**Noname manuscript No.**
(will be inserted by the editor)

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

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

Received: date / Accepted: date

**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 manually, which leads to more key bits involved in the key-recovery attack, so the complexity is too high unnecessarily. In this paper, we introduce a new MILP model and make the cube attacks better on the Keccak keyed modes. Using this new MILP tool, we find the optimal cube variables for Keccak-MAC, Keyak and Ketje, which makes that a minimum number of key bits are involved in the key-recovery attack. For example, when the capacity is 256, we find a new 32-dimension cube for Keccak-MAC that involves only 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. We get the 8-round key-recovery attacks on Lake Keyak in nonce-respected setting. In addition, we get the best attacks on Ketje Major/Minor. For Ketje Major, when the length of nonce is 9 lanes, we could improve the best previous 6-round attack to 7-round. Our attacks do not threaten the full-round (12) Keyak/Ketje or the full-round (24) Keccak-MAC. When comparing with Huang et al.'s conditional cube attack, the MILP-aided cube-attack-like cryptanalysis has larger effective range and gets the best results on the Keccak keyed variants with relatively smaller number of degrees of freedom.

**Keywords** Keccak-MAC · Keyak · Ketje · MILP · Cube attack

**Mathematics Subject Classification (2000)** 94A60

## 1 Introduction

As a countermeasure of the collision attacks on MD5 and SHA-1 by Wang et al. [33,34], 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 [9,10,17,19,24] and evaluation tools [8,13,23] have been proposed, including the recent impressive collision attacks [26,28]. However, the cryptanalysis progress of Keccak is still limited.

At EUROCRYPT 2015, Dinur et al. [11] 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. [14] evaluated the security of Ascon [15] against the cube-attack-like cryptanalysis. Later, Dong et al. [16] applied the cube-like method to Ketje Sr [3]. At EUROCRYPT 2017, Huang et al. [20] introduced

[*] Both Xiaoyang Dong and Xiaoyun Wang are corresponding authors.
Xiaoyang Dong
E-mail: xiaoyangdong@tsinghua.edu.cn
Xiaoyun Wang
E-mail: xiaoyunwang@tsinghua.edu.cn
1. Key Laboratory of Cryptologic Technology and Information Security,
Ministry of Education, Shandong University, Jinan 250100, China
2. Institute for Advanced Study, Tsinghua University, Beijing 100084, China

the *conditional cube attack*, which takes advantage of the large state freedom of Keccak to find a so-called *conditional cube variable* that does not multiply with all the other *cube variables* (called *ordinary cube variables*) in the first round and second round of Keccak. Later, Li et al. [22] applied the *conditional cube attack* to Ascon.

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. [25] first applied MILP method to count active Sboxes of word-based block ciphers. 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. [30] extended this technique to search differential and linear trails. Since then, many more MILP-based automatic cryptanalysis tools have been introduced, such as searching integral distinguishers [36], differentials of ARX ciphers [18], (related-tweak/key) impossible differentials [7, 27, 38, 39].

At Asiacrypt 2017, Li et al. [21] 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. [29] 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. The tedious cryptanalytic works become much easier because of MILP model.

### 1.1 Our Contributions

In this paper, we find Dinur et al.'s [11] 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 as 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 [2], Dinur et al. found 32/64-dimension *linear-cube*s that are not multiplied with 64 key bits in Keccak-MAC-128 (with capacity 256).

In this paper, we propose a new 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, Keyak 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 number of degrees of freedom is small[1]. Hence, the conditional cube attack can only be applied to 6-round Keccak-MAC-512 and the attacks on Keyak in nonce-respected setting are still limited, etc. However, MILP-aided cube-attack-like cryptanalysis could not only attack the same number of rounds with conditional cube attack with the same number of degrees of freedom, but also get better results on the Keccak keyed variants with relatively smaller number of degrees 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[2] 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 multiplies 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 $2^{66}$. This key-recovery attack is experimentally verified and the verification programs could be found in the above public domain. We find a new 64-dimension

[1] In Keccak-MAC, the capacity is larger, the number of degrees of freedom is smaller; in Keyak and Ketje, the nonce or size of state is smaller, the number of degrees of freedom is smaller.

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

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

| Variant | Capacity | Type | Attacked Rounds | Time | Data | Memory | Source |
|---|---|---|---|---|---|---|---|
| Keccak-MAC-128 | 256 | Cube-attack-like | 6 | $2^{66}$ | $2^{32}$ | $2^{32}$ | [11] |
| | | Conditional Cube Attack | 6 | $2^{40}$ | $2^{32}$ | – | [20] |
| | | Balanced Divide-and-Conquer | 6 | $2^{45}$ | $2^{32}$ | $2^{13}$ | [37] |
| | | MILP-aided Cube-attack-like | 6 | $2^{42}$ | $2^{32}$ | $2^{9}$ | Sect. 6 |
| | | Cube-attack-like | 7 | $2^{97}$ | $2^{64}$ | $2^{32}$ | [11] |
| | | Conditional Cube Attack | 7 | $2^{72}$ | $2^{64}$ | – | [20] |
| | | Balanced Divide-and-Conquer | 7 | $2^{84}$ | $2^{64}$ | $2^{67}$ | [37] |
| | | MILP-aided Cube-attack-like | 7 | $2^{80}$ | $2^{64}$ | $2^{15}$ | Sect. 6 |
| Keccak-MAC-512 | 1024 | Conditional Cube Attack | 5 | $2^{24}$ | $2^{16}$ | – | [20] |
| | | Conditional Cube Attack | 6 | $2^{58.3}$ | $2^{32}$ | – | [21] |
| | | Conditional Cube Attack | 6 | $2^{41}$ | $2^{32}$ | – | [29] |
| | | MILP-aided Cube-attack-like | 7 | $2^{112.6}$ | $2^{64}$ | $2^{47}$ | Sect. 6 |

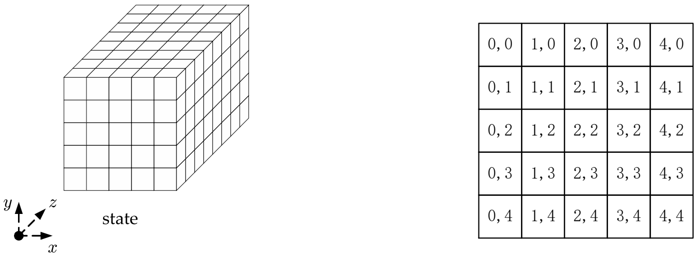
| Variant | Nonce | Type | Attacked Rounds | Time | Data | Memory | Source |
|---|---|---|---|---|---|---|---|
| Lake Keyak nonce respected | 128 | Cube Attack | 6 | $2^{37}$ | $2^{32}$ | – | [11] |
| | Default | Conditional Cube Attack | 8 | $2^{71.01}$ | $2^{64}$ | – | [29] |
| | Default | MILP-aided Cube-attack-like | 7 | $2^{42}$ | $2^{32}$ | $2^{9}$ | Sect .7.1 |
| | Default | MILP-aided Cube-attack-like | 8 | $2^{79.6}$ | $2^{64}$ | $2^{14}$ | Sect .7.1 |
| Ketje Minor | 654 | Cube-attack-like | 7 | $2^{96}$ | $2^{64}$ | $2^{32}$ | [16] |
| | 654 | Conditional Cube Attack | 7 | $2^{81}$ | $2^{64}$ | – | [21] |
| | 288 | MILP-aided Cube-attack-like | 7 | $2^{113}$ | $2^{64}$ | $2^{48}$ | Sect .8.1 |
| Ketje Major | 576 | Conditional Cube Attack | 6 | $2^{41}$ | $2^{64}$ | – | [21] |
| | 576 | MILP-aided Cube-attack-like | 7 | $2^{94}$ | $2^{64}$ | $2^{29}$ | Sect .8.2 |

*linear-cube* that only multiplies 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 $2^{17}$.

2. In Keccak sponge function, when the capacity reaches 1024, the number of degrees 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 gave cube-attack-like cryptanalysis when the capacity is smaller than 576. At EUROCRYPT 2017, Huang et al. gave the first 5-round key-recovery attack on Keccak-MAC-512 using conditional cube attack. Then at Asiacrypt 2017, Li et al. gave 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. Considering Keyak in nonce-respected setting, using our MILP tool, we improve Dinur et al.'s 6-round key-recovery attack on Lake Keyak to 8-round with the recommended nonce. 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 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.

### 1.2 Organization of the Paper

Sect. 2 gives some notations, and a brief description on Keccak-permutations, Keccak-MAC, Keyak 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 cryptanalysis results on Lake Keyak with the nonce-respected setting. Sect. 8 gives the applications to Ketje. Sect. 9 concludes this paper.



|  |  |  |  |  |
|---|---|---|---|---|
| 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

Table 2: Rotation constants $r[x, y]$ in Keccak $\rho$ 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 |

## 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 $\chi$ in the 1st round of Keccak-$p$, |
| $A$ | used in tables: for Keccak-MAC and Keyak, the initial state; for Ketje, the state after $\pi^{-1}$ 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 $i$th cube variable |
| $a_i$ | the $i$th 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 and Lake Keyak, $K = k_0\|\|k_1\|\|k_2$, $k_0$ 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 $j$th bit of $k_i$ |

### 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:

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

Keccak-$p[b]$ works on a state $A$ of size $b$, which can be represented as $5 \times 5$ $\frac{b}{25}$-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.

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

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. $\pi^{-1}$ is the inverse of $\pi$: $\pi^{-1} : A[x + 3y][x] = A[x][y]$.

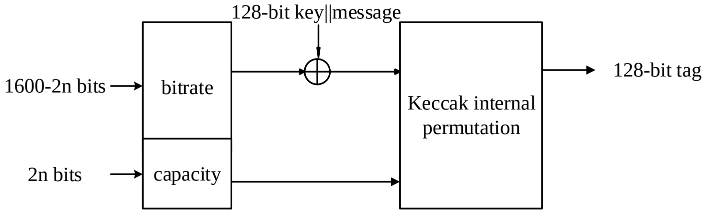


Fig. 2: Construction of Keccak-MAC-$n$

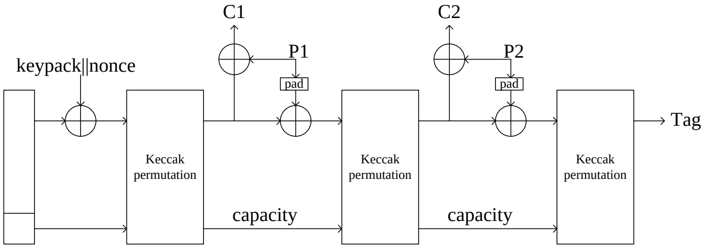


Fig. 3: Construction of Keyak on two blocks

Table 3: Five instances of Keyak

| Name | b | $\Pi$ | Main use case | 2nd use case |
|---|---|---|---|---|
| River Keyak | 800 | 1 | defense-in-depth | lightweight |
| Lake Keyak | 1600 | 1 | defense-in-depth | high performance |
| Sea Keyak | 1600 | 2 | defense-in-depth | high performance |
| Ocean Keyak | 1600 | 4 | defense-in-depth | high performance |
| Lunar Keyak | 1600 | 8 | defense-in-depth | high performance |

### 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, $n$ is half of the capacity length.

### 2.4 Keyak

Authenticated Encryption cipher Keyak [4] is one of the 16 candidates in the 3rd round CAESAR competition, whose mode is based on Motorist mode [4], which is sponge-based and supports one or more duplex instances operating in parallel.

In Keyak, five instances are proposed. For all instances, the round number of Keccak-$p[b]$ is $n_r = 12$, the capacity $c = 256$ and the tag length $\tau = 128$. The primary recommendation is the Lake Keyak. As shown in Fig. 3, 128-bit key is encoded by a key pack:

$$keypack(key, l) = enc_8(l)||key||pad10*[8l-8](|key|)$$

In Lake Keyak, the key pack $l$ is 40 bytes and 150-byte nonce is recommended. According to the specification of Keyak, in order to assure confidentiality of data, a user must respect the nonce requirement. Namely, a nonce cannot be reused, otherwise, confidentiality is not guaranteed. However, for authenticity and integrity of data, a variable nonce is not required. Readers can refer to [4] for more details.

### 2.5 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 [3], which is based on MonkeyDuplex [5]. 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.

In Ketje v2, four concrete instances are proposed, shown in Table 4, $n_{start} = 12$, $n_{step} = 1$ 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 4: Four Instances in Ketje v2

| Name | $f$ | $\rho$ | 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 [12], 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, $v_0, ..., v_{m-1}$ are the public variables (e.g. IV or nonce bits).

**Theorem 1** *( [12] )*

$$f(k_0, ..., k_{n-1}, v_0, ..., v_{m-1}) = t \cdot P + Q(k_0, ..., k_{n-1}, v_0, ..., v_{m-1}) \tag{1}$$

*$t$ is called maxterm and is a product of certain public variables, for example $(v_0, ..., v_{s-1})$, $1 \le s \le m$, denoted as cube $C_t$. None of the monomials in $Q$ is divisible by $t$. $P$ is called superpoly, which does not contain any variables of $C_t$. Then the sum of $f$ over all values of the cube $C_t$ (cube sum) is*

$$\sum_{v'=(v_0,...,v_{s-1}) \in C_t} f(k_0, ..., k_{n-1}, v', v_s, ..., v_{m-1}) = P \tag{2}$$

*where $C_t$ contains all binary vectors of the length $s$, $v_s, ..., v_{m-1}$ are fixed to constant.*

The basic idea is to find enough $t$ whose $P$ is linear and not a constant. This enables the key recovery through solving a system of linear equations. Recently, the cube-attack technique was improved a lot by Todo et al. [32] and Wang et al. [35] using division property [31].

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

At EUROCRYPT 2015, Dinur et al. [11] 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 found 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 $\theta$, $\rho$ and $\pi$, 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 $A[1][0]$.

In addition, Dinur et al. introduced 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,

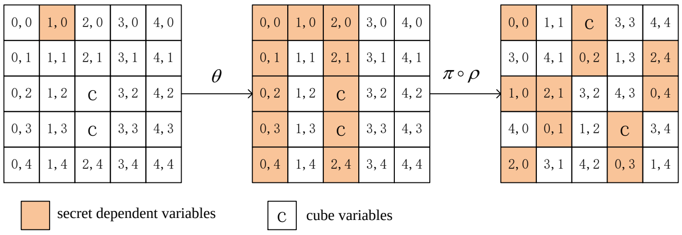


Fig. 4: Dinur et al.'s Work

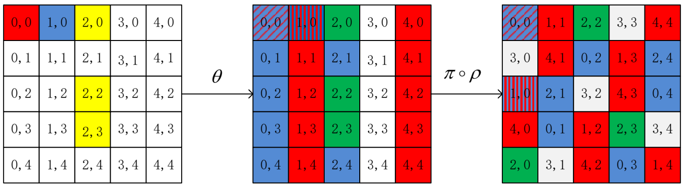


Fig. 5: The diffusion of key bits and cube variables in one round

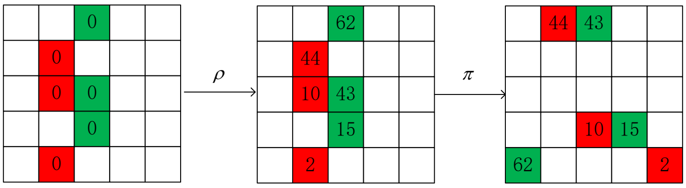


Fig. 6: The offset of key bits and cube variables in one round

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 [11].

## 4 An Improvement of Dinur et al.'s Idea

In Dinur et al.'s divide-and-conquer strategy, the cube was chosen manually and not optimal. Recently, Ye et al. [37] introduced a method to choose the cube variables more precisely, then the number of secret key bits, which multiply with the cube variables in the first round, will decrease and the complexity would reduce as well, which leads to longer-round attacks sometimes. 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*), we describe this idea as follows.

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 cube variables as follows:

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

Firstly, we explore that how many lanes with key bits would multiply with the cube variables after the first round. After $\theta$ operation, these lanes in red are diffused by $k_0$ as shown in Fig. 5 and the lanes in blue are diffused by $k_1$, while the cube variables in green are just added by the key $k_1$, not multiplied with them. After $\rho$ and $\pi$ operation, we could see that only three lanes diffused by $k_0$ will multiply with cube variables (in green), other lanes especially diffused by $k_1$ would do not multiply with cube variables in the first round.

Secondly, we explore how many bits in $k_0$ (only in 3 lanes) would multiply with these cube variables. As the $\rho$ operation rotates a different offset (Table 2) for different lanes, we use the number in each lane to denote the rotated offset compared with the initial state, as shown in Fig. 6. In the $\chi$ operation, the cube

Table 5: The key bits multiplied with the 32-dimension cube variable

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

variables in $A[2][0]$ (in the initial state $A[2][0]$) would multiply with the key bits $k_0[3 \cdot i + 62 - 2]$ (mod 64) for $0 \leq i \leq 15$, the cube variables in $A[2][2]$ would multiply with the key bits $k_0[3 \cdot i + 43 - 44]$ (mod 64) for $0 \leq i \leq 15$, and the cube variables in $A[2][3]$ would multiply with the key bits $k_0[3 \cdot i + 15 - 10]$ (mod 64) for $0 \leq i \leq 15$, we list the key bits they multiplied in Table 5 for each lane. As we can see that in different lines in Table 5, each related key bit emerges two or three times corresponding to different lane, which means that only a few key bits are involved in the cube attacks after diffusion if we choose the cube variables more precisely.

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 manually 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 a new MILP method to solve the above optimization problem and then improve cube-attack-like cryptanalysis on Keccak keyed modes especially when the number of degrees of freedom is relatively small.

## 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 with each other 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 choose one of them as cube variable at most.

In order to control the diffusion of the cube variables, we make use of the *CP-like-kernel* which was formalized by Guo et al. [19] and studied by the related work [21]. We keep the sum of the variables within the same column is constant (usually zero) which makes the following $\theta$ 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 $[x][z]$ contains 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 free bits 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 free bits.

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:

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

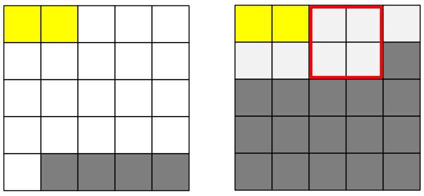


Fig. 7: The Initial State of Keccak-MAC-128(left) and Keccak-MAC-512(right)

Since we would like to get the minimum number 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

In this section, we apply our MILP tool to round-reduced Keccak-MAC. In order to get explicit comparison with previous the cube-attack-like on Keccak-MAC and to verify our key-recovery attack experimentally, we present the application when the capacity is 256 for Keccak-MAC at first, i.e. Keccak-MAC-128, our attack is slightly inferior to Huang et al.'s conditional cube attack, but much better than previous cube-like attack. And then, we give our main result, the application when the capacity is 1024, i.e. Keccak-MAC-512, our attack is the first 7-round key-recovery attack on Keccak-MAC-512.

### 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 shown in Fig. 7, 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.

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

$$Min \sum_{y=0,x\in\{0,1\},z\in\{0,1\dots63\}} A[x][y][z].$$

With the help of Gurobi [1], the objective function is optimized under all the constraints in Sect. 5, the minimum number of related key bits is 18 for 32-dimension *linear cube*s and 30 for 64-dimension *linear cube*s. The cube variables and related bits are listed in Table 6 and Table 7 in Appendix A.

**Attack on 6-round Keccak-MAC-128**. The attack includes preprocessing phase and online phase. The related key bits in the Table 6 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 6 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 they do not multiplied with cube variables so that the cube sums do not depend on these key bits. We choose the auxiliary variables precisely 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 [3]. After recovering these key bits, we just shift the positions of all the cube variables along the $z$-direction and there would be another set of key bits involved in the key-recovery attack because of the translation invariance in the direction of the z axis of Keccak. We present the attack procedure as follows:

**Preprocessing Phase.**

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

- 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 6, calculate the $2^{32}$ cube sums after 6 rounds for all the output bits according to the 32-dimension cube variables in Table 6. 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 the time complexity is $2^9 \times 2^{32} = 2^{41}$ 6-round Keccak-MAC, and the memory complexity is $2^9$ 128-bit word.

**Online Phase.**

- For each possible value of 9-bit auxiliary variables listed in Table 6, 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 6.

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 $2^{66}$.

**Attack on 7-round Keccak-MAC-128**. We find 30 related key bits with 64-dimension cube variables which are listed in Table 7 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 $2^{15}$ 128-bit word.

### 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 shown in Fig. 7, 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.

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

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

**Attack on 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 $2^{26}$ 128-bit word.

**Attack on 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 $2^{47}$ 128-bit word.

## 7 Attacks on round-reduced Keyak

Authenticated Encryption cipher Keyak is one of the 16 candidates in the 3rd round CAESAR competition, the cryptanalytic results before such as [6, 20] aim at its authenticity only, in other word, nonce is reused. While if confidentiality and authenticity of data are both required, a nonce can not be reused [4]. In this section, we present 7/8-round key recovery attacks for Lake Keyak (1600-bit state) with nonce-respected setting, we use the recommended length of nonce 150 bytes for Lake Keyak, according to the design documents [4]. Our result on Lake Keyak is the first 8-round key-recovery attack in the nonce-respected setting as we know, but nearly at the same time, Song et al. [29] provide another result with conditional cube attack, which is sightly better than ours.

### 7.1 Attacks on round-reduced Lake Keyak

For Lake Keyak in the nonce-respected setting, we get 18 related key bits with 32-dimension *linear cube* and 29 related key bits with 64-dimension *linear cube*. The related key bits, guessed key bits, auxiliary variables and cube variables are listed in Table 10 and Table 11 in Appendix A for 7-round/8-round key-recovery attack respectively. We present the 7/8-round key-recovery attack for Lake Keyak with time complexity $2^{42}$ and $2^{79.6}$ respectively as follows.

**Attack on 7-round Lake Keyak**. We use the 32-dimension cube variables listed in Table 10, as well as the related key bits, guessed key bits and auxiliary variables.

**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 10, calculate the output bits after 7 rounds (with 256 capacity, there are 1344-bit output). Then compute the first 10-lane output backward through $\iota$ and $\chi$ operations (as the state $S_{6.5}$ has the same algebraic degree with $S_7$) and compute the cube sums for the $2^{32}$ messages according to the 32-dimension cube variables listed in Table 10. Store the 640-bit cube sums in a sorted list $L$ with the value of 9-bit guessed key.

**Online Phase.**

- For each possible value of 9-bit auxiliary variables list in Table 10, request the 7-round River Keyak outputs of $2^{32}$ messages that make up the 32 cube variables, then compute the first 10-lane output backward through $\iota$ and $\chi$ operations.
- Calculate the cube sums for the first 10-lane 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 10.

We need compute the 32-dimension cube sums for $2^9$ values of guessed key bits in the preprocessing phase, the time complexity is $2^9 \times 2^{32} = 2^{41}$ and memory complexity is $2^9$ 640-bit word. In the online phase, we need compute the cube sums for each possible value of 9 auxiliary variables, the time complexity is also $2^{41}$. Thus, the time complexity is $2^{42}$ and memory complexity is $2^9$ for the whole attack.

**Attack on 8-round Lake Keyak**. The attack procedure is very similar to the 7-round attack, so we only introduce the complexity of our attacks in the following. We need compute the cube sums for $2^{14}$ values of 14 guessed key bits in the preprocessing phase, the time complexity is $2^{14} \times 2^{64} = 2^{78}$ and memory complexity is $2^{14}$ 640-bit word. In the online phase, we need compute the cube sums for each possible value of 15 auxiliary variables, the time complexity is $2^{15} \times 2^{64} = 2^{79}$. Thus, the time complexity is $2^{78} + 2^{79} = 2^{79.6}$ and memory complexity is $2^{14}$.

## 8 Applications to round-reduced Initialization of Ketje

At 6 March 2017, the Keccak team announces the Ketje cryptanalysis prize to encourage the cryptanalysis. In [21], 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 search the possible cube variables in the instances with different lengths of nonce, and study the borderline length of nonce to provide enough cube variables for conditional cube attack. 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 number of degrees of freedom for Ketje. We should point out that the MILP-aided cube-like-attack could work better when the number of degrees of freedom is smaller. We present our attacks on Ketje as follows.

### 8.1 Attacks on round-reduced Initialization of Ketje Minor

Since we would like to explore how smaller the number of degrees 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 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 12 in Appendix A as well as the auxiliary variables and guessed key bits. 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 cube sums in these two phase are 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.

### 8.2 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 13 in Appendix A. 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.

## 9 Conclusion

In this paper, we give a new MILP-based method to improve Dinur et al.'s cube-attack-like method. We find the optimal *linear-cube*s that are multiplied with minimum number of key bits for Keccak-MAC, Lake Keyak and Ketje. Then, we give the first 7-round key-recovery attack on Keccak-MAC-512. In Lake Keyak, our attack could work in the nonce-respected setting, while not as good as conditional cube attack. In Ketje Minor/Major, we get better results than before in aspect of complexity or attacked rounds with smaller length of 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 number of degrees of freedom, MILP-aided cube-attack-like cryptanalysis and conditional cube attack could attack the same rounds. In variants with relatively smaller number of 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.

## Acknowledgements

This work is supported by National Key Research and Development Program of China (No.2017YFA0303903), National Cryptography Development Fund (No.MMJJ20170121), National Natural Science Foundation of China (No.61672019), Fundamental Research Funds of Shandong University (No.2016JC029), China Post-doctoral Science Foundation (No.2017M620807).

## References

1. Http://www.gurobi.com/
2. Berton, G., Daemen, J., Peeters, M., Assche, G.V.: The Keccak sponge function family, http://keccak.noekeon.org/
3. Berton, G., Daemen, J., Peeters, M., Assche, G.V., Keer, R.V.: CAESAR submission: Ketje v2 (2016), http://competitions.cr.yp.to/round3/ketjev2.pdf
4. Berton, G., Daemen, J., Peeters, M., Assche, G.V., Keer, R.V.: CAESAR submission: Keyak v2 (2016), http://competitions.cr.yp.to/round3/keyakv22.pdf
5. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the sponge: Singlepass authenticated encryption and other applications. In: SAC 2011. pp. 320-337 (2011)

6. Bi, W., Li, Z., Dong, X., Li, L., Wang, X.: Conditional cube attack on round reduced River Keyak. Designs, Codes and Cryptography. vol 86, pp. 1295-1310 (2018)
7. Cui, T., Jia, K., Fu, K., Chen, S., Wang, M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptology ePrint Archive, 2016/689 (2016)
8. Daemen, J., Van Assche, G.: Differential propagation analysis of Keccak. In: FSE 2012, vol. 7549, pp. 422-441. Springer (2012)
9. Dinur, I., Dunkelman, O., Shamir, A.: New attacks on Keccak-224 and Keccak-256. In: FSE 2012. pp. 442-461. Springer (2012)
10. Dinur, I., Dunkelman, O., Shamir, A.: Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In: FSE 2013. pp. 219-240. Springer (2013)
11. Dinur, I., Morawiecki, P., Pieprzyk, J., Srebrny, M., Straus, M.: Cube attacks and cube-attack-like cryptanalysis on the round-reduced Keccak sponge function. In: EUROCRYPT 2015. pp. 733-761 (2015)
12. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: EUROCRYPT 2009. pp. 278-299 (2009)
13. Dobraunig, C., Eichlseder, M., Mendel, F.: Heuristic tool for linear cryptanalysis with applications to CAESAR candidates. In: ASIACRYPT 2015. pp. 490-509 (2015)
14. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Cryptanalysis of Ascon. In: CT-RSA 2015. pp. 371-387 (2015)
15. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1. 2. Submission to the CAESAR Competition (2016)
16. Dong, X., Li, Z., Wang, X., Qin, L.: Cube-like attack on round-reduced initialization of Ketje Sr. IACR Trans. Symmetric Cryptol. 2017, pp. 259-280 (2017)
17. Duc, A., Guo, J., Peyrin, T., Wei, L.: Unaligned rebound attack: Application to Keccak. In: FSE 2012. pp. 402-421. Springer (2012)
18. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: MILP-Based automatic search algorithms for differential and linear trails for Speck. In: FSE 2016. pp. 268–288. Springer (2016)
19. Guo, J., Liu, M., Song, L.: Linear structures: Applications to cryptanalysis of round-reduced Keccak. In: ASIACRYPT 2016, Part I. pp. 249-274. Springer (2016)
20. Huang, S., Wang, X., Xu, G., Wang, M., Zhao, J.: Conditional cube attack on reduced-round Keccak sponge function. In: EUROCRYPT 2017. pp. 259-288 (2017)
21. Li, Z., Bi, W., Dong, X., Wang, X.: Improved conditional cube attacks on Keccak keyed modes with MILP method. In: ASIACRYPT 2017, Part I. pp. 99-127. Springer (2017)
22. Li, Z., Dong, X., Wang, X.: Conditional cube attack on round-reduced ASCON. IACR Trans. Symmetric Cryptol. 2017, pp. 175-202 (2017)
23. Mella, S., Daemen, J., Assche, G.V.: New techniques for trail bounds and application to differential trails in Keccak. IACR Trans. Symmetric Cryptol. 2017(1), pp. 329-357 (2017), http://tosc.iacr.org/index.php/ToSC/article/view/597
24. Morawiecki, P., Pieprzyk, J., Srebrny, M.: Rotational cryptanalysis of roundreduced Keccak. In: FSE2013. pp. 241-262. Springer (2013)
25. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Inscrypt 2011. pp. 57-76. Springer (2011)
26. Qiao, K., Song, L., Liu, M., Guo, J.: New collision attacks on round-reduced Keccak. In: EUROCRYPT 2017. pp. 216-243. Springer (2017)
27. Sasaki, Y., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: EUROCRYPT 2017, Part III. pp. 185-215 (2017)
28. Song, L., Liao, G., Guo, J.: Non-full sbox linearization: Applications to collision attacks on round-reduced Keccak. In: CRYPTO 2017. pp. 428-451. Springer (2017)
29. Song, L., Guo, J., Shi, D.: 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
30. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: ASIACRYPT 2014. pp. 158-178. Springer (2014)
31. Todo, Y.: Structural evaluation by generalized integral property. In: EUROCRYPT 2015, Part I. pp. 287-314. Springer (2015)
32. Todo, Y., Isobe, T., Hao, Y., Meier, W.: Cube attacks on non-blackbox polynomials based on Division Property. In: CRYPTO 2017, Part III. pp. 250-279. Springer (2017)
33. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: CRYPTO 2005. pp. 17-36. Springer (2005)
34. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: EUROCRYPT 2005. pp. 19-35. Springer (2005)
35. Wang, Q., Hao, Y., Todo, Y., Li, C., Isobe, T., Meier, W.: Improved division property based cube attacks exploiting algebraic properties of superpoly. To appear in CRYPTO 2018. https://eprint.iacr.org/2017/1063
36. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: ASIACRYPT 2016, Part I. pp. 648-678. Springer (2016)
37. Ye, C., Tian, T.: 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
38. Zong, R., Dong, X., Wang, X.: Related-tweakey impossible differential attack on reduced-round Deoxys-BC-256. Cryptology ePrint Archive, Report 2018/680, 2018. https://eprint.iacr.org/2018/680
39. Zong, R., Dong, X., Wang, X.: MILP-aided related-tweak/key impossible differential attack and its applications to QARMA, Joltik-BC. Cryptology ePrint Archive, Report 2018/142, 2018. https://eprint.iacr.org/2018/142

## A Parameters set for attack

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

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

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

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

Table 8: Parameters set for attack on 6-round Keccak-MAC-512

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

Table 9: 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** |
| $A[0][1][0]=a_0$,$A[0][1][1]=a_1$,$A[0][1][2]=a_2$,$A[0][1][3]=a_3$,$A[0][1][4]=a_4$,$A[0][1][5]=a_5$, $A[0][1][6]=a_6$,$A[0][1][7]=a_7$, $A[0][1][8]=a_8$,$A[0][1][9]=a_9$,$A[0][1][10]=a_{10}$, $A[0][1][13]=a_{11}$,$A[0][1][14]=a_{12}$,$A[0][1][15]=a_{13}$,$A[0][1][16]=a_{14}$, $A[0][1][17]=a_{15}$, $A[0][1][18]=a_{16}$,$A[0][1][19]=a_{17}$,$A[0][1][20]=a_{18}$,$A[0][1][21]=a_{19}$,$A[0][1][22]=a_{20}$, $A[0][1][25]=a_{21}$, $A[0][1][26]=a_{22}$,$A[0][1][27]=a_{23}$,$A[0][1][28]=a_{24}$,$A[0][1][29]=a_{25}$, $A[0][1][30]=a_{26}$,$A[0][1][31]=a_{27}$,$A[0][1][32]=a_{28}$, $A[0][1][33]=a_{29}$,$A[0][1][34]=a_{30}$, $A[0][1][37]=a_{31}$,$A[0][1][38]=a_{32}$,$A[0][1][39]=a_{33}$,$A[0][1][40]=a_{34}$,$A[0][1][41]=a_{35}$, $A[0][1][42]=a_{36}$,$A[0][1][43]=a_{37}$,$A[0][1][44]=a_{38}$,$A[0][1][45]=a_{39}$,$A[0][1][46]=a_{40}$, $A[0][1][49]=a_{41}$,$A[0][1][50]=a_{42}$, $A[0][1][51]=a_{43}$,$A[0][1][52]=a_{44}$,$A[0][1][53]=a_{45}$, $A[0][1][54]=a_{46}$,$A[0][1][55]=a_{47}$ |
| **Guessed Key Bits** |
| $k_0[56]$,$k_0[57]$,$k_0[58]$,$k_0[59]$,$k_0[61]$, $k_0[62]$,$k_1[0]$,$k_1[1]$,$k_1[2]$,$k_1[3]$,$k_1[4]$,$k_1[5]$,$k_1[6]$,$k_1[7]$,$k_1[13]$,$k_1[14]$, $k_1[15]$,$k_1[16]$, $k_1[17]$,$k_1[18]$,$k_1[19]$,$k_1[25]$,$k_1[26]$,$k_1[27]$,$k_1[28]$, $k_1[29]$,$k_1[30]$,$k_1[31]$,$k_1[32]$,$k_1[38]$,$k_1[39]$,$k_1[40]$,$k_1[41]$,$k_1[42]$, $k_1[43]$, $k_1[44]$,$k_1[45]$,$k_1[46]$,$k_1[51]$,$k_1[52]$,$k_1[53]$,$k_1[54]$,$k_1[55]$, $k_1[56]$,$k_1[57]$,$k_1[58]$,$k_1[59]$ |

Table 10: Parameters set for attack on 7-round Lake Keyak

| **Cube Variables** |
|---|
| $A[0][1][2]{=}A[0][3][2]{=}v_0$,$A[0][1][6]{=}v_1$,$A[0][2][6]{=}v_2$,$A[0][3][6]{=}v_1 + v_2$, $A[0][1][12]{=}A[0][2][12]{=}v_3$, $A[0][1][13]{=}A[0][3][13]{=}v_4$,$A[0][1][19]{=}v_5$,$A[0][2][19]{=}v_6$, $A[0][3][19]{=}v_5 + v6$,$A[0][2][57]{=}A[0][3][57]{=}v_7$, $A[0][1][59]{=}A[0][2][59]{=}v_8$, $A[0][2][63]{=}A[0][3][63]{=}v_9$,$A[2][2][1]{=}A[2][4][1]{=}v_{10}$,$A[2][3][4]{=}A[2][4][4]{=}v_{11}$, $A[2][3][5]{=}A[2][4][5]{=}v_{12}$,$A[2][2][10]{=}v_{13}$,$A[2][3][10]{=}v_{14}$,$A[2][4][10]{=}v_{13} + v_{14}$, $A[2][2][11]{=}A[2][4][11]{=}v_{15}$, $A[2][3][12]{=}A[2][4][12]{=}v_{16}$,$A[2][2][15]{=}A[2][3][15]{=}v_{17}$, $A[2][2][16]{=}v_{18}$,$A[2][3][16]{=}v_{19}$,$A[2][4][16]{=}v_{18} + v_{19}$, $A[2][2][18]{=}A[2][4][18]{=}v_{20}$, $A[2][2][21]{=}A[2][3][21]{=}v_{21}$,$A[2][2][22]{=}v_{22}$,$A[2][3][22]{=}v_{23}$,$A[2][4][22]{=}v_{22} + v_{23}$, $A[2][3][23]{=}A[2][4][23]{=}v_{24}$,$A[2][2][27]{=}A[2][3][27]{=}v_{25}$,$A[2][2][29]{=}v_{26}$, $A[2][3][29]{=}v_{27}$,$A[2][4][29]{=}v_{26} + v_{27}$, $A[2][2][33]{=}A[2][3][33]{=}v_{28}$,$A[2][2][40]{=}A[2][3][40]{=}v_{29}$, $A[2][2][53]{=}A[2][3][53]{=}v_{30}$,$A[2][2][59]{=}A[2][3][59]{=}v_{31}$ |
| **Related Key Bits** |
| $k_0[9]$,$k_0[10]$,$k_0[14]$,$k_0[15]$,$k_0[17]$,$k_0[20]$,$k_0[21]$, $k_0[26]$,$k_0[27]$,$k_0[28]$,$k_0[32]$,$k_0[34]$,$k_0[38]$, $k_0[39]$,$k_0[41]$,$k_0[45]$,$k_0[52]$, $k_0[58]$, |
| **Guessed Key bits** |
| $k_0[28]$,$k_0[32]$,$k_0[34]$,$k_0[38]$, $k_0[39]$,$k_0[41]$,$k_0[45]$,$k_0[52]$,$k_0[58]$ |
| **Auxiliary Variables** |
| $A[0][1][9]{=}a_0$,$A[0][1][10]{=}a_1$,$A[0][1][14]{=}a_2$,$A[0][1][15]{=}a_3$, $A[0][1][17]{=}a_4$, $A[0][1][20]{=}a_5$,$A[0][1][21]{=}a_6$, $A[0][1][26]{=}a_7$,$A[0][1][27]{=}a_8$, |

Table 11: Parameters set for attack on 8-round Lake Keyak

| **Cube Variables** |
|---|
| $A[0][1][4]{=}v_0$,$A[0][2][4]{=}v_1$,$A[0][3][4]{=}v_0 + v_1$,$A[0][1][11]{=}v_2$,$A[0][2][11]{=}v_3$, $A[0][3][11]{=}v_2 + v_3$,$A[0][1][18]{=}v_4$, $A[0][2][18]{=}v_5$,$A[0][3][18]{=}v_4 + v_5$, $A[0][2][22]{=}A[0][3][22]{=}v_6$,$A[0][1][25]{=}A[0][3][25]{=}v_7$,$A[0][2][28]{=}A[0][3][28]{=}v_8$, $A[0][1][29]{=}A[0][3][29]{=}v_9$,$A[0][1][35]{=}v_{10}$,$A[0][2][35]{=}v_{11}$,$A[0][3][35]{=}v_{10} + v_{11}$, $A[0][1][42]{=}A[0][3][42]{=}v_{12}$, $A[0][1][47]{=}A[0][2][47]{=}v_{13}$,$A[0][1][48]{=}A[0][2][48]{=}v_{14}$, $A[0][1][54]{=}v_{15}$,$A[0][2][54]{=}v_{16}$,$A[0][3][54]{=}v_{15} + v_{16}$, $A[0][1][55]{=}A[0][3][55]{=}v_{17}$, $A[0][1][58]{=}A[0][2][58]{=}v_{18}$,$A[0][1][61]{=}v_{19}$,$A[0][2][61]{=}v_{20}$,$A[0][3][61]{=}v_{19} + v_{20}$, $A[0][2][62]{=}A[0][3][62]{=}v_{21}$,$A[2][2][0]{=}v_{22}$,$A[2][3][0]{=}v_{23}$,$A[2][4][0]{=}v_{22} + v_{23}$, $A[2][2][1]{=}v_{24}$,$A[2][3][1]{=}v_{25}$, $A[2][4][1]{=}v_{24} + v_{25}$,$A[2][2][4]{=}v_{26}$,$A[2][3][4]{=}v_{27}$, $A[2][4][4]{=}v_{26} + v_{27}$,$A[2][2][5]{=}A[2][3][5]{=}v_{28}$,$A[2][2][7]{=}v_{29}$, $A[2][3][7]{=}v_{30}$, $A[2][4][7]{=}v_{29} + v_{30}$,$A[2][2][10]{=}A[2][4][10]{=}v_{31}$,$A[2][2][11]{=}A[2][3][11]{=}v_{32}$, $A[2][2][13]{=}A[2][4][13]{=}v_{33}$, $A[2][3][14]{=}A[2][4][14]{=}v_{34}$,$A[2][2][17]{=}A[2][4][17]{=}v_{35}$, $A[2][2][18]{=}A[2][3][18]{=}v_{36}$,$A[2][2][20]{=}A[2][4][20]{=}v_{37}$, $A[2][2][21]{=}v_{38}$, $A[2][3][21]{=}v_{39}$,$A[2][4][21]{=}v_{38} + v_{39}$,$A[2][2][24]{=}v_{40}$,$A[2][3][24]{=}v_{41}$, $A[2][4][24]{=}v_{40} + v_{41}$, $A[2][2][27]{=}A[2][4][27]{=}v_{42}$,$A[2][3][28]{=}A[2][4][28]{=}v_{43}$, $A[2][2][31]{=}A[2][3][31]{=}v_{44}$,$A[2][2][34]{=}A[2][4][34]{=}v_{45}$, $A[2][3][35]{=}A[2][4][35]{=}v_{46}$, $A[2][2][38]{=}A[2][3][38]{=}v_{47}$,$A[2][3][39]{=}A[2][4][39]{=}v_{48}$,$A[2][2][41]{=}A[2][4][41]{=}v_{49}$, $A[2][2][45]{=}v_{50}$,$A[2][3][45]{=}v_{51}$,$A[2][4][45]{=}v_{50} + v_{51}$,$A[2][3][46]{=}A[2][4][46]{=}v_{52}$, $A[2][2][51]{=}v_{53}$,$A[2][3][51]{=}v_{54}$, $A[2][4][51]{=}v_{53} + v_{54}$,$A[2][2][52]{=}v_{55}$, $A[2][4][52]{=}v_{55} + v_{56}$,$A[2][2][56]{=}A[2][3][56]{=}v_{57}$,$A[2][2][57]{=}v_{58}$, $A[2][3][57]{=}v_{59}$, $A[2][3][52]{=}v_{56}$,$A[2][4][57]{=}v_{58} + v_{59}$,$A[2][2][58]{=}v_{60}$,$A[2][3][58]{=}v_{61}$, $A[2][4][58]{=}v_{60} + v_{61}$,$A[2][2][62]{=}A[2][3][62]{=}v_{62}$,$A[2][2][63]{=}A[2][3][63]{=}v_{63}$ |
| **Related Key Bits** |
| $k_0[9]$,$k_0[10]$,$k_0[12]$,$k_0[16]$,$k_0[17]$,$k_0[19]$,$k_0[20]$,$k_0[23]$, $k_0[26]$,$k_0[29]$,$k_0[30]$,$k_0[33]$,$k_0[36]$,$k_0[37]$, $k_0[40]$,$k_0[43]$,$k_0[44]$,$k_0[50]$, $k_0[51]$,$k_0[55]$, $k_0[56]$,$k_0[57]$,$k_0[61]$,$k_0[62]$,$k_0[63]$,$k_2[2]$, $k_2[3]$,$k_2[4]$,$k_2[5]$, |
| **Guessed Key bits** |
| $k_0[29]$,$k_0[44]$,$k_0[50]$,$k_0[51]$,$k_0[55]$, $k_0[56]$,$k_0[57]$,$k_0[61]$,$k_0[62]$,$k_0[63]$,$k_2[2]$, $k_2[3]$,$k_2[4]$,$k_2[5]$ |
| **Auxiliary Variables** |
| $A[0][1][9]{=}a_0$,$A[0][1][10]{=}a_1$,$A[0][1][12]{=}a_2$,$A[0][1][16]{=}a_3$,$A[0][1][17]{=}a_4$, $A[0][1][19]{=}a_5$,$A[0][1][20]{=}a_6$,$A[0][1][23]{=}a_7$, $A[0][1][26]{=}a_8$,$A[0][1][30]{=}a_9$, $A[0][1][33]{=}a_{10}$,$A[0][1][36]{=}a_{11}$,$A[0][1][37]{=}a_{12}$,$A[0][1][40]{=}a_{13}$,$A[0][1][43]{=}a_{14}$ |

Table 12: 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** |
| $k_0[8]$,$k_0[9]$,$k_0[10]$,$k_0[11]$,$k_0[12]$,$k_0[13]$, $k_0[14]$,$k_0[15]$,$k_0[16]$,$k_0[17]$,$k_0[18]$,$k_0[19]$,$k_0[20]$, $k_0[21]$,$k_0[22]$,$k_0[23]$,$k_0[24]$, $k_0[25]$,$k_0[26]$,$k_0[27]$, $k_0[28]$,$k_0[29]$,$k_0[30]$,$k_0[31]$,$k_3[16]$,$k_3[17]$, $k_3[18]$,$k_3[19]$,$k_3[20]$,$k_3[21]$,$k_3[22]$,$k_3[23]$, $k_3[24]$, $k_3[25]$,$k_3[26]$,$k_3[27]$,$k_3[28]$,$k_3[29]$,$k_3[30]$, $k_3[31]$,$k_4[0]$,$k_4[1]$,$k_4[2]$,$k_4[3]$,$k_4[4]$,$k_4[5]$,$k_4[6]$,$k_4[7]$ |
| **Auxiliary Variables** |
| $A[1][3][0]=a_0+v_0$,$A[1][3][1]=a_1+v_1$,$A[1][3][2]=a_2+v_2$,$A[1][3][3]=a_3+v_3$, $A[1][3][4]=a_4+v_4$,$A[1][3][5]=a_5+v_5$, $A[1][3][6]=a_6+v_6$,$A[1][3][7]=a_7+v_7$, $A[1][3][8]=a_8+v_8$,$A[1][3][9]=a_9+v_9$,$A[1][3][10]=a_{10}+v_{10}$, $A[1][3][11]=a_{11}+v_{11}$, $A[1][3][12]=a_{12}+v_{12}$,$A[1][3][13]=a_{13}+v_{13}$,$A[1][3][14]=a_{14}+v_{14}$,$A[1][3][15]=a_{15}+v_{15}$, $A[1][3][16]=a_{16}+v_{16}$,$A[1][3][17]=a_{17}+v_{17}$,$A[1][3][18]=a_{18}+v_{18}$,$A[1][3][19]=a_{19}+v_{19}$, $A[1][3][20]=a_{20}+v_{20}$, $A[1][3][21]=a_{21}+v_{21}$,$A[1][3][22]=a_{22}+v_{22}$,$A[1][3][23]=a_{23}+v_{23}$, $A[1][3][24]=a_{24}+v_{24}$,$A[1][3][25]=a_{25}+v_{25}$, $A[1][3][26]=a_{26}+v_{26}$,$A[1][3][27]=a_{27}+v_{27}$, $A[1][3][28]=a_{28}+v_{28}$,$A[1][3][29]=a_{29}+v_{29}$,$A[1][3][30]=a_{30}+v_{30}$, $A[1][3][31]=a_{31}+v_{31}$, $A[3][0][0]=a_{32}+v_{32}$,$A[3][0][1]=a_{33}+v_{33}$,$A[3][0][2]=a_{34}+v_{34}$,$A[3][0][3]=a_{35}+v_{35}$, $A[3][0][4]=a_{36}+v_{36}$,$A[3][0][5]=a_{37}+v_{37}$,$A[3][0][6]=a_{38}+v_{38}$,$A[3][0][7]=a_{39}+v_{39}$, $A[3][0][8]=a_{40}+v_{40}$, $A[3][0][9]=a_{41}+v_{41}$,$A[3][0][10]=a_{42}+v_{42}$,$A[3][0][11]=a_{43}+v_{43}$, $A[3][0][12]=a_{44}+v_{44}$,$A[3][0][13]=a_{45}+v_{45}$, $A[3][0][14]=a_{46}+v_{46}$,$A[3][0][15]=a_{47}+v_{47}$ |

Table 13: 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** |
| $A[1][3][0]=a_0+v_{19}$,$A[1][3][1]=a_1$,$A[1][3][2]=a_2+v_{20}$,$A[1][3][3]=a_3$, $A[1][3][4]=a_4+v_{21}$,$A[1][3][5]=a_5$,$A[1][3][7]=a_7$, $A[1][3][6]=a_6+v_{22}$,$A[1][3][8]=a_8$, $A[1][3][9]=a_9$,$A[1][3][10]=a_{10}+v_{23}$,$A[1][3][11]=a_{11}$,$A[1][3][13]=a_{13}+v_{24}$, $A[1][3][12]=a_{12}$,$A[1][3][14]=a_{14}$,$A[1][3][15]=a_{15}+v_{25}$,$A[1][3][27]=a_{27}+v_{34}$,$A[1][3][19]=a_{19}$, $A[1][3][17]=a_{17}+v_{26}$, $A[1][3][18]=a_{18}+v_{27}$,$A[1][3][20]=a_{20}+v_{28}$, $A[1][3][21]=a_{21}$,$A[1][3][22]=a_{22}+v_{29}$,$A[1][3][23]=a_{23}+v_{30}$, $A[1][3][24]=a_{24}+v_{31}$, $A[1][3][25]=a_{25}+v_{32}$,$A[1][3][26]=a_{26}+v_{33}$,$A[1][3][16]=a_{16}$,$A[1][3][28]=a_{28}+v_{35}$ |