}+v_{30}$,$A[1][3][31]=a_{31}+v_{31}$,
$A[3][0][0]=a_{32}+v_{32}$,$A[3][0][1]=a_{33}+v_{33}$,$A[3][0][2]=a_{34}+v_{34}$,$A[3][0][3]=a_{35}+v_{35}$,
$A[3][0][4]=a_{36}+v_{36}$,$A[3][0][5]=a_{37}+v_{37}$,$A[3][0][6]=a_{38}+v_{38}$,$A[3][0][7]=a_{39}+v_{39}$,
$A[3][0][8]=a_{40}+v_{40}$,$A[3][0][9]=a_{41}+v_{41}$,$A[3][0][10]=a_{42}+v_{42}$,$A[3][0][11]=a_{43}+v_{43}$,
$A[3][0][12]=a_{44}+v_{44}$,$A[3][0][13]=a_{45}+v_{45}$,$A[3][0][14]=a_{46}+v_{46}$,$A[3][0][15]=a_{47}+v_{47}$

7.2 Attacks on round-reduced Initialization of Ketje Major

For Ketje Major, we search for 64 cube variables with 58 related key bits when the length of nonce is reduced to 576 bits. We list the cube variables, related key bits, guessed key bits as well as the auxiliary variables in Table 12. We omit the attack procedure here but present the complexity. In preprocessing phase, we compute the cube sums over the 64 cube variables for each possible value of 29 guessed key bits and store them, while in the online phase, we compute the cube sums for each possible of 29 auxiliary variables, if the values of cube sums for these two phase equal, we regard the combination of 58-bit related key as the right key. The time complexity of this attack is $2^{29} \times 2^{64} \times 2 = 2^{94}$, and the memory complexity is 2^{29} for our 7-round key-recovery attack.

Table 12: Parameters set for attack on 7-round KETJE MAJOR

Cube Variables
$A[3][0][1]=A[3][3][1]=v_0$,$A[3][0][32]=A[3][3][32]=v_1$,$A[3][0][58]=A[3][3][58]=v_2$, $A[3][0][59]=A[3][3][59]=v_3$,$A[3][0][60]=A[3][3][60]=v_4$,$A[3][0][61]=A[3][3][61]=v_5$, $A[3][0][62]=A[3][3][62]=v_6$,$A[3][0][63]=A[3][3][63]=v_7$,$A[4][1][24]=A[4][4][24]=v_8$, $A[4][1][25]=A[4][4][25]=v_9$,$A[4][1][50]=A[4][4][50]=v_{10}$,$A[4][1][51]=A[4][4][51]=v_{11}$, $A[4][1][52]=A[4][4][52]=v_{12}$,$A[4][1][53]=A[4][4][53]=v_{13}$,$A[4][1][54]=A[4][4][54]=v_{14}$, $A[4][1][55]=A[4][4][55]=v_{15}$,$A[4][1][56]=A[4][4][56]=v_{16}$,$A[4][1][57]=A[4][4][57]=v_{17}$, $A[4][1][58]=A[4][4][58]=v_{18}$,$A[1][0][0]=A[1][3][0]=v_{19}$,$A[1][0][2]=A[1][3][2]=v_{20}$, $A[1][0][4]=A[1][3][4]=v_{21}$,$A[1][0][6]=A[1][3][6]=v_{22}$,$A[1][0][10]=A[1][3][10]=v_{23}$, $A[1][0][13]=A[1][3][13]=v_{24}$,$A[1][0][15]=A[1][3][15]=v_{25}$,$A[1][0][17]=A[1][3][17]=v_{26}$, $A[1][0][18]=A[1][3][18]=v_{27}$,$A[1][0][20]=A[1][3][20]=v_{28}$,$A[1][0][22]=A[1][3][22]=v_{29}$, $A[1][0][23]=A[1][3][23]=v_{30}$,$A[1][0][24]=A[1][3][24]=v_{31}$,$A[1][0][25]=A[1][3][25]=v_{32}$, $A[1][0][26]=A[1][3][26]=v_{33}$,$A[1][0][27]=A[1][3][27]=v_{34}$,$A[1][0][28]=A[1][3][28]=v_{35}$, $A[1][0][29]=A[1][3][29]=v_{36}$,$A[1][0][30]=A[1][3][30]=v_{37}$,$A[1][0][31]=A[1][3][31]=v_{38}$, $A[1][0][32]=A[1][3][32]=v_{39}$,$A[1][0][33]=A[1][3][33]=v_{40}$,$A[1][0][34]=A[1][3][34]=v_{41}$, $A[1][0][35]=A[1][3][35]=v_{42}$,$A[1][0][37]=A[1][3][37]=v_{43}$,$A[1][0][38]=A[1][3][38]=v_{44}$, $A[1][0][39]=A[1][3][39]=v_{45}$,$A[1][0][40]=A[1][3][40]=v_{46}$,$A[1][0][41]=A[1][3][41]=v_{47}$, $A[1][0][42]=A[1][3][42]=v_{48}$,$A[1][0][43]=A[1][3][43]=v_{49}$,$A[1][0][44]=A[1][3][44]=v_{50}$, $A[1][0][45]=A[1][3][45]=v_{51}$,$A[1][0][46]=A[1][3][46]=v_{52}$,$A[1][0][48]=A[1][3][48]=v_{53}$, $A[1][0][50]=A[1][3][50]=v_{54}$,$A[1][0][51]=A[1][3][51]=v_{55}$,$A[1][0][52]=A[1][3][52]=v_{56}$, $A[1][0][53]=A[1][3][53]=v_{57}$,$A[1][0][55]=A[1][3][55]=v_{58}$,$A[1][0][56]=A[1][3][56]=v_{59}$, $A[1][0][57]=A[1][3][57]=v_{60}$,$A[1][0][59]=A[1][3][59]=v_{61}$,$A[1][0][61]=A[1][3][61]=v_{62}$, $A[1][0][63]=A[1][3][63]=v_{63}$
Related Key Bits
$k_0[8]$,$k_0[38]$,$k_0[39]$,$k_1[0]$,$k_1[1]$,$k_1[2]$,$k_1[3]$, $k_1[4]$,$k_1[5]$,$k_1[6]$,$k_1[7]$,$k_1[8]$,$k_1[9]$,$k_1[10]$, $k_1[11]$,$k_1[12]$,$k_1[13]$,$k_1[14]$,$k_1[15]$,$k_1[16]$,$k_1[17]$, $k_1[18]$,$k_1[19]$,$k_1[20]$,$k_1[21]$,$k_1[22]$, $k_1[23]$,$k_1[24]$,$k_1[25]$,$k_1[26]$,$k_1[27]$,$k_1[28]$, $k_1[29]$,$k_1[30]$,$k_1[32]$,$k_1[33]$,$k_1[34]$,$k_1[35]$, $k_1[36]$,$k_1[37]$,$k_1[38]$,$k_1[39]$,$k_1[40]$,$k_1[41]$, $k_1[43]$,$k_1[45]$,$k_1[46]$,$k_1[47]$,$k_1[48]$,$k_1[50]$, $k_1[51]$,$k_1[52]$,$k_1[54]$,$k_1[56]$,$k_1[58]$,$k_1[59]$,$k_1[61]$,$k_1[63]$
Guessed Key bits
$k_0[8]$,$k_0[38]$,$k_0[39]$,$k_1[29]$,$k_1[30]$,$k_1[32]$,$k_1[33]$,$k_1[34]$,$k_1[35]$, $k_1[36]$,$k_1[37]$,$k_1[38]$, $k_1[39]$,$k_1[40]$,$k_1[41]$, $k_1[43]$,$k_1[45]$,$k_1[46]$,$k_1[47]$,$k_1[48]$,$k_1[50]$, $k_1[51]$,$k_1[52]$,$k_1[54]$, $k_1[56]$,$k_1[58]$,$k_1[59]$,$k_1[61]$,$k_1[63]$
Auxiliary Variables
$A[1][3][0]=a_0+v_{19}$,$A[1][3][1]=a_1$,$A[1][3][2]=a_2+v_{20}$,$A[1][3][3]=a_3$, $A[1][3][4]=a_4+v_{21}$,$A[1][3][5]=a_5$,$A[1][3][6]=a_6+v_{22}$,$A[1][3][7]=a_7$,$A[1][3][8]=a_8$, $A[1][3][9]=a_9$,$A[1][3][10]=a_{10}+v_{23}$,$A[1][3][11]=a_{11}$,$A[1][3][13]=a_{13}+v_{24}$, $A[1][3][12]=a_{12}$,$A[1][3][14]=a_{14}$,$A[1][3][15]=a_{15}+v_{25}$,$A[1][3][27]=a_{27}+v_{34}$, $A[1][3][17]=a_{17}+v_{26}$,$A[1][3][18]=a_{18}+v_{27}$,$A[1][3][19]=a_{19}$,$A[1][3][20]=a_{20}+v_{28}$, $A[1][3][21]=a_{21}$,$A[1][3][22]=a_{22}+v_{29}$,$A[1][3][23]=a_{23}+v_{30}$,$A[1][3][24]=a_{24}+v_{31}$, $A[1][3][25]=a_{25}+v_{32}$,$A[1][3][26]=a_{26}+v_{33}$,$A[1][3][16]=a_{16}$,$A[1][3][28]=a_{28}+v_{35}$

8 Conclusion

In this paper, we give a new idea to improve Dinur et al.'s cube-attack-like method. By using a new MILP tool, we find the optimal *linear-cube*s that are multiplied with minimum key bits for Keccak-MAC and Ketje. Then, we give the first 7-round key-recovery attack on Keccak-MAC-512. In Ketje Minor/Major, we also get better results in aspect of complexity or attacked rounds with smaller nonce.

When comparing with Huang et al.'s conditional cube attack, the advantage of the MILP-aided cube-attack-like cryptanalysis is that it has larger effective range. In variants with the same degrees of freedom, MILP-aided cube-attack-like cryptanalysis and conditional cube attack could achieve the same attacked rounds. In variants with relatively smaller degrees of freedom, MILP-aided cube-attack-like cryptanalysis could get better results than conditional cube attack.

Currently, the cryptanalysis progress of symmetric-key ciphers heavily depends on automated evaluation tools. Due to Keccak's robust design, its cryptanalysis is still hard and limited. In this paper, we provide a new MILP model to study Keccak. As we put the tedious cryptanalysis work to the MILP solver, the study of Keccak becomes easier.

References

1. http://www.gurobi.com/.
2. Guido Berton, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak sponge function family. `http://keccak.noekeon.org/`.
3. Guido Berton, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: Ketje v2, 2016. `http://competitions.cr.yp.to/round3/ketjev2.pdf`.
4. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In *SAC 2011*, pages 320–337, 2011.
5. Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. *IACR Cryptology ePrint Archive*, 2016:689, 2016.
6. Joan Daemen and Gilles Van Assche. Differential propagation analysis of Keccak. In *FSE*, volume 7549, pages 422–441. Springer, 2012.
7. Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on Keccak-224 and Keccak-256. In *FSE 2012*, pages 442–461. Springer, 2012.
8. Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In *FSE 2013*, pages 219–240. Springer, 2013.
9. 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 *EUROCRYPT 2015*, pages 733–761, 2015.
10. Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In *EUROCRYPT 2009*, pages 278–299, 2009.

11. Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Heuristic tool for linear cryptanalysis with applications to CAESAR candidates. In *ASIACRYPT 2015*, pages 490–509, 2015.
12. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In *CT-RSA 2015*, pages 371–387, 2015.
13. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1. 2. *Submission to the CAESAR Competition*, 2016.
14. Xiaoyang Dong, Zheng Li, Xiaoyun Wang, and Ling Qin. Cube-like attack on round-reduced initialization of Ketje Sr. *IACR Trans. Symmetric Cryptol.*, 2017:259–280, 2017.
15. Alexandre Duc, Jian Guo, Thomas Peyrin, and Lei Wei. Unaligned rebound attack: Application to Keccak. In *FSE 2012*, pages 402–421. Springer, 2012.
16. Jian Guo, Meicheng Liu, and Ling Song. Linear structures: Applications to cryptanalysis of round-reduced Keccak. In *ASIACRYPT 2016, Part I*, pages 249–274. Springer, 2016.
17. Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao. Conditional cube attack on reduced-round Keccak sponge function. In *EUROCRYPT 2017*, pages 259–288, 2017.
18. Zheng Li, Wenquan Bi, Xiaoyang Dong, and Xiaoyun Wang. Improved conditional cube attacks on Keccak keyed modes with milp method. Cryptology ePrint Archive, Report 2017/804, 2017. `http://eprint.iacr.org/2017/804`.
19. Silvia Mella, Joan Daemen, and Gilles Van Assche. New techniques for trail bounds and application to differential trails in Keccak. *IACR Trans. Symmetric Cryptol.*, 2017(1):329–357, 2017.
20. Paweł Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced Keccak. In *FSE2013*, pages 241–262. Springer, 2013.
21. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *Inscrypt 2011*, pages 57–76. Springer, 2011.
22. Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New collision attacks on round-reduced Keccak. In *EUROCRYPT 2017*, pages 216–243. Springer, 2017.
23. Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In *EUROCRYPT 2017, Part III*, pages 185–215, 2017.
24. Ling Song, Jian Guo, and Danping Shi. New milp modeling: Improved conditional cube attacks to Keccak-based constructions. Cryptology ePrint Archive, Report 2017/1030, 2017. `https://eprint.iacr.org/2017/1030.pdf`.
25. Ling Song, Guohong Liao, and Jian Guo. Non-full sbox linearization: Applications to collision attacks on round-reduced Keccak. In *CRYPTO 2017*, pages 428–451. Springer, 2017.
26. 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 *ASIACRYPT 2014*, pages 158–178. Springer, 2014.
27. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In *CRYPTO 2005*, pages 17–36. Springer, 2005.
28. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In *EUROCRYPT 2005*, pages 19–35. Springer, 2005.
29. Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying milp method to searching integral distinguishers based on division property for 6

lightweight block ciphers. In *ASIACRYPT 2016, Part I*, pages 648–678. Springer, 2016.
30. Chendong Ye and Tian Tian. New insights into divide-and-conquer attacks on the round-reduced Keccak-mac. Cryptology ePrint Archive, Report 2018/059, 2018. `https://eprint.iacr.org/2018/059.pdf`.