Improved Preimage Attacks on Round-Reduced Keccak-384/512 via Restricted Linear Structures

Le He, Xiaoen Lin, and Hongbo Yu

Department of Computer Science and Technology, Tsinghua University, Bejing 100084, China

Abstract. This paper provides improved preimage analysis on round-reduced Keccak-384/512. Unlike low-capacity versions, Keccak-384/512 outputs from two planes of its inner state: an entire 320-bit plane and a second plane containing 64/192 bits. Due to lack of degrees of freedom, most existing preimage analysis can only control the 320-bit plane and cannot achieve good results. In this paper, we find out a method to construct linear relations between corresponding bits from the two planes, which means attacker can control two output planes simultaneously with degrees of freedom much less than 320. Besides, we design several linear structures for each different version with additional restrictions that can leave more degrees of freedom. As a result, the complexity of preimage attacks on 2-round Keccak-384/512 and 3-round Keccak-384/512 can be decreased to $2^{28}/2^{252}$ and $2^{271}/2^{426}$ respectively, which are all the best known results so far. To support the analysis, this paper also provides the first preimage of all '0' digest for 2-round Keccak-384, which can be obtained in hours level by a personal computer. It is worth noting that although our structures contain non-linear parts, the attack algorithms only involve the solution of linear equation systems.

Keywords: Keccak · Preimage attack · Linear relation

1 Introduction

The Keccak function, designed by Bertoni et al. [3, 4], was selected as the winner of SHA-3 competition in 2012 and finally standardized in 2015 by NIST. Since Keccak was proposed in 2008, there have been kinds of security cryptanalysis from public research community, including preimage [2, 14, 15], collision [5, 16, 19], distinguishing [1, 7, 8], keyed modes [6, 10, 18], and many other unmentioned security settings. Those advanced attack methods work well even with practical results in low-capacity Keccak — round-reduced Keccak-224/256. Yet for round-reduced Keccak-384/512, due to lack of freedom in message block setting, most methods cannot work as efficiently as they do in low-capacity versions.

In this paper, we mainly focus on preimage attacks on round-reduced Keccak-384/512 — more specifically, linear analysis. Our research is inspired by four creative works: Guo et al.'s [8], Li et al.'s [13], Rajasree's [17] and He et al.'s [9]. In 2016, Guo et al. [8] adopted a strategy of linear structure in both distinguishing and preimage cryptanalysis on different Keccak versions. Their idea is to linearize the whole inner state after several rounds with large amount of freedom space left. Yet for Keccak-384/512, their structures can only pass through 1 round and thus they have to adopt other advanced technologies to achieve good results, which are not required in this paper. Then in 2019, Li et al. [13] proposed an improved structure with an allocating model: the first message block aims to generate a restricted middle state satisfying several conditions, and the second message block (XORed with the restricted middle state) can obtain extra gains in preimage searching. They adopted this model merely on reduced-round Keccak-224/256. Yet it can also be simply applied to 3-round Keccak-384 (we will give a brief discussion in Section 3.1). Rajasree [17] made an improvement from another perspective. He noticed that the number of degrees of freedom left is much less than the number of linear equations. Thus he allowed non-linear parts existing in the structure and just constructed equations in linear parts. This idea would not increase the number of degrees of freedom, but enlarge the space of random constants, which is also a noticeable problem of high-capacity versions. In 2021, He et al. [9] proposed a technology named *zero coefficient*. It actually refers to some linear-dependent bit pairs in the inner state. Using this technology, they successfully set 173 equations within 162 degrees of freedom, obtaining 11 linear-dependent bit pairs. This result is limited just because their analysis object is Keccak-224/256. For Keccak-384/512 with two output planes, the number can be increased to lanes (a set of 64 bits) level. **Table 1** summarizes some existing preimage cryptanalysis results on round-reduced Keccak-384/512.

Table 1. Summary of preimage cryptanalysis on round-reduced Keccak-384/512.

Round	Instance	Complexity*	Reference
2	Keccak-384	2^{129}	[8]
2	Keccak-384	2^{113}	[17]
2	Keccak-384	2^{89}	[11]
2	Keccak-384	2^{28}	Section 4.1
2	Keccak-512	2^{384}	[8]
2	Keccak-512	2^{321}	[17]
2	Keccak-512	2^{252}	Section 4.2
3	Keccak-384	2^{322}	[8]
3	Keccak-384	2^{321}	[17]
3	Keccak-384	2^{271}	Section 5.1
3	Keccak-512	2^{482}	[8]
3	Keccak-512	2^{475}	[17]
3	Keccak-512	2^{426}	Section 5.2
4	Keccak-384	2^{371}	[17]

*Note: The results here refer to guess times instead of Keccak calls, which do not include the complexity of solving the linear equation systems (the constant factor is about $2^8 \sim 2^{10}$).

Our contributions. Inspired by previous works, we propose two new technologies on linear analysis of round-reduced Keccak. One is named *non-linear liberalization*. Unlike Rajasree's thought [17], this technology can indeed improve the linear structure with more degrees of freedom left. The other is named *extra linear dependence*. Through this technology, attacker can construct extra linear relations between corresponding bits from different planes without spending any degrees of freedom. These technologies are especially suitable for round-reduced high-capacity Keccak, which contains two output planes with linear structures passing through. It is worth mentioning that in [12] the authors have proposed a practical attack for 1-round Keccak-512. Thus we apply these technologies to versions starting from two rounds. As a result, we improve the preimage analysis of 2-round Keccak-384/512 and 3-round Keccak-384/512 in two aspects:

1. With non-linear parts existing, we optimize the linear structures of different versions that can leave 320/192 and 64/128 degrees of freedom.

2. By carefully setting restricting constants and counting random space, we obtain 128/192 and 96/128 linear-dependent bit pairs from two output planes.

After analyzing the probability of eliminating those quadratic terms and the gain of those linear-dependent bit pairs, it is concluded that the complexity of preimage attacks on 2-round Keccak-384/512 and 3-round Keccak-384/512 can be decreased to $2^{28}/2^{252}$ and $2^{271}/2^{426}$ respectively (message padding has been taken into account). An example preimage attack of all '0' digest for 2-round Keccak-384 is given in Section 4.1.

Organization. The paper starts with some preliminaries and notations of Keccak in Section 2. In Section 3, we introduce the core technologies involved in our work. Improved preimage attacks on 2-round Keccak-384/512 and 3-round Keccak-384/512 are provided separately in Section 4 and Section 5. Conclusions are summarized in Section 6.

2 Preliminaries

This section gives the descriptions about sponge construction, Keccak-f permutation, SHA-3 standard, some properties of Keccak-f, and the meanings of those notations used in this paper.

2.1 Sponge Construction

The Keccak function adopts a new iterative construction named *sponge*, which involves three parameters r, c, ℓ and a permutation Keccak-$f[b]$ with $b = r + c$ (as depicted in **Fig. 1**). This construction processes a message in two phases — absorbing phase and squeezing phase. In absorbing phase, the message M (after padding) is split into r-bit blocks. Starting with a b-bit all '0' IV, its first r bits are XORed with the first message block, followed by an execution of Keccak-f. After all message blocks are processed similarly, it comes to the squeezing phase. In the squeezing phase, the construction outputs an r-bit digest and mixes the inner state by executing Keccak-f, repeating until the digest length reaches ℓ. Finally, the digest is truncated to the first ℓ bits.

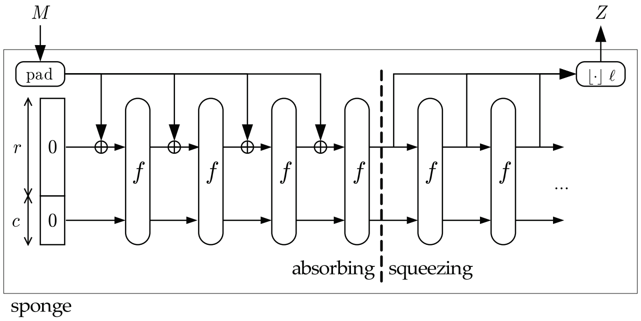

Fig. 1. The sponge construction.

2.2 Keccak-f Permutation

The core of Keccak-f is its b-bit inner state. In [4], the designers provided seven Keccak-f permutations with $b \in \{25, 50, 100, 200, 400, 800, 1600\}$. NIST finally chose $b = 1600$ as SHA-3 standard. In this paper, we also consider the case of $b = 1600$ only.

In the case of $b = 1600$, the inner state can be organized as 5×5 64-bit lanes like **Fig. 2**. Each bit is denoted as $A_{x,y,z}$, where x varies from 0 to 4, y varies from 0 to 4, and z varies down from 63 to 0 (counting from the most significant bit) as the arrows in **Fig. 2** show. The r-bit part of the inner state piles in order of $A_{0,0,0} \sim A_{0,0,63}, A_{1,0,0} \sim A_{1,0,63}, \ldots, A_{4,0,0} \sim A_{4,0,63}, A_{0,1,0} \sim A_{0,1,63} \cdots$

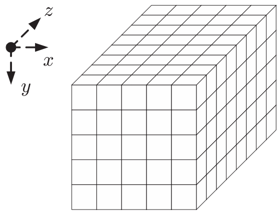

Fig. 2. The inner state of Keccak-f.

The Keccak-f permutation consists of 24 rounds of function R, and each R consists of five steps $R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$, where:

$$\begin{aligned}
\theta : A_{x,y,z} &= A_{x,y,z} \oplus \bigoplus_{j=0\sim 4} (A_{x-1,j,z} \oplus A_{x+1,j,z-1}) \\
\rho : A_{x,y,z} &= A_{x,y,z-r_{x,y}} \\
\pi : A_{x,y,z} &= A_{x+3y,x,z} \\
\chi : A_{x,y,z} &= A_{x,y,z} \oplus (A_{x+1,y,z} \oplus 1) \cdot A_{x+2,y,z} \\
\iota : A_{0,0,z} &= A_{0,0,z} \oplus RC_z
\end{aligned}$$

In the formulas above, "$\oplus$" means bit-wise XOR and "$\cdot$" means bit-wise AND. Indices x and y are calculated modulo 5 and index z is calculated modulo 64. Besides, $r_{x,y}$ refers to a lane-dependent rotation constant as shown in **Table 2**. RC is a round-dependent constant. We omit the details of RC here since those constants do not affect our attack methods.

Table 2. The offsets of ρ.

	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$
$y=0$	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

2.3 SHA-3 Standard

Any Keccak instance can be denoted as Keccak$[r, c, \ell]$ with bitrate r, capacity c and digest length ℓ. In [20], NIST standardized four SHA-3 versions that have $r = 1600 - 2\ell$ and $c = 2\ell$, where $\ell \in \{224, 256, 384, 512\}$. Therefore, we can use Keccak-ℓ or SHA-3-ℓ to denote a SHA-3 version for short.

The only difference between Keccak-ℓ and SHA-3-ℓ is padding rule: Keccak pads the message by 10*1 while SHA-3 pads the message by 0110*1. This means that for both Keccak and SHA-3, the last bit of the r-bit part (corresponding to $z = 63$) must be '1' and for SHA-3 only, the penultimate '1' must follow "01". The preimage cryptanalysis results in this paper are applicable for Keccak: when applied to SHA-3, the complexity will be a bit higher.

2.4 Properties of Keccak-f

Most properties of Keccak-f have been discussed in previous works [8, 9, 13, 17]. For simplicity, here we just state related properties without any proof.

1. Step θ. θ is the linear diffusion layer of Keccak-f. Through θ, the variation of a single bit will spread to 11 bits. An important property of θ is, the XOR value of two bits from identical column always holds, which can be written as:

$$A_{x,y_1,z} \oplus A_{x,y_2,z} = \theta(A)_{x,y_1,z} \oplus \theta(A)_{x,y_2,z}$$

2. Step $\pi \circ \rho$. $\pi \circ \rho$ can be regarded as a bit-level permutation function of the inner state. In this paper, we always combine ρ together with π.

3. Step χ. χ is the only non-linear step of Keccak-f, which can be regarded as a 5-bit Sbox. The output of χ can keep linear if and only if the input contains at most two discontinuous unknown variables. When 5 output bits are all known

(the 320-bit plane), the input bits can be entirely inversed. When 5 output bits are partially known (the 64/192-bit plane), extra input restrictions are required to match the known part. Particularly, to match 3 continuous output bits $b_0b_1b_2$ (the 192-bit plane), two linear restrictions of input bits a_i can be inversed:

$$a_0 \oplus (b_1 \oplus 1) \cdot a_2 = b_0$$
$$a_1 \oplus (b_2 \oplus 1) \cdot a_3 = b_1$$

4. Step ι. Since ι only runs in $A_{0,0,z}$, the last ι can always be inversed in any Keccak version. In this paper, except for the instance of 2-round Keccak-384 that correctly matches an all '0' digest, we omit the last ι in other Keccak versions: we regard the digest as truncated from the inner state after the last χ.

2.5 Notations

From this section on, we will no longer use A to denote the inner state, since it cannot accurately show its execution process. Instead, we will use capital Greek letters (in $\{\Theta, P, \Pi, X, I\}$) with a superscript (from 1 to 3) to express the state exactly **after** the corresponding step is executed. For example, Π^2 denotes the inner state after the second π, and X^3 denotes the inner state after the third χ. In particular, I^0 denotes the initial input (after XORing the message block).

To avoid ambiguity, we will always use three indices in subscript to denote a part of the inner state. However, we may use "$*$" to indicate all possible values. For example, $I^1_{*,y,z}$ is a 5-bit row, $I^1_{x,*,z}$ is a 5-bit column, $I^1_{x,y,*}$ is a 64-bit lane, $I^1_{*,y,*}$ is a 320-bit plane, and $I^1_{*,*,z}$ is a 5×5 slice. If the subscript is omitted, it indicates the 1600-bit whole state (like the notations above).

Column sum setting is the core issue of our preimage attacks. In this paper, we use S_A with two parameters x, z to denote the sum of a certain column from state A, which is:

$$S_A(x,z) = \bigoplus_{y=0\sim 4} A_{x,y,z}$$

3 Technology Overview

This section introduces four core technologies involved in our preimage attacks: restricted linear structure, allocating model, non-linear liberalization and extra linear dependence.

3.1 Restricted Linear Structure

Linear structure is a significant idea in linear analysis on round-reduced Keccak — not limited to preimage attack. Our paper inherits this idea from Guo et al.'s work [8]. Actually the linear structures in [8] for Keccak-384/512 can only pass through 1 round. Here we take a 2-round structure for Keccak-384 as an example (as depicted in **Fig. 3**). In order to pass through the second non-linear Sbox, I^0 must satisfy extra restrictions. Thus it is named *restricted linear structure*.

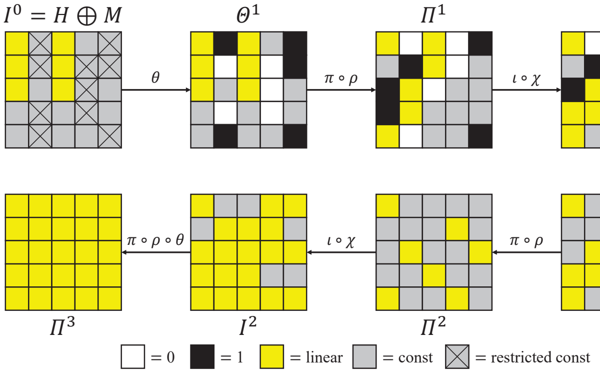

Fig. 3. A 2-round linear structure for Keccak-384.

Explanations for **Fig. 3** are in order of Keccak-f execution:

1. State I^0. I^0 contains 6 variable lanes and 10 restricted (and 9 free) constant lanes, corresponding to 384 initial degrees of freedom and 448 restrictions.

2. State I^0 to state Θ^1. The number of variable lanes remains 6 in Θ^1, which means variables could not spread through the diffusion layer θ. To prevent such diffusion, related column sums need to be fixed, which correspond to 128 linear equations ($z = 0 \sim 63$, red indicates variables, the same in other equations):

$$\begin{cases} I^0_{0,0,z} \oplus I^0_{0,1,z} \oplus I^0_{0,2,z} = S_{I^0}(0,z) \oplus I^0_{0,3,z} \oplus I^0_{0,4,z} \\ I^0_{2,0,z} \oplus I^0_{2,1,z} \oplus I^0_{2,2,z} = S_{I^0}(2,z) \oplus I^0_{2,3,z} \oplus I^0_{2,4,z} \end{cases}$$

In addition, Θ^1 restricts 10 constant lanes to be all '0' or '1'. Then according to the property of θ stated in Section 2.4, 448 restrictions are required before θ:

$$\begin{cases} I^0_{1,0,z} = I^0_{1,1,z} \oplus 1 = I^0_{1,3,z} \oplus 1 = I^0_{1,4,z} \\ I^0_{3,1,z} = I^0_{3,2,z} = I^0_{3,3,z} \\ I^0_{4,0,z} = I^0_{4,1,z} = I^0_{4,4,z} \end{cases}$$

3. State Θ^1 to state Π^1. In structure figures, the 1600-bit permutation $\pi \circ \rho$ can be regarded as a 25-lane color-swap.

4. State Π^1 to state I^1. The constants of '0' and '1' in Π^1 can partly control the output of the first non-linear Sbox. Notice that this structure doesn't restrict $\Pi^1_{2,3,*}$ to be all '0'. Thus variables may possibly spread to $I^1_{0,3,*}$ after χ.

5. State I^1 to state Θ^2. Similarly, 192 column sums are fixed to prevent the diffusion of variables ($I^1_{0,3,z}$ may be constant or variable, indicated by purple):

$$\begin{cases} I^1_{0,0,z} \oplus I^1_{0,3,z} \oplus I^1_{0,4,z} = S_{I^1}(0,z) \oplus I^1_{0,1,z} \oplus I^1_{0,2,z} \\ I^1_{1,2,z} \oplus I^1_{1,3,z} = \oplus S_{I^1}(1,z) \oplus I^1_{1,0,z} \oplus I^1_{1,1,z} \oplus I^1_{1,4,z} \\ I^1_{2,0,z} \oplus I^1_{2,1,z} = \oplus S_{I^1}(2,z) \oplus I^1_{2,2,z} \oplus I^1_{2,3,z} \oplus I^1_{2,4,z} \end{cases}$$

6. State Θ^2 to state Π^2. Variable lanes are swapped to different sites.

7. State Π^2 to state I^2. According to the property of χ stated in Section 2.4, without any continuous variables in Π^2, the outputs in I^2 can keep linear.

8. State I^2 to state Π^3. Through three more linear steps, the linear structure finally reaches the third χ with $384 - 128 - 192 = 64$ degrees of freedom left.

This linear structure can be directly applied in preimage attack on 3-round Keccak-384. According to the inversion property, $\Pi^3_{*,0,*}$ can be entirely inversed from the target digest. Thus this structure can support a preimage attack with a correct rate of 2^{-321} (counting 1-bit message padding).

The basic principle of other restricted linear structures in this paper (mainly in Section 4 and Section 5) is just the same. For simplicity, there we only provide a structure figure with concrete column sum equations and I^0 restrictions.

3.2 Allocating Model

The linear structure in Section 3.1 requires 448 I^0 restrictions. However, among these restrictions, $I^0_{1,3,z} \oplus 1 = I^0_{1,4,z}$ and $I^0_{3,2,z} = I^0_{3,3,z}$ cannot simply be satisfied by modifying the message block, since r-bit M merely reaches $I^0_{2,2,63}$. Those 128 restrictions must be satisfied before XORing the message block, which involves an allocating model (as depicted in **Fig. 4**).

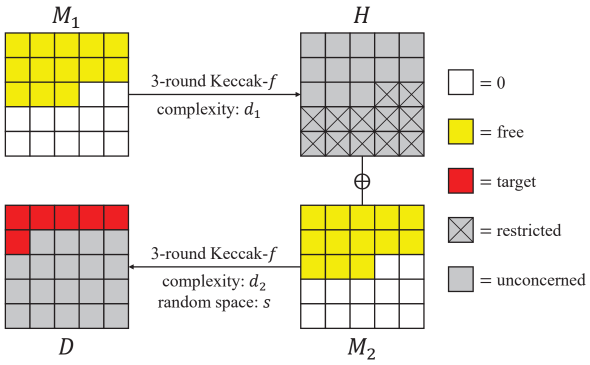

Fig. 4. The allocating model (example of 3-round Keccak-384).

In this allocating model, attacker first finds out a message block M_1 to let the middle state H meet those restrictions. The searching complexity is denoted as d_1. XORed with the middle state, attacker then finds out a message block M_2 to match the target digest. The searching complexity is denoted as d_2. Certainly, those restrictions can help decrease d_2. Another important issue is, the random space size s of M_2 must not be less than d_1. For high-capacity Keccak, usually $s < d_2$, which means attacker requires $[d_2/s]$ H to construct a preimage. Then if $s < d_1$, the total complexity becomes $[d_1/s] \times d_2$ [1], instead of d_2.

[1] Here we ignore the factor difference from solving the linear equation systems.

The random space consists of two parts: free bits from the message block, and controllable column sums from the linear equation system. Take the structure in **Fig. 3** as an example.

The message block contains two free constant lanes $I^0_{1,2,*}$ and $I^0_{3,0,*}$, which bring a space of 2^{128}. As for the linear equation system, $S_{I^1}(0,*)$, $S_{I^1}(1,*)$ and $S_{I^1}(2,*)$ in the second θ are all free to be '0' or '1', which bring a space of 2^{192}. Yet $S_{I^0}(0,*)$ and $S_{I^0}(2,*)$ in the first θ are restricted by the equations below:

$$\begin{cases} S_{I^0}(0,z) \oplus S_{I^0}(2,z-1) \oplus I^0_{1,3,z} = \Theta^1_{1,3,z} = 0 \\ S_{I^0}(2,z) \oplus S_{I^0}(4,z-1) \oplus I^0_{3,3,z} = \Theta^1_{3,3,z} = 0 \\ S_{I^0}(3,z) \oplus S_{I^0}(0,z-1) \oplus I^0_{4,4,z} = \Theta^1_{4,4,z} = 1 \end{cases}$$

In above linear equation system, $S_{I^0}(4,*)$ has been fixed by I^0 restrictions, and $S_{I^0}(3,*)$ is corresponding to $I^0_{3,0,*}$. Therefore, there is a space loss of 2^{64} in the first θ. In total, the random space size of M_2 is $s = 2^{256}$. With $d_1 = 2^{128}$ and $d_2 = 2^{321}$, the allocating model for 3-round Keccak-384 meets the relations.

3.3 Non-Linear Liberalization

The improved linear structures in this paper (mainly in Section 4 and Section 5) adopt two kinds of non-linear liberalization: liberalization to enlarge the random space and liberalization for extra degrees of freedom. The former is first adopted by Rajasree [17]. Although the structures in [17] cannot leave extra degrees of freedom, his idea really inspires us to design the latter.

Non-Linear Liberalization to Enlarge the Random Space The inspiration of this kind is: since the number of degrees of freedom left is less than the number of linear bits in Π^3, attacker can loosen some non-linear terms, partly polluting Π^3 but enlarging the random space. An example is given in **Fig. 5**.

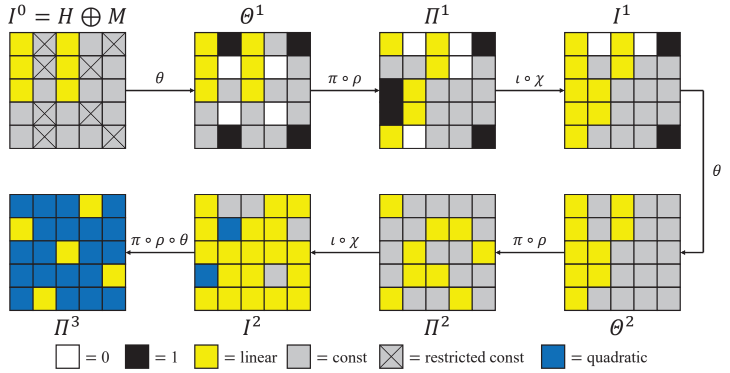

Fig. 5. A 2-round partly non-linear structure for Keccak-384.

Compared to **Fig. 3**, this structure loosens two restricted constant lanes $I^0_{3,2,*}$ and $I^0_{4,1,*}$. This results in two extra diffused variable lanes $I^1_{0,1,*}$ and $I^1_{0,2,*}$ after the first χ. Then in the second χ, continuous variables exist and generate two quadratic lanes. Those quadratic terms spread through the third θ and almost pollute the whole Π^3. However, 5 lanes can still remain linear in this structure. Meanwhile, the number of degrees of freedom left is equally $384 - 128 - 192 = 64$, which means attacker can exactly set 64 linear equations on $\Pi^3_{3,0,*}$.

In summary, this partly non-linear structure can support a preimage attack on 3-round Keccak-384 with a complexity of equally 2^{321}. However, through this structure, the random space size greatly increases: since this structure loosens 128 I^0 restrictions, the model parameters of such attack become $d_1 = 2^{64}$ and $s = 2^{320}$.

Non-Linear Liberalization for Extra Degrees of Freedom The inspiration of this kind is: since a quadratic term can be eliminated (equal to '0') with a probability of 3/4 instead of 1/2, if the quadratic term can exchange an extra degree of freedom, it can still bring a gain of $2^1 \times 3/4 = 2^{0.58}$. An example is given in **Fig. 6**.

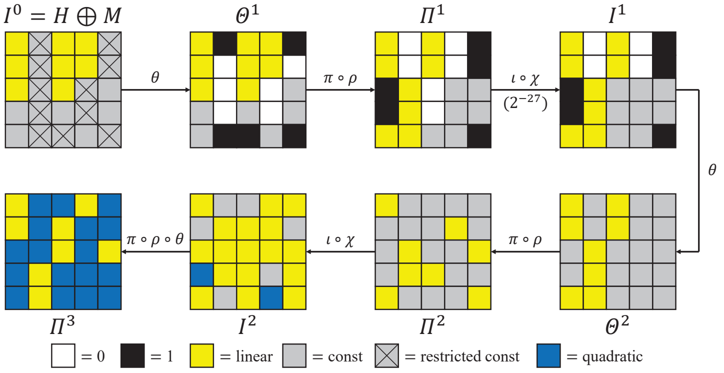

Fig. 6. An improved 2-round partly non-linear structure for Keccak-384.

In this improved structure, we directly add two variable lanes $I^0_{3,0,*}$ and $I^0_{3,1,*}$ (and extra I^0 restrictions). This results in continuous variables existing in Π^1. Then the core of this structure is, we regard $I^1_{4,4,z} = \Pi^1_{4,4,z} \oplus (\Pi^1_{0,4,z} \oplus 1) \cdot \Pi^1_{1,4,z} = \Pi^1_{4,4,z}$ with a correct rate of 3/4. The second round is similar to that in **Fig. 5**. Finally the number of degrees of freedom left increases to $512 - 192 - 192 = 128$, and attacker can exactly set 128 linear equations on $\Pi^3_{0,0,*}$ and $\Pi^3_{3,0,*}$. However, the probability of one solution passing through the first χ is only $(3/4)^{64} = 2^{-27}$.

In summary, this improved 2-round structure can support a preimage attack on 3-round Keccak-384 with a complexity of $2^{384-128+27+1} = 2^{284}$. The model parameters are $d_1 = 2^{192}$ and $s = 2^{192}$, which construct a legal attack. Notice that d_1/s has reached the limit and thus the technology of extra linear dependence cannot be applied in this structure. By applying extra linear dependence in an ordinary structure, the final complexity can be even lower.

3.4 Extra Linear Dependence

This technology is inspired by a similar technology in [9] named *zero coefficient*. The origin technology aimed to find out some linear-dependent pairs in a fixed linear equation system. As a result, the authors saved 11 degrees of freedom from 173 equations. Yet in this paper, we tend to "construct" rather than "find out" linear dependence — under this thought, attacker can even choose 128 target linear bits and match them one-to-one, saving 64 degrees of freedom. With those linear-dependent bit pairs, the complexity of preimage attacks on round-reduced high-capacity Keccak can be greatly decreased.

Principle of Extra Linear Dependence The principle of extra linear dependence is revealed in **Fig. 7**.

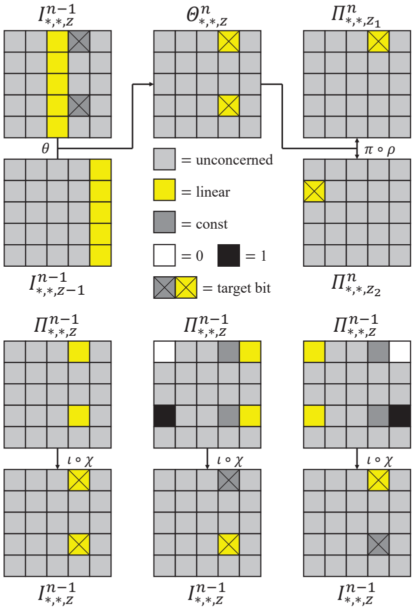

Fig. 7. Principle of extra linear dependence.

Suppose we set two linear equations on Π^n. Each equation is composed of 11 bits from Θ^n. Then, if two targets are permuted from the same column $\Theta^n_{x,*,z}$, 10 out of 11 bits are identical in the equation pair. Furthermore, if two unique bits are ensured to be constant, the linear dependence of the equation pair can be ensured. One step backward, the attribute of $I^{n-1}_{x,y,z}$ depends on related bits in Π^{n-1}. As shown in **Fig. 7**, if $\Pi^{n-1}_{x,y,z}$ has been a variable, $I^{n-1}_{x,y,z}$ can never be a constant, and if $\Pi^{n-1}_{x,y,z}$ is a constant, $I^{n-1}_{x,y,z}$ can be ensured by fixing related bits to be '0' or '1'. In general cases, constructing linear dependence between 1 pair of $I^{n-1}_{x,y_1,z}$ and $I^{n-1}_{x,y_2,z}$ requires 3 restrictions as below. Converting Π^{n-1} to Θ^{n-1}, then attacker can construct (1 pair of) linear dependence by setting $S_{I^{n-2}}$.

$$\begin{cases} \Pi^{n-1}_{x+1+c_1,y_1,z} = c_1 \oplus 1,\ c_1 = 0 \text{ or } 1 \\ \Pi^{n-1}_{x+1+c_2,y_2,z} = c_2 \oplus 1,\ c_2 = 0 \text{ or } 1 \\ \Pi^{n-1}_{x,y_1,z} \oplus \Pi^{n-1}_{x,y_2,z} = 0 \text{ or } 1 \end{cases}$$

Lanes Level Extra Linear Dependence We have introduced the method to construct 1 linear-dependent pair. However, when the number increases to lanes level, additional problems may arise — at least all restrictions must be satisfied without any contradictions. Meanwhile, since the random space size s must not be less than the complexity of d_1, we'd better reduce the number of restrictions if possible. Take the structure in **Fig. 5** as an example.

Since $\Pi^3_{3,0,*}$ and $\Pi^3_{0,1,*}$ are the only linear lanes related to the 384-bit digest, we just construct linear dependence between these two lanes. The target linear-dependent pairs are (for all $z = 0 \sim 63$):

$$(I^2_{3,0,z} \to \Pi^3_{0,1,z+28}) \oplus (I^2_{3,3,z} \to \Pi^3_{3,0,z+21}) = c(z)$$

where the constant attribute of $I^2_{3,0,z}$ requires fixed $\Pi^2_{4,0,z} = 1$, and the constant attribute of $I^2_{3,3,z}$ has been ensured.

Tracing back to Θ^2, the target bit pairs become (yellow indicates linear bits):

$$\begin{cases} I^2_{3,0,z} = \Pi^2_{3,0,z} \oplus (\Pi^2_{4,0,z} \oplus 1) \cdot \Pi^2_{0,0,z} = \Pi^2_{3,0,z} \\ I^2_{3,3,z} = \Pi^2_{3,3,z} \oplus (\Pi^2_{4,3,z} \oplus 1) \cdot \Pi^2_{0,3,z} \\ \Pi^2_{4,0,z} = \Theta^2_{4,4,z-14} = S_{I^1}(3, z-14) \oplus S_{I^1}(0, z-15) \oplus I^1_{4,4,z-14} = 1 \\ \Pi^2_{3,0,z} = \Theta^2_{3,3,z-21} = S_{I^1}(2, z-21) \oplus S_{I^1}(4, z-22) \oplus I^1_{3,3,z-21} \\ \Pi^2_{3,3,z} = \Theta^2_{2,3,z-15} = S_{I^1}(1, z-15) \oplus S_{I^1}(3, z-16) \oplus I^1_{2,3,z-15} \\ \Pi^2_{4,3,z} = \Theta^2_{3,4,z-56} = S_{I^1}(2, z-56) \oplus S_{I^1}(4, z-57) \oplus I^1_{3,4,z-56} \\ \Pi^2_{0,3,z} = \Theta^2_{4,0,z-27} = S_{I^1}(3, z-27) \oplus S_{I^1}(0, z-28) \oplus I^1_{4,0,z-27} \end{cases}$$

From the formulas above, it seems that from the perspective of S_{I^1}, $I^3_{3,0,z} \oplus I^3_{3,3,z} = c(z)$ is a quadratic equation, and to linearize the whole system, extra 64 restrictions on $\Pi^2_{4,3,*}$ or $\Pi^2_{0,3,*}$ need to be fixed. However, notice that in I^1 $I^1_{4,0,z} = I^1_{4,4,z} = 1$, which infers $\Pi^2_{4,0,z} = 1 \Leftrightarrow \Pi^2_{0,3,z} = 1$. Thus the whole system has actually been linearized by restricting $\Pi^2_{4,0,z} = 1$ for all $z = 0 \sim 63$. In other words, as long as we choose to fix $\Pi^2_{0,3,z} = 1$, the system can be solved without contradictions and 64 restrictions can be reduced. Under this case, the equation system of $I^3_{3,0,z} \oplus I^3_{3,3,z} = c(z)$ becomes:

$$\begin{cases} S_{I^1}(0,z) = S_{I^1}(3,z+1) \\ S_{I^1}(1,z) = S_{I^1}(2,z-6) \oplus S_{I^1}(2,z-41) \oplus S_{I^1}(3,z-1) \oplus S_{I^1}(4,z-7) \\ \qquad\qquad \oplus S_{I^1}(4,z-42) \oplus c(z+15) \oplus I^1_{2,3,z} \oplus I^1_{3,3,z-6} \oplus I^1_{3,4,z-41} \oplus 1 \end{cases}$$

In summary, by controlling S_{I^1} as above, $\Pi^3_{0,1,z+28} \oplus \Pi^3_{3,0,z+21} = c(z)$ can be ensured for all $z = 0 \sim 63$, where $c(z)$ is surely decided by $\Pi^3_{0,1,z+28} = I^3_{0,1,z+28}$ (with a correct rate of $3/4$) for the greatest gain. Thus these linear-dependent pairs can bring a gain of $(2^1 \times 3/4)^{64} = 2^{37}$. The model parameters of this attack are $d_1 = 2^{64}$, $s = 2^{192}$ and $d_2 = 2^{284}$, better than the attack's shown in **Fig. 6**.

Since d_1/s has not reached the limit, space of improvements still exists in preimage attack on 3-round Keccak-384. In following sections, we will directly present the best preimage cryptanalysis for different Keccak versions.

4 Preimage Cryptanalysis of 2-Round Keccak-384/512

Improved preimage cryptanalysis of 2-round Keccak-384/512 is presented in this section. For these two versions, we first design 1-round restricted linear structures with 320/192 degrees of freedom left. Both structures hold with a probability of 2^{-27} to eliminate quadratic terms. Then extra linear dependence is applied to construct linear relations between the output planes. As a result, we obtain 128 and 192 linear-dependent pairs which can bring extra gains of 2^{64} and 2^{96} (the latter meets the average case of 2-round Keccak-512). Finally, the complexity of preimage attacks on 2-round Keccak-384/512 is 2^{28} and 2^{252} respectively.

4.1 Improved Preimage Attack on 2-Round Keccak-384

The restricted linear structure for 2-round Keccak-384 is given in **Fig. 8**.

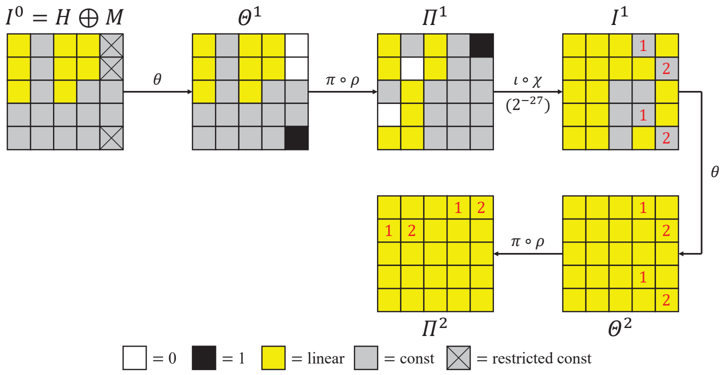

Fig. 8. Restricted linear structure for 2-round Keccak-384.

This structure requires 192 column sum equations and 128 I^0 restrictions as below, leaving 320 degrees of freedom. Since all I^0 restrictions can be ensured by modifying M, a 1-block model is enough for the attack.

$$\begin{cases} I^0_{4,0,z} = I^0_{4,1,z} = I^0_{4,4,z} \oplus 1 \\ I^0_{0,0,z} \oplus I^0_{0,1,z} \oplus I^0_{0,2,z} = S_{I^0}(0,z) \oplus I^0_{0,3,z} \oplus I^0_{0,4,z} \\ I^0_{2,0,z} \oplus I^0_{2,1,z} \oplus I^0_{2,2,z} = S_{I^0}(2,z) \oplus I^0_{2,3,z} \oplus I^0_{2,4,z} \\ I^0_{3,0,z} \oplus I^0_{3,1,z} = S_{I^0}(3,z) \oplus I^0_{3,2,z} \oplus I^0_{3,3,z} \oplus I^0_{3,4,z} \end{cases}$$

The target linear-dependent pairs are (as marked by red numbers in **Fig. 8**):

$$\begin{cases} (I^1_{3,0,z} \rightarrow \mathit{\Pi}^2_{0,1,z+28}) \oplus (I^1_{3,3,z} \rightarrow \mathit{\Pi}^2_{3,0,z+21}) = c_1(z) \\ (I^1_{4,1,z} \rightarrow \mathit{\Pi}^2_{1,1,z+20}) \oplus (I^1_{4,4,z} \rightarrow \mathit{\Pi}^2_{4,0,z+14}) = c_2(z) \end{cases}$$

where the constant attributes of $I^1_{3,0,z}$ and $I^1_{4,1,z}$ require $\mathit{\Pi}^1_{4,0,z} = 1$ and $\mathit{\Pi}^1_{1,1,z} = 0$, the constant attribute of $I^1_{4,4,z}$ depends on probable $(\mathit{\Pi}^1_{0,4,z} \oplus 1) \cdot \mathit{\Pi}^1_{1,4,z} = 0$ (with a correct rate of 3/4), and the constant attribute of $I^1_{3,3,z}$ has been ensured. Similarly, choosing $\mathit{\Pi}^1_{0,3,z} = 0$ can avoid contradictions and reduce restrictions.

Tracing back to $\mathit{\Theta}^1$, the target bit pairs become:

$$\begin{cases} I^1_{3,0,z} = \mathit{\Pi}^1_{3,0,z} \oplus (\mathit{\Pi}^1_{4,0,z} \oplus 1) \cdot \mathit{\Pi}^1_{0,0,z} = \mathit{\Pi}^1_{3,0,z} \\ I^1_{3,3,z} = \mathit{\Pi}^1_{3,3,z} \oplus (\mathit{\Pi}^1_{4,3,z} \oplus 1) \cdot \mathit{\Pi}^2_{0,3,z} = \mathit{\Pi}^1_{3,3,z} \\ I^1_{4,1,z} = \mathit{\Pi}^1_{4,1,z} \oplus (\mathit{\Pi}^1_{0,1,z} \oplus 1) \cdot \mathit{\Pi}^1_{1,1,z} = \mathit{\Pi}^1_{4,1,z} \\ I^1_{4,4,z} = \mathit{\Pi}^1_{4,4,z} \oplus (\mathit{\Pi}^1_{0,4,z} \oplus 1) \cdot \mathit{\Pi}^1_{1,4,z} = \mathit{\Pi}^1_{4,4,z} \\ \mathit{\Pi}^1_{4,0,z} = \mathit{\Theta}^1_{4,4,z-14} = S_{I^0}(3,z-14) \oplus S_{I^0}(0,z-15) \oplus I^0_{4,4,z-14} = 1 \\ \mathit{\Pi}^1_{0,3,z} = \mathit{\Theta}^1_{4,0,z-27} = S_{I^0}(3,z-27) \oplus S_{I^0}(0,z-28) \oplus I^0_{4,0,z-27} = 0 \\ \mathit{\Pi}^1_{1,1,z} = \mathit{\Theta}^1_{4,1,z-20} = S_{I^0}(3,z-20) \oplus S_{I^0}(0,z-21) \oplus I^0_{4,1,z-20} = 0 \\ \mathit{\Pi}^1_{3,0,z} = \mathit{\Theta}^1_{3,3,z-21} = S_{I^0}(2,z-21) \oplus S_{I^0}(4,z-22) \oplus I^0_{3,3,z-21} \\ \mathit{\Pi}^1_{3,3,z} = \mathit{\Theta}^1_{2,3,z-15} = S_{I^0}(1,z-15) \oplus S_{I^0}(3,z-16) \oplus I^0_{2,3,z-15} \\ \mathit{\Pi}^1_{4,1,z} = \mathit{\Theta}^1_{2,4,z-61} = S_{I^0}(1,z-61) \oplus S_{I^0}(3,z-62) \oplus I^0_{2,4,z-61} \\ \mathit{\Pi}^1_{4,4,z} = \mathit{\Theta}^1_{1,4,z-2} = S_{I^0}(0,z-2) \oplus S_{I^0}(2,z-3) \oplus I^0_{1,4,z-2} \end{cases}$$

Under this case, the linear equation system of S_{I^0} is:

$$\begin{cases} S_{I^0}(3,z) \oplus S_{I^0}(0,z-1) = I^0_{4,4,z} \oplus 1 \\ S_{I^0}(1,z-15) \oplus S_{I^0}(3,z-16) \oplus S_{I^0}(2,z-21) \oplus S_{I^0}(4,z-22) \\ = c_1(z) \oplus I^0_{2,3,z-15} \oplus I^0_{3,3,z-21} \\ S_{I^0}(0,z-2) \oplus S_{I^0}(2,z-3) \oplus S_{I^0}(1,z-61) \oplus S_{I^0}(3,z-62) \\ = c_2(z) \oplus I^0_{1,4,z-2} \oplus I^0_{2,4,z-61} \end{cases}$$

where $c_1(z)$ and $c_2(z)$ are set in line with $\mathit{\Pi}^2_{0,1,z} = I^2_{0,1,z}$ and $\mathit{\Pi}^2_{1,1,z} = 1$ for the greatest gain (2^{64}).

Each time attacker chooses qualified S_{I^0} and sets 192 column sum equations. With 320 target equations on $\Pi^2_{*,0,*}$, the 384-bit digest can be matched as long as the solution passes through the first χ with a probability of 2^{-27}. In summary, the complexity of preimage attack on 2-round Keccak-384 is only 2^{28} (including 1-bit padding). **Table 3** gives a practical preimage of all '0' digest.

Table 3. A preimage of all '0' digest for 2-round Keccak-384 (in big-endian order).

Starting State I^0 (one message block)
65fbd7e20b5fe6b4 0000000000000000 b7fb5afa8f3f1ffb dd2d29a4b4194993 ffffffffffffffff
9bec84cf16dc95f5 fffffffffd9c96b1 09e053aed207f2d7 dd2d292436194993 ffffffffffffffff
01e8ac92a37c8cbe fffffffffd9c96b1 be1b097079a8ed2c 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000

4.2 Improved Preimage Attack on 2-Round Keccak-512

The restricted linear structure for 2-round Keccak-512 is given in **Fig. 9**.

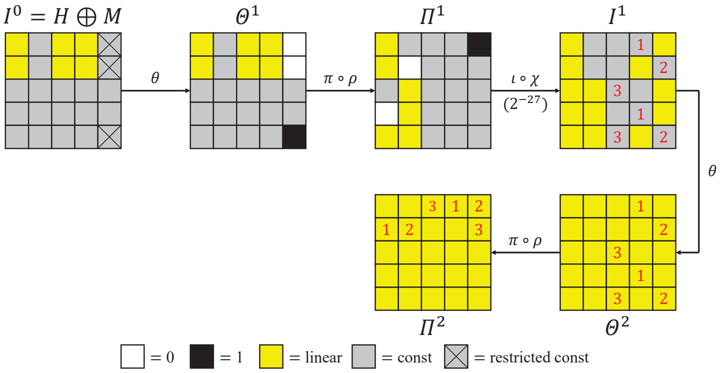

Fig. 9. Restricted linear structure for 2-round Keccak-512.

This structure requires 192 column sum equations and 128 I^0 restrictions as below, leaving 192 degrees of freedom. Since M cannot affect $I^0_{4,1,*}$, the attack requires a 2-block model with $d_1 = 2^{64}$.

$$\begin{cases} I^0_{4,0,z} = I^0_{4,1,z} = I^0_{4,4,z} \oplus 1 \\ {\color{red}I^0_{0,0,z}} \oplus {\color{red}I^0_{0,1,z}} = S_{I^0}(0,z) \oplus I^0_{0,2,z} \oplus I^0_{0,3,z} \oplus I^0_{0,4,z} \\ {\color{red}I^0_{2,0,z}} \oplus {\color{red}I^0_{2,1,z}} = S_{I^0}(2,z) \oplus I^0_{2,2,z} \oplus I^0_{2,3,z} \oplus I^0_{2,4,z} \\ {\color{red}I^0_{3,0,z}} \oplus {\color{red}I^0_{3,1,z}} = S_{I^0}(3,z) \oplus I^0_{3,2,z} \oplus I^0_{3,3,z} \oplus I^0_{3,4,z} \end{cases}$$

The target linear-dependent pairs are (as marked by red numbers in **Fig. 9**):

$$\begin{cases} (I^1_{3,0,z} \to \varPi^2_{0,1,z+28}) \oplus (I^1_{3,3,z} \to \varPi^2_{3,0,z+21}) = c_1(z) \\ (I^1_{4,1,z} \to \varPi^2_{1,1,z+20}) \oplus (I^1_{4,4,z} \to \varPi^2_{4,0,z+14}) = c_2(z) \\ (I^1_{2,2,z} \to \varPi^2_{2,0,z+43}) \oplus (I^1_{2,4,z} \to \varPi^2_{4,1,z+61}) = c_3(z) \end{cases}$$

where the constant attributes of $I^1_{3,0,z}$ and $I^1_{4,1,z}$ require $\varPi^1_{4,0,z} = 1$ and $\varPi^1_{1,1,z} = 0$, the constant attribute of $I^1_{4,4,z}$ depends on probable $(\varPi^1_{0,4,z} \oplus 1) \cdot \varPi^1_{1,4,z} = 0$ (with a correct rate of 3/4), and the constant attributes of $I^1_{3,3,z}$, $I^1_{2,2,z}$ and $I^1_{2,4,z}$ have been ensured.

Among those target pairs, all c_1 and c_2 parts can be ensured by an equation system exactly the same as that in Section 4.1:

$$\begin{cases} S_{I^0}(3,z) \oplus S_{I^0}(0,z-1) = I^0_{4,4,z} \oplus 1 \\ S_{I^0}(1,z-15) \oplus S_{I^0}(3,z-16) \oplus S_{I^0}(2,z-21) \oplus S_{I^0}(4,z-22) \\ = c_1(z) \oplus I^0_{2,3,z-15} \oplus I^0_{3,3,z-21} \\ S_{I^0}(0,z-2) \oplus S_{I^0}(2,z-3) \oplus S_{I^0}(1,z-61) \oplus S_{I^0}(3,z-62) \\ = c_2(z) \oplus I^0_{1,4,z-2} \oplus I^0_{2,4,z-61} \end{cases}$$

In this linear equation system, $S_{I^0}(4,*)$ has been fixed by I^0 restrictions, and $S_{I^0}(1,*)$ is controllable by setting $I^0_{1,0,*}$ and $I^0_{1,1,*}$. Moreover, as long as $S_{I^0}(1,*)$ is fixed, the whole S_{I^0} will be decided. Then from the perspective of $S_{I^0}(1,*)$, ensuring $c_3(z)$ becomes a quadratic problem:

$$\begin{cases} I^1_{2,2,z} = \varPi^1_{2,2,z} \oplus (\varPi^1_{3,2,z} \oplus 1) \cdot \varPi^1_{4,2,z} \\ I^1_{2,4,z} = \varPi^1_{2,4,z} \oplus (\varPi^1_{3,4,z} \oplus 1) \cdot \varPi^1_{4,4,z} \\ \varPi^1_{2,2,z} = \varTheta^1_{3,2,z-25} = S_{I^0}(2,z-25) \oplus S_{I^0}(4,z-26) \oplus I^0_{3,2,z-25} \\ \varPi^1_{3,2,z} = \varTheta^1_{4,3,z-8} = S_{I^0}(3,z-8) \oplus S_{I^0}(0,z-9) \oplus I^0_{4,3,z-8} \\ \qquad\quad = I^0_{4,3,z-8} \oplus I^0_{4,4,z-8} \oplus 1 \\ \varPi^1_{4,2,z} = \varTheta^1_{0,4,z-18} = S_{I^0}(4,z-18) \oplus S_{I^0}(1,z-19) \oplus I^0_{0,4,z-18} \\ \varPi^1_{2,4,z} = \varTheta^1_{4,2,z-39} = S_{I^0}(3,z-39) \oplus S_{I^0}(0,z-40) \oplus I^0_{4,2,z-39} \\ \qquad\quad = I^0_{4,2,z-39} \oplus I^0_{4,4,z-39} \oplus 1 \\ \varPi^1_{3,4,z} = \varTheta^1_{0,3,z-41} = S_{I^0}(4,z-41) \oplus S_{I^0}(1,z-42) \oplus I^0_{0,3,z-41} \\ \varPi^1_{4,4,z} = \varTheta^1_{1,4,z-2} = S_{I^0}(0,z-2) \oplus S_{I^0}(2,z-3) \oplus I^0_{1,4,z-2} \end{cases}$$

Since the space of S_{I^0} only remains 2^{64}, this quadratic problem cannot be solved through linearization methods. However, the problem can be well solved by exhaustive search (a complete satisfaction for all $c_3(z)$ may not exist). Notice that although $S_{I^0}(1,*)$ is fixed, with two free constant lanes, the random space size still remains 2^{64}, meeting $d_1 \leq s$. Thus attacker can spend a 2^{64} search in (almost) ensuring $c_3(z)$, and the fixed S_{I^0} can support a preimage search of 2^{64}, which is an amortized-$O(1)$ solving algorithm.

Table 4. An example for 2-round Keccak-512 (in big-endian order).

Starting State I^0 (the second message block)				
268c296b1554f13d	000000020cdde05b	00ddc28718198fb9	0d6f443343bbe87b	ffffffffffffffff
268cd56b15540d3d	001cfffdf2c2efa4	fcddc147181e7c46	f29143ccbc45ef84	ffffffffffffffff
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
Target State Π^2 (the first two planes)				
1bdbd867cb38d33d	2f9874f320040d8d	0000000000000000	0000000000000000	0000000000000000
ffffffff00000000	00000000ffffffff	6b7fbd45e6be5596	0a1b0874659065b0	fffffe430000ffff

Table 4 gives a special example that ensures 48 of 64 $c_3(z)$. This particular solving algorithm only works for partial and continuous target bits. The details are excluded in this paper.

The effect of those 192 linear-dependent bit pairs is analyzed below. Suppose we aim to match a 3-bit output row (denoted as $b_0b_1b_2??$) from the second plane, and we are able to arbitrarily set 3 input bits a_0, a_1, a_4. Then we surely choose $a_0a_1??a_4$ that is most likely to match the output. For example, to match 011??, we choose $a_0 = 0$, $a_1 = 1$ and $a_4 = 0$. Under these restrictions, among 4 cases of different a_2a_3, 01100 → 01101 and 01110 → 01111, with matching probability of 1/2. **Table 5** summarizes the most effective restrictions for different outputs. If an output corresponds to multiple choices, the table would only present one.

Table 5. Most effective input restrictions for different 3-bit outputs ($a_0a_1??a_4$ ver).

Output	Input Restrictions	Prob.	Gain
000??	$a_0 = 0\&a_1 = 1$	2/8	2^1
	$a_0 = 0\&a_1 = 0\&a_4 = 0$	1/4	2^1
001??	$a_0 = 1\&a_1 = 0$	3/8	$2^{1.58}$
	$a_0 = 1\&a_1 = 0\&a_4 = 0$	2/4	2^2
010??	$a_0 = 0\&a_1 = 0$	2/8	2^1
	$a_0 = 0\&a_1 = 0\&a_4 = 0$	1/4	2^1
011??	$a_0 = 0\&a_1 = 1$	4/8	2^2
	$a_0 = 0\&a_1 = 1\&a_4 = 0$	2/4	2^2
100??	$a_0 = 1\&a_1 = 1$	2/8	2^1
	$a_0 = 0\&a_1 = 0\&a_4 = 1$	1/4	2^1
101??	$a_0 = 0\&a_1 = 0$	3/8	$2^{1.58}$
	$a_0 = 0\&a_1 = 0\&a_4 = 0$	2/4	2^2
110??	$a_0 = 1\&a_1 = 0$	2/8	2^1
	$a_0 = 1\&a_1 = 0\&a_4 = 0$	1/4	2^1
111??	$a_0 = 1\&a_1 = 1$	4/8	2^2
	$a_0 = 1\&a_1 = 1\&a_4 = 0$	2/4	2^2

From **Table 5** we can see that not all 3-bit outputs require the satisfaction of $c_3(z)$. In average case (8 of 64 of each kind), the attack is required to construct 144 linear-dependent bit pairs, whose total gain is $2^{(1+2+1+2+1+2+1+2)\times 8} = 2^{96}$. In summary, the model parameters of preimage attack on 2-round Keccak-512 are $d_1 = 2^{64}$, $s = 2^{64}$ and $d_2 = 2^{512-192-96+27+1} = 2^{252}$.

5 Preimage Cryptanalysis of 3-Round Keccak-384/512

Improved preimage cryptanalysis of 3-round Keccak-384/512 is presented in this section. For these two versions, we first design 2-round restricted linear structures with 64/128 degrees of freedom left. The latter structure holds with a probability of 2^{-27} to eliminate quadratic terms, while the former structure has not applied non-linear liberalization for extra degrees of freedom due to lack of random space. For 3-round Keccak-384, a basic application of extra linear dependence has been discussed in Section 3.4 with 64 linear-dependent bit pairs. Here we propose an improved application with up to 96 linear-dependent bit pairs, which can bring an increased gain of 2^{50}. As for 3-round Keccak-512, because of the diffusion of quadratic bits in the third θ, the linear structure cannot reach Π^3. Yet we can spend degrees of freedom in constructing 128 linear-dependent bit pairs, which can bring a gain of 2^{114} (the average case). Finally, the complexity of preimage attacks on 3-round Keccak-384/512 is 2^{271} and 2^{426} respectively.

5.1 Improved Preimage Attack on 3-Round Keccak-384

The restricted linear structure for 3-round Keccak-384 is given in **Fig. 10**.

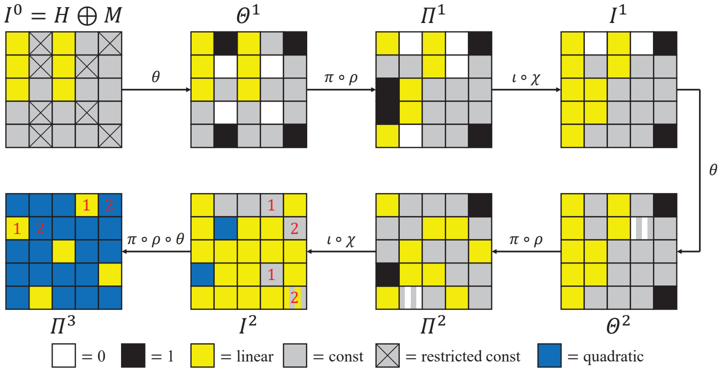

Fig. 10. Restricted linear structure for 3-round Keccak-384.

This structure requires 320 column sum equations and 320 I^0 restrictions as below, leaving 64 degrees of freedom. Since M cannot ensure $I^0_{1,3,z} \oplus 1 = I^0_{1,4,z}$, the attack requires a 2-block model with $d_1 = 2^{64}$.

$$\begin{cases} I^0_{1,0,z} = I^0_{1,1,z} \oplus 1 = I^0_{1,3,z} \oplus 1 = I^0_{1,4,z} \\ I^0_{3,1,z} = I^0_{3,3,z} \\ I^0_{4,0,z} = I^0_{4,4,z} \\ I^0_{0,0,z} \oplus I^0_{0,1,z} \oplus I^0_{0,2,z} = S_{I^0}(0,z) \oplus I^0_{0,3,z} \oplus I^0_{0,4,z} \\ I^0_{2,0,z} \oplus I^0_{2,1,z} \oplus I^0_{2,2,z} = S_{I^0}(2,z) \oplus I^0_{2,3,z} \oplus I^0_{2,4,z} \\ I^1_{0,0,z} \oplus I^1_{0,1,z} \oplus I^1_{0,2,z} \oplus I^1_{0,3,z} \oplus I^1_{0,4,z} = S_{I^1}(0,z) \\ I^1_{1,2,z} \oplus I^1_{1,3,z} = S_{I^1}(1,z) \oplus I^1_{1,0,z} \oplus I^1_{1,1,z} \oplus I^1_{1,4,z} \\ I^1_{2,0,z} \oplus I^1_{2,1,z} = S_{I^1}(2,z) \oplus I^1_{2,2,z} \oplus I^1_{2,3,z} \oplus I^1_{2,4,z} \end{cases}$$

In the equation system above, $S_{I^0}(0,*)$ and $S_{I^0}(2,*)$ can be decided by:

$$\begin{cases} S_{I^0}(0,z) \oplus S_{I^0}(2,z-1) = I^0_{1,3,z} \\ S_{I^0}(2,z) \oplus S_{I^0}(4,z-1) = I^0_{3,3,z} \\ S_{I^0}(3,z) \oplus S_{I^0}(0,z-1) = I^0_{4,4,z} \oplus 1 \end{cases}$$

where $S_{I^0}(3,*)$ and $S_{I^0}(4,*)$ are all controllable with corresponding free constant lanes. Counting controllable $S_{I^0}(1,*)$ in, setting S_{I^0} can generate a random space of $s = 2^{128}$, enough to ensure $d_1 \leq s$. Therefore, we can even fix all controllable S_{I^1} to apply extra linear dependence.

The target linear-dependent pairs are (as marked by red numbers in **Fig. 10**):

$$\begin{cases} (I^2_{3,0,z} \to \varPi^3_{0,1,z+28}) \oplus (I^2_{3,3,z} \to \varPi^3_{3,0,z+21}) = c_1(z) \\ (I^2_{4,1,z} \to \varPi^3_{1,1,z+20}) \oplus (I^2_{4,4,z} \to \varPi^3_{4,0,z+14}) = c_2(z) \end{cases}$$

where the constant attributes of $I^2_{3,0,z}$ and $I^2_{4,4,z}$ require $\varPi^2_{4,0,z} = 1$ and $\varPi^2_{1,4,z} = 0$, and the constant attributes of $I^2_{3,3,z}$ and $I^2_{4,1,z}$ have been ensured.

It is worth emphasizing that although we cannot set any target equations on $\varPi^3_{4,0,*}$ (quadratic), the linear dependence between $\varPi^3_{1,1,z+20}$ and $\varPi^3_{4,0,z+14}$ still helps the search of preimage. Suppose $\varPi^3_{4,0,*}$ randomly matches the digest with a probability of 2^{-64}. Under proper linear relations, $\varPi^3_{1,1,*}$ can simultaneously match. The gain of those 64 linear-dependent pairs of c_2 part is $2^{0.42\times 64} = 2^{26}$.

The satisfaction of $c_1(z)$ has been discussed in Section 3.4. By fixing $\varPi^2_{0,3,*}$ and $\varPi^2_{4,0,*}$ to be all '1', the target relations can be linearized to S_{I^1} restrictions as below:

$$\begin{cases} S_{I^1}(0,z) = S_{I^1}(3,z+1) \\ S_{I^1}(1,z) = S_{I^1}(2,z-6) \oplus S_{I^1}(2,z-41) \oplus S_{I^1}(3,z-1) \oplus S_{I^1}(4,z-7) \\ \qquad\qquad \oplus S_{I^1}(4,z-42) \oplus c_1(z+15) \oplus I^1_{2,3,z} \oplus I^1_{3,3,z-6} \oplus I^1_{3,4,z-41} \oplus 1 \end{cases}$$

Then $S_{I^1}(0,*)$ has been decided and $S_{I^1}(1,*)$ depends on $S_{I^1}(2,*)$. Similarly, from the perspective of $S_{I^1}(2,*)$, ensuring $c_2(z)$ becomes a quadratic problem. However, this time we have to adopt linearization methods because the constant attribute of $I^2_{4,4,z}$ requires $\varPi^2_{1,4,z} = 0$. Related formulas are given below:

$$\begin{cases} I^2_{4,1,z} &= \Pi^2_{4,1,z} \oplus (\Pi^2_{0,1,z} \oplus 1) \cdot \Pi^2_{1,1,z} \\ I^2_{4,4,z} &= \Pi^2_{4,4,z} \oplus (\Pi^2_{0,4,z} \oplus 1) \cdot \Pi^2_{1,4,z} = \Pi^2_{4,4,z} \\ \Pi^2_{1,4,z} &= \Theta^2_{3,1,z-55} = S_{I^1}(2, z-55) \oplus S_{I^1}(4, z-56) \oplus I^1_{3,1,z-55} = 0 \\ \Pi^2_{4,1,z} &= \Theta^2_{2,4,z-61} = S_{I^1}(1, z-61) \oplus S_{I^1}(3, z-62) \oplus I^1_{2,4,z-61} \\ &= S_{I^1}(2, z-3) \oplus S_{I^1}(2, z-38) \oplus \cdots \\ \Pi^2_{0,1,z} &= \Theta^2_{3,0,z-28} = S_{I^1}(2, z-28) \oplus S_{I^1}(4, z-29) \oplus I^1_{3,0,z-28} \\ \Pi^2_{1,1,z} &= \Theta^2_{4,1,z-20} = S_{I^1}(3, z-20) \oplus S_{I^1}(0, z-21) \oplus I^1_{4,1,z-20} \\ &= I^1_{4,1,z-20} \\ \Pi^2_{4,4,z} &= \Theta^2_{1,4,z-2} = S_{I^1}(0, z-2) \oplus S_{I^1}(2, z-3) \oplus I^1_{1,4,z-2} \end{cases}$$

A basic idea is, choosing 32 $S_{I^1}(2, z)$ to fix $\Pi^2_{1,4,z+55} = 0$, and the remaining 32 can ensure corresponding $I^2_{4,1,z} \oplus I^2_{4,4,z} = c_2(z)$. But one problem is that the chosen set may decide some $I^2_{4,1,z} \oplus I^2_{4,4,z}$ and cause contradictions. For example, suppose a certain $\Pi^2_{1,1,z} = 0$. Then $I^2_{4,1,z} \oplus I^2_{4,4,z} = \Pi^2_{4,1,z} \oplus \Pi^2_{4,4,z}$ is decided by $S_{I^1}(2, z-38)$ (notice that the term of $S_{I^1}(2, z-3)$ is always eliminated). If this column sum has been fixed, the choosing algorithm may fail to ensure $c_2(z)$.

Since ensuring a certain $c_2(z)$ requires a fixed $S_{I^1}(2, z-55)$ firstly, the core of any choosing algorithm is: the chosen set cannot contain any pair of $S_{I^1}(2, z)$ and $S_{I^1}(2, z-17)$. It is easily known that the set of odd z (or even z) meets the restriction. An example is given in **Table 6**.

Table 6. An example for 3-round Keccak-384 (in big-endian order).

Starting State I^0 (the second message block)				
29f2a8022c6bc4f1	ffffffffffffffff	c4a1d57c66425fb8	0000000000000000	0000000000000000
945ff683ef81a471	0000000000000000	ad7bda1596b6ed4e	0000000000000000	ffffffffffffffff
4252a17e3c159f7f	a913e08d6cdfe4f7	9625f0960f0b4d09	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	ffffffffffffffff	0000000000000000	0000000000000000	0000000000000000
Target State Π^3 (the first two planes)				
c131d81cb43192e3	668d5e436102ac54	5e131b071ee3f03d	0000000000000000	f4436871e72b9eb5
ffffffff00000000	e587e304b5187222	2a80f3ffd586b8d5	b0fcf6d2cf772225	9f1b23d55c71c2f7

(f4436871e72b9eb5$\lll$ 6)$\oplus$e587e304b5187222=f55dff7d7fffdf5f with even sites all ‘1’.

In summary, by fully spending the random space of identical restricted linear structure, at most 96 linear-dependent bit pairs from the output planes can be constructed in improved preimage attack on 3-round Keccak-384. The total gain of those linear-dependent bit pairs is $2^{37+26/2} = 2^{50}$. And the model parameters of this attack are $d_1 = 2^{64}$, $s = 2^{128}$ and $d_2 = 2^{384-64-50+1} = 2^{271}$.

5.2 Improved Preimage Attack on 3-Round Keccak-512

The restricted linear structure for 3-round Keccak-512 is given in **Fig. 11**.

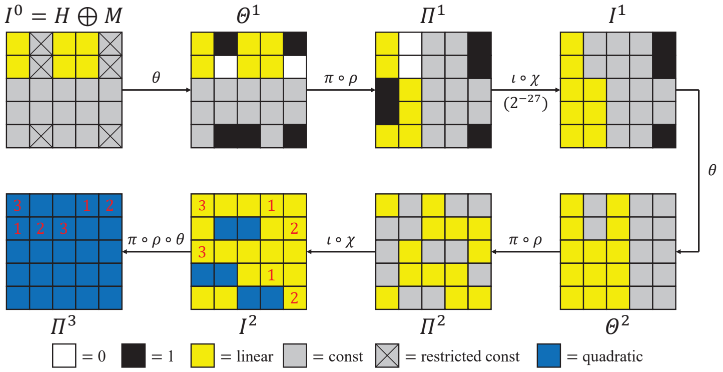

Fig. 11. Restricted linear structure for 3-round Keccak-512.

This structure requires 256 column sum equations and 256 I^0 restrictions as below, leaving 128 degrees of freedom. Since M cannot affect $I^0_{4,1,*}$, the attack requires a 2-block model with $d_1 = 2^{64}$.

$$\begin{cases} I^0_{1,0,z} = I^0_{1,1,z} \oplus 1 = I^0_{1,4,z} \\ I^0_{4,0,z} = I^0_{4,1,z} \oplus 1 = I^0_{4,4,z} \\ I^0_{0,0,z} \oplus I^0_{0,1,z} = S_{I^0}(0,z) \oplus I^0_{0,2,z} \oplus I^0_{0,3,z} \oplus I^0_{0,4,z} \\ I^0_{2,0,z} \oplus I^0_{2,1,z} = S_{I^0}(2,z) \oplus I^0_{2,2,z} \oplus I^0_{2,3,z} \oplus I^0_{2,4,z} \\ I^0_{3,0,z} \oplus I^0_{3,1,z} = S_{I^0}(3,z) \oplus I^0_{3,2,z} \oplus I^0_{3,3,z} \oplus I^0_{3,4,z} \\ I^1_{0,0,z} \oplus I^1_{0,1,z} \oplus I^1_{0,2,z} \oplus I^1_{0,3,z} \oplus I^1_{0,4,z} = S_{I^1}(0,z) \end{cases}$$

In the equation system above, $S_{I^0}(0,*)$, $S_{I^0}(2,*)$ and $S_{I^0}(3,*)$ are fixed by:

$$\begin{cases} S_{I^0}(0,z) \oplus S_{I^0}(2,z-1) = I^0_{1,4,z} \oplus 1 \\ S_{I^0}(1,z) \oplus S_{I^0}(3,z-1) = I^0_{2,4,z} \oplus 1 \\ S_{I^0}(3,z) \oplus S_{I^0}(0,z-1) = I^0_{4,4,z} \oplus 1 \end{cases}$$

since $S_{I^0}(1,*)$ has been fixed by I^0 restrictions. In other words, setting S_{I^0} will not generate any random space. The random space of this structure is entirely provided by controllable $S_{I^1}(0,*)$ with $s = 2^{64}$, meeting $d_1 \leq s$.

It is easily found that this structure abandons the column sum equations of $I^1_{1,1,z} \oplus I^1_{1,2,z} \oplus I^1_{1,3,z} = S_{I^1}(1,z) \oplus I^1_{1,0,z} \oplus I^1_{1,4,z}$ for extra degrees of freedom. Then without these restrictions, variables spread to $\Theta^1_{2,*,*}$ and generate several quadratic lanes in Π^2. Finally quadratic bits pollute the whole Π^3 through the third θ. Therefore, in this structure we cannot spend degrees of freedom in fixing $\Pi^3_{*,0,*}$, but spend degrees of freedom in ensuring the linear-dependent bit pairs from the output planes. The target linear-dependent pairs are (as marked by red numbers in **Fig. 11**):

$$\begin{cases} (I^2_{3,0,z} \to \Pi^3_{0,1,z+28}) \oplus (I^2_{3,3,z} \to \Pi^3_{3,0,z+21}) = c_1(z) \\ (I^2_{4,1,z} \to \Pi^3_{1,1,z+20}) \oplus (I^2_{4,4,z} \to \Pi^3_{4,0,z+14}) = c_2(z) \\ (I^2_{0,0,z} \to \Pi^3_{0,0,z}) \oplus (I^2_{0,2,z} \to \Pi^3_{2,1,z+3}) = c_3(z) \end{cases}$$

Since the number of degrees of freedom left is 128, attacker can only choose any 128 of them. An example is given in **Table 7**.

Table 7. An example for 3-round Keccak-512 (in big-endian order).

Starting State I^0 (the second message block)				
a26df3be8c43aa78	0000000000000000	71a5148504cdf108	f7efffbbe7fbed7e	0000000000000000
5d920c4173bc5587	ffffffffffffffff	71a5148504cdf108	f7efffbbe7fbed7e	ffffffffffffffff
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
Target State Π^3 (the first two planes)				
3327f7cc992a672d	3ec07f37d73be269	4636786d13ae0854	04d45037d9c52357	f25c21711cd4d327
6a281bece291ab82	97085c693116aafc	7afd317cc953396b	ac29a826a456e7f0	4b16e8378663a85e

3327f7cc992a672d$\oplus$(7afd317cc953396b$\lll$ 61) =5c7851e300000000

04d45037d9c52357$\oplus$(6a281bece291ab82$\lll$ 57) =0000000000000000

f25c21711cd4d327$\oplus$(97085c693116aafc$\lll$ 58) =00000000b810898c

Yet an unsolved problem is, how should attacker choose the linear-dependent bit pairs. Actually attacker can choose not only "bit" but also "combination". For example, suppose a target 3-bit output row is 000??. Then according to the inversion property stated in Section 2.4, $a_0 \oplus a_2 = 0$ is inversed from the output. To satisfy this combination, attacker can spend 1 degree of freedom in setting $I^2_{0,0,z-3} \oplus I^2_{0,2,z-3} \oplus I^2_{3,0,z-28} \oplus I^2_{3,3,z-28} = \Pi^3_{0,0,z-3} \oplus \Pi^3_{3,0,z-7}$, where $\Pi^3_{0,0,z-3}$ and $\Pi^3_{3,0,z-7}$ are decided by the digest. If $\Pi^3_{0,0,z-3}$ and $\Pi^3_{3,0,z-7}$ (quadratic) match the digest with a probability of 2^{-2}, $\Pi^3_{0,1,z} \oplus \Pi^3_{2,1,z} = 0$ can be simultaneously ensured with a gain of 2^1. It is concluded that attacker should spend degrees of freedom on those inversed parts firstly with a gain of 2^1 each, and then spend remaining degrees of freedom on extra restrictions.

Table 8 summarizes the effect of inversed equations and extra restrictions for different 3-bit outputs. Each output contains two lines of content. The first line is about linear equations (among $a_0a_1a_2$) inversed from the output. Each of them can bring a gain of 2^1. The second line is about the most effective choices of the 3 input bits. Unlike **Table 5**, this table presents the only choice.

Table 8. Most effective input restrictions for different 3-bit outputs ($a_0a_1a_2$?? ver).

Output	Input Restrictions	Prob.	Gain
000??	$a_0 \oplus a_2 = 0$	4/16	2^1
	$a_0 = 0 \& a_1 = 1 \& a_2 = 0$	2/4	2^2
001??	$a_0 \oplus a_2 = 0 \& a_1 = 0$	4/8	2^2
	$a_0 = 1 \& a_1 = 0 \& a_2 = 1$	3/4	$2^{2.58}$
010??	$a_0 = 0$	4/16	2^1
	$a_0 = 0 \& a_1 = 0 \& a_2 = 0$	2/4	2^2
011??	$a_0 = 0 \& a_1 = 1$	4/8	2^2
	$a_0 = 0 \& a_1 = 1 \& a_2 = 1$	3/4	$2^{2.58}$
100??	$a_0 \oplus a_2 = 1$	4/16	2^1
	$a_0 = 1 \& a_1 = 1 \& a_2 = 0$	2/4	2^2
101??	$a_0 \oplus a_2 = 1 \& a_1 = 0$	4/8	2^2
	$a_0 = 0 \& a_1 = 0 \& a_2 = 1$	3/4	$2^{2.58}$
110??	$a_0 = 1$	4/16	2^1
	$a_0 = 1 \& a_1 = 0 \& a_2 = 0$	2/4	2^2
111??	$a_0 = 1 \& a_1 = 1$	4/8	2^2
	$a_0 = 1 \& a_1 = 1 \& a_2 = 1$	3/4	$2^{2.58}$

From **Table 8** we can see that the second output plane is expected to inverse 96 linear equations in average case (8 of 64 of each kind). Counting remaining extra restrictions in, those 128 linear-dependent bit pairs can bring a total gain of $2^{96+32\times 0.58} = 2^{114}$. In summary, the model parameters of preimage attack on 3-round Keccak-512 are $d_1 = 2^{64}$, $s = 2^{64}$ and $d_2 = 2^{512-114+27+1} = 2^{426}$.

6 Conclusion

This paper provides improved preimage cryptanalysis on round-reduced Keccak-384/512. The core of our preimage attacks is linear analysis. We inherit the ideas of linear structure and allocating model from previous works, and improve the preimage cryptanalysis results in two aspects:

1. We adopt improved linear structures with proper non-linear liberalization that can bring extra degrees of freedom and enlarge the random space.

2. We construct lane-level extra linear dependence between two output planes without spending degrees of freedom (by restricting the random space instead), and those linear relations can much decrease the complexity of preimage search.

As a result, the complexity of preimage attacks on 2-round Keccak-384/512 and 3-round Keccak-384/512 is decreased to $2^{28}/2^{252}$ and $2^{271}/2^{426}$ respectively.

It is noted that our attack algorithm is still far from threatening the security of full-round Keccak. It seems that methods based on linear analysis can hardly threaten the preimage resistance of even 5-round Keccak: Rajasree has proposed a linear attack for 4-round Keccak-384 in [17], while linear analysis for 4-round

Keccak-512 is still an open problem. However, although this paper only focuses on linear analysis, the idea of constructing linear dependence between the output planes might be able to combine with non-linear technologies.

References

1. Aumasson, J.P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. NIST Mailing List (2009). https://131002.net/data/papers/AM09.pdf
2. Bernstein, D.J.: Second preimages for 6 (7? (8??)) rounds of Keccak? NIST Mailing List (2010). https://ehash.iaik.tugraz.at/uploads/6/65/NIST-mailing-list_Bernstein-Daemen.txt
3. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Cryptographic sponge functions. Submission to NIST (Round 3) (2011). https://sponge.noekeon.org/CSF-0.1.pdf
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference, version 3.0. Submission to NIST (Round 3) (2011). https://keccak.noekeon.org/Keccak-reference-3.0.pdf
5. Dinur, I., Dunkelman, O., Shamir, A.: New attacks on Keccak-224 and Keccak-256. In: FSE 2012. LNCS, vol. 7549, pp. 442–461. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34047-5_25
6. 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, Part I. LNCS, vol. 9056, pp. 733–761. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_28
7. Duan, M., Lai, X.: Improved zero-sum distinguisher for full round Keccak-f permutation. Cryptology ePrint Archive, Report 2011/023 (2011). https://eprint.iacr.org/2011/023
8. Guo, J., Liu, M., Song, L.: Linear structures: applications to cryptanalysis of round-reduced Keccak. In: ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 249–274. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_9
9. He, L., Lin, X. Yu, H.: Improved preimage attacks on 4-round Keccak-224/256. IACR Transactions on Symmetric Cryptology 2021(1), 217–238 (2021). https://doi.org/10.46586/tosc.v2021.i1.217-238
10. Huang, S., Wang, X., Xu, G., Wang, M., Zhao, J.: Conditional cube attack on reduced-round Keccak sponge function. In: EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 259–288. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-56614-6_9
11. Kumar, R., Mittal, N., Singh, S.: Cryptanalysis of 2 round Keccak-384. In: Progress in Cryptology – INDOCRYPT 2018. LNCS, vol. 11356, pp. 120–133. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-030-05378-9_7
12. Kumar, R., Rajasree, M.S., AlKhzaimi, H.: Cryptanalysis of 1-round KECCAK. In: Progress in Cryptology – AFRICACRYPT 2018. LNCS, vol. 10831, pp. 124–137. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-89339-6_8
13. Li, T., Sun, Y.: Preimage attacks on round-reduced Keccak-224/256 via an allocating approach. In: EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 556–584. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-17659-4_19
14. Morawiecki, P., Pieprzyk, J., Srebrny, M.: Rotational cryptanalysis of round-reduced Keccak. In: FSE 2013. LNCS, vol. 8424, pp. 241–262. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-662-43933-3_13

15. Morawiecki, P., Srebrny, M.: A SAT-based preimage analysis of reduced KECCAK hash functions. Information Processing Letters 113(10-11), 392–397 (2013). `https://doi.org/10.1016/j.ipl.2013.03.004`
16. Qiao, K., Song, L., Liu, M., Guo, J.: New collision attacks on round-reduced Keccak. In: EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 216–243. Springer, Heidelberg (2017). `https://doi.org/10.1007/978-3-319-56617-7_8`
17. Rajasree, M.S.: Cryptanalysis of round-reduced KECCAK using non-linear structures. In: Progress in Cryptology – INDOCRYPT 2019. LNCS, vol. 11898, pp. 175–192. Springer, Heidelberg (2019). `https://doi.org/10.1007/978-3-030-35423-7_9`
18. Song, L., Guo, J., Shi, D., Ling, S.: New MILP modeling: improved conditional cube attacks on Keccak-based constructions. In: ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 65–95. Springer, Heidelberg (2018). `https://doi.org/10.1007/978-3-030-03329-3_3`
19. Song, L., Liao, G., Guo, J.: Non-full sbox linearization: applications to collision attacks on round-reduced Keccak. In: CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 428–451. Springer, Heidelberg (2017). `https://doi.org/10.1007/978-3-319-63715-0_15`
20. The U.S. National Institute of Standards and Technology: SHA-3 standard: permutation-based hash and extendable-Output functions. Federal Information Processing Standard, FIPS 202 (2015). `http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf`