keccak preimage

May 27, 2019

1 Introduction

The observations based on the following paper : **Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak**.

1.1 2R Keccak-512

See Fig. 8, for each guess : we set

$$A[0,1] = A[0,0] \oplus \alpha_0$$

$$A[2,1] = A[2,0] \oplus \alpha_2$$

with α_0 and α_2 as random constants

Since $A[0,0]$ and $A[2,0]$ have 128 bits. So we have a complexity gain over bruteforce of 2^{128}. Hence the time complexity $= 2^{512-128} = 2^{384}$.

Input degree of freedom :

1. 64 bits from $A[0,0]$, 64 bits from $A[2,0]$.
2. (320 -1) bits from white lanes
3. 128 bits from α_0, α_2

This sums upto 575 bits larger than required 512 bits.

1.2 2R Keccak-384

1. Attack similar to Keccak-512
2. Can obtain a linear structure of 256 bits variables from

$$(A[0,0], A[0,1], A[2,0], A[2,1])$$

3. with : $A[0,2] = A[0,0] \oplus A[0,1] \oplus \alpha_0$
4. and $A[2,2] = A[2,0] \oplus A[2,1] \oplus \alpha_2$

5. Hence a linear system of 256-bit equations
6. Msg satisfying padding rule, need a solution with the last bit of A[2,2] being 1
7. Time Complexity of Attack = $2^{384-256+1} = 2^{129}$

Note : In the following document WM refers to Willi Meier

2 Suggestion by WM

2.1 Mail : 5 Feb

Mail contents are :

> I mainly refer to : https://eprint.iacr.org/2016/878
>
> We try to loosen restrictions imposed by linear structures to increase freedom degrees.
>
> An example is 2-round Keccak-384. I refer to Fig. 9. In addition to the colored 6 lanes that are variable, we also keep lanes 3,0 and 3,1 variable, still so that sum of columns is kept constant. Then χ produces one quadratic lane. We substitute quadratic monomials by 64 linear variables.
>
> We can invert the first row of image (green), and we require that the map to the 6-th green lane is linear (i.e. quadratic part happens to vanish). The probability for this event is about $.75^{64} = 2^{27}$
>
> Then we have $4*64+64+64 = 384$ variables and the same number of conditions. Solving this linear system has about 1 solution, that we may check by substituting it into 2-round Keccak. Genuine number of freedom degrees is 320, and artficial ones 384. It is likely to find a correct solution in $2^{64} * 2^{27} = 2^{91}$ trials. This case was found in discussion with Meicheng Liu and Jian Guo. I believe that something similar (or better) will work for 2-round Keccak-512. In addition to substitution of a quadratic lane by linear variables, we know that variables in same column are linearly related. This means that in quadratic lanes after χ, some linear factors are essentially "the same".
>
> The probability of a1*a2 =a1*a3 is 0.75 for any values a1, a2, a3. We substitute monomials of the form a1*a2 and a1*a3 by the same linear variable to get more linear equations that will hold true with reasonable probability. We introduce artificial linear variables economically, so that we don't have more linear variables (genuine and artificial ones) than equations.Details need to be checked.
>
> Wonder whether this could also help, e.g., in 3-round Keccak-256 in `https://tosc.iacr.org/index.php/ToSC/article/view/802`.

An issue is how far these observations extend to improve 3 round preimages of Keccak-384 and Keccak-512, as described in Sect. 6.3.

As complexities of known preimages of 4-round Keccak-512 are still close to 2^{512}, there is some slight hope that we might reach 4 rounds for this case.

Do you have any comments on this?

1. Here in addition to Keccak-384 setting for 2 Round attack.
2. In Fig. 9 we keep lanes (3, 0) and (3, 1) also as variables
3. $A[3,1] = A[3,0] \oplus \alpha_3$ (so that sum of columns remains constant)
4. Linear structure of 320-bit :
5. $(A[0,0], A[0,1], A[2,0], A[2,1], A[3,0])$
6. A quadratic term is introduced after 1st χ by product of $(2,0), (3,1)$ as both are variables.
7. For this we substitute quadratic monomials by 64 linear variables.
8. Now next problem was for last round : $(\chi o \iota)^{-1}$
9. For the green lane of second row, this happens to be linear with probability $= 0.75$
10. For 64 slices, this $= 0.75^{64}$, which adds 2^{27} trials
11. Complexity comes out to be $= 2^{64} * 2^{27} = 2^{91}$ trials.
12. **Comments :**
    (a) Any issue of Linear independent variable for solution of equations?
    (b) Otherwise the analysis is fine here.

2.2 Suggestion for Keccak-384, 2R

1. **Note :** After verification this comes out to be incorrect
2. Same as the setting introduced by WM for 2R Keccak-384 where we keep (3,0) and (3,1) also as variables
3. Here we introduced a new variable for each quadratic term
4. The suggestion here is for reducing the trials due to the last round linear χ inversion.
5. Instead of inverting χ for 6th green lane as suggested by WM we can do something like this :

6. **Observation :** When only one output bit is known after χ step, then the corresponding input bits have 2^4 possibilities. A way to fix the first output bit to be the same as input bit and the second bit as 1. It is shown in the Figure 2.

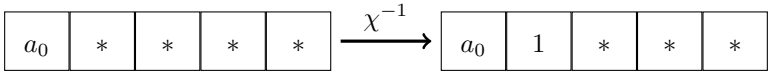

Figure 1: Computation of χ^{-1}

7. Here in Fig. 2, we fix the lane adjacent to 6th green lane as 1 before χ operation. If this is assumed then the 6th lane is inverted as it is and seems like a constant.
8. No. of variables : 5 lane variables + 1(substitute for quadratic terms) and 6 linear conditions. So we Linear structure : $(A[0,0], A[0,1], A[2,0], A[2,1], A[3,0])$
9. White lane (constants) : 5 lanes and $64 * 3$ bits from $\alpha_0, \alpha_2, \alpha_3$
10. So the degree of freedom is much larger than required 384 bits.
11. We just want a solution with last bit of $A[2,2]$ being 1.
12. Time complexity = $2^{384-(64*5)+1} = 2^{65}$
13. If this turns out to be correct then we have a much better attack than proposed by us in our indocrypt-2018 paper with complexity of 2^{88}
14. Please verify/correct this. If this is correct then something similar can also be applied to Keccak-512.

2.3 WM mail : 08/02/19

For 2R Keccak-512 :

1. 1st row of hashes can be inverted
2. For 2nd row, we know 3 consecutive outputs. So as in paper (Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak)
3. Using table 4, we know that for 3 consecutive output bits in a row known we get 2 linear equations.
4. For 64 lane size, we get $2 * 64 = 128$ equations.
5. For the Message : we keep columns 0, 1, 2 and 3 as variables

6.
$$A[0,1] = A[0,0] \oplus \alpha_0$$
$$A[1,1] = A[1,0] \oplus \alpha_1$$
$$A[2,1] = A[2,0] \oplus \alpha_2$$
$$A[3,1] = A[3,0] \oplus \alpha_3$$

7. This state produces quadratic variables after $\pi \circ \rho \circ \theta$.
8. This creates 3 quadratic lanes after χ of 1st round, caused by product of [0,0] with [1,1], and [1,0] with [2,1], and [2,0] with [3,1].
9. Substitute each of the above quadratic variable by a new linear variable.
10. So we need $3 * 64$ new linear variables
11. Degree of freedom :
    (a) Linear structure : $(A[0,0], A[1,0], A[2,0], A[3,0])$ i.e. $4 * 64$ variables
    (b) Artificial : $3 * 64$ linear variables
    (c) Adds upto overall : $7 * 64$
12. No. of linear conditions :
    (a) 5 from inversion of chi from the first row of hashes
    (b) 2 from inversion of chi from 3 consecutive output bits of hash
    (c) Hence, overall 7 equations
13. We have 7 linear variables and 7 equations so we can expect to find a solution.
14. But Actual degree of freedom : 4 i.e. $(A[0,0], A[1,0], A[2,0], A[3,0])$ and 8 conditions on the final hash (8 lanes)
15. So, no. of trials $= 2^{512-4*64} = 2^{64*4} = 2^{256}$

2.4 Suggestion for Keccak-384, 2R

1. Same as the setting introduced by WM for 2R Keccak-384 where we keep (4,0) and (4,1) also as variables
2. Here we introduced two new variable for the quadratic terms.
3. 2 quadratic lanes after χ of 1st round, caused by product of (4,0), (0,1) and (4,1), (0,2).
4. The suggestion here is for reducing the trials due to the last round linear χ inversion.

5. Instead of inverting χ for 6th green lane as suggested by WM we can do something like this :

6. **Observation :** When only one output bit is known after χ step, then the corresponding input bits have 2^4 possibilities. A way to fix the first output bit to be the same as input bit and the second bit as 1. It is shown in the Figure 2.

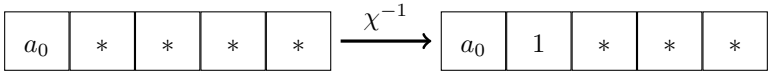

Figure 2: Computation of χ^{-1}

7. Here in Fig. 2, we fix the lane adjacent to 6th green lane as 1 before χ operation. If this is assumed then the 6th lane is inverted as it is and seems like a constant.

8. No. of variables : 5 lane variables + 2(substitute for quadratic terms) and 7 (6 linear + 1 constant) conditions. So we have Linear structure : $(A[0,0], A[0,1], A[2,0], A[2,1], A[4,0])$

9. White lane (constants) : 5 lanes and $64 * 3$ bits from $\alpha_0, \alpha_2, \alpha_4$

10. So the degree of freedom is much larger than required 384 bits.

11. We just want a solution with last bit of $A[2,2]$ being 1.

12. Time complexity = $2^{384-(64*5)+1} = 2^{65}$

13. If this turns out to be correct then we have a much better attack than proposed by us in our indocrypt-2018 paper with complexity of 2^{88}

14. We indeed have 7 variable lanes and 7 conditions.

15. However, two of them are now artificial, and genuine number of freedom degrees is still 5. Thus we need about 2^128 trials to satisfy the 7 conditions. Hence this doesn't work.

16. Complexity here is = $2^{65*\text{number\_artificial\_degree}}$

17. But still this is our own solution for the complexity 2^{130}.

2.5 Keccak-512, 1R

1. The idea is based upon linear structures though they are used for more no. of rounds but I will be using them for 1 Round.

2. Here we linearize Keccak-f permutation for 1 round.

3. The Preimage attack using linear structures depends directly on the space size of the variables of the linear structures formed below.
4. d = No. of output bits and $c = 2 * d$
5. $d = 512$, $c = 1024$
6. $r = 1600 - 1025 = 576$
7. For the Message : we keep columns 0, 1, 2 and 3 as variables
8. $$A[0,1] = A[0,0] \oplus \alpha_0$$
$$A[1,1] = A[1,0] \oplus \alpha_1$$
$$A[2,1] = A[2,0] \oplus \alpha_2$$
$$A[3,1] = A[3,0] \oplus \alpha_3$$
9. Keep A[4,0] as constant
10. Here $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ are random constants
11. After θ the state is only affected by constants.
12. The ρ step only rotation in lane. [Still the variables remain same]
13. After π step, the variables namely $(A[0,0], A[1,0], A[2,0], A[3,0])$ are shifted to different lanes.
14. This was going forward. We have a linear structure after the above steps.
15. Now going backward, invert ι from the hash.
16. Invert the first 320 bits of hash by χ^{-1} i.e. the first row of hashes.
17. We have $4 * 64$ free variables.
18. So, we have a complexity gain over bruteforce of 2^{4*64}, i.e., $2^{512-256} = 2^{256}$.
19. Attack complexity of 2^{256} for 1R, Keccak-512.
20. Verification : The Degree of freedom should be sufficient to expect a solution.
21. Degree of freedom : $4 * 64$ (from $A[0,0], A[1,0], A[2,0], A[3,0]$) and $4 * 64$ (from $\alpha_0, \alpha_1, \alpha_2, \alpha_3$) and $1 * 64$ (from $A[4,0]$ constant lane)
22. This sums up to be $9 * 64 = 576$, larger than required 512.
23. Note : More variable bits will result in lower attack complexity.
24. This method works for all possible hash values.

25. Present attack for 1R, Keccak-512 is `https://eprint.iacr.org/2017/1028.pdf`.
26. The best attack complexity for the above is 2^{191}.
27. Note : Try to improve the above complexity.

2.6 Keccak-384, 1R

1. The idea is based upon linear structures though they are used for more no. of rounds but I will be using them for 1 Round.
2. Here we linearize Keccak-f permutation for 1 round.
3. The Preimage attack using linear structures depends directly on the space size of the variables of the linear structures formed below.
4. d = No. of output bits and $c = 2 * d$
5. $d = 384$, $c = 768$
6. $r = 1600 - 768 = 832$
7. For the Message : we keep columns 0, 2 and 4 as variables
8. $$A[0,2] = A[0,0] \oplus A[0,1] \oplus \alpha_0$$
$$A[2,1] = A[2,0] \oplus A[2,1] \oplus \alpha_2$$
$$A[4,1] = A[4,0] \oplus \alpha_4$$
9. Keep column 1, 3 as constant
10. Here $\alpha_0, \alpha_2, \alpha_4$ are random constants
11. After θ the state is only affected by constants.
12. The ρ step only rotation in lane. [Still the variables remain same]
13. After π step, the variables namely $(A[0,0], A[0,1], A[2,0], A[2,1], A[4,0])$ are shifted to different lanes.
14. This was going forward. We have a linear structure after the above steps.
15. Now going backward, invert ι from the hash.
16. Invert the first 320 bits of hash by χ^{-1} i.e. the first row of hashes.
17. We have $5 * 64$ free variables.
18. So, we have a complexity gain over bruteforce of 2^{5*64}, i.e., $2^{384-5*64} = 2^{64}$.
19. Hence, a linear system of 320-bit equations.

20. For generating a message satisfying the padding rule, we just need a solution with the last bit of A[2,2] being 1.

21. Attack complexity of $2^{64+1} = 2^{65}$ for 1R, Keccak-384.

22. Verification : The Degree of freedom should be sufficient to expect a solution.

23. Degree of freedom : $5*64$ (from $A[0,0], A[1,0], A[2,0], A[2,1], A[4,0]$) and $3*64$ (from $\alpha_0, \alpha_2, \alpha_4$) and many from constant lanes

24. This sums up to be larger than required 384.

25. This method works for all possible hash values.

26. Don't know the state of the art for 1R, keccak-384.

2.7 Preimage attack on 2-round Keccak-256

1. **Note :** This subsection contains my explaination for the attack mentioned in section 6.1 from paper : Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak.

2. The message here is in lanes (0, 0), (0, 1), (0, 2), (2, 0), (2, 1), (2, 2).

3. Keep the sum of variables in column 0, 2 constant by choosing the sum of variables in a column to be α_0 and α_2 resp as constants.

4. $d = 256 \rightarrow 4$ lanes.

5. $c = 512 \rightarrow 8$ lanes.

6. We can get 4 linear equations on the input bits given 4 output bits of the 5-bits.

7. Therefore, we need 4 variables in our state to build a linear system of 256-bit equation.

8. We have h_0, h_1, h_2, h_3 hash lanes in the output.

9. By using property of χ, we can get 4 linear equations on the input to the χ when 4 output bits are given.

10. The above is true for each lane in row 0. i.e. we can get $4*64$ linear equations on the input to the χ.

11. So in one slice, we need 4 variables to map them to 4 output bits given. (according to χ)

12. So we build initial state such that we have $4*64$ free variables.

13. So take the same structure as for 2R, Keccak-384

14. Take $A[0,2] = A[0,0] \oplus A[0,1] \oplus \alpha_0$
15. and $A[2,2] = A[2,0] \oplus A[2,1] \oplus \alpha_2$
16. All the variable lanes will be linear.
17. By solving the system of linear equations just once we get a solution i.e. Time complexity 1.
18. Time complexity of attack $= 2^{256-256} = 2^0 = 1$

2.8 Preimage attack on 3-round Keccak-512

1. **Note :** This subsection contains my explaination for the attack mentioned in section 6.3 from paper : Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak.
2. We proceed as shown in section 6.1, and complete the 2 rounds.
3. The bits input to step χ of the second round are all linear.
4. Directly inverse 320 bits through χ^{-1} from a given hash value.
5. Of the inverted state, each bit is a sum of 11 bits of the output of the second round.
6. Since $\pi \circ \rho$ just permutate the positions of the bits and ι just add a constant to the first lane, they do not increase the nonlinear terms, and thus we neglect these steps in the last one and a half rounds.
7. As in section 6.3 per equation (14) the equation of $C[x][y][z]$
8. Expanding it :
9. $$C[x][y][z] = B[x][y][z] \oplus \oplus_{y'=0}^{4} B[x-1][y'][z] \oplus \oplus_{y'=0}^{4} B[x+1][y'][z-1]$$
10. Open all the expressions and separate two terms $B[x][y][z]$ and $B[x-1][y][z]$ and rest 9 terms remain as it is.
11. So $$B[x][y][z] \oplus B[x-1][y][z] = (a \oplus c + b) \oplus d$$
12. Where $$a = A[x][y][z], b = A[x+1][y][z], c = A[x+2][y][z], d = A[x-1][y][z]$$
13. So guessing d and other 9 terms would make $C[x][y][z]$ linear.
14. Hence, We linearize $C[x][y][z]$ by guessing 10 bits input to step χ.

15. That is, we obtain $11 = 1 + 10$ linear equations and match 1 bit of the hash value.

16. As such, we can match $128/11 = 11$ bits of the hash value since we have 128 variables.

17. Time complexity of preimage attack $= 2^{512-11} = 2^{501}$.

18. **Note:** There is an improvement for the above attack mentioned in 6.3 by which attack complexity is $= 2^{482}$.

2.9 Preimage attack on 3-round Keccak-256

1. **Note** : The time complexity for attack on 3R, Keccak-256 mentioned in section 6.2 of Guo-Song-Liu paper is $= 2^{192}$.

2. In this method, we try to extend the structure used for preimage attack on 2R, Keccak-384 as shown in Fig. 9 of section 6.1 .

3. For Keccak-256, $d = 256$

4. $c = 512 = 8$ lanes

5. We add one more variable lane as compared to the initial state shown in Fig. 9

6. So in Column 0, we keep $A[0,0], A[0,1], A[0,2], A[0,3]$ as variables.

7. In Column 2, we keep $A[2,0], A[2,1], A[2,2]$ as variables.

8. Keep
$$A[0,3] = A[0,0] \oplus A[0,1] \oplus A[0,2] \oplus \alpha_0$$

9.
$$A[2,2] = A[2,0] \oplus A[2,1] \oplus \alpha_2$$

10. Here, α_0, α_2 are random constants.

11. After applying $\iota \circ \chi \circ \pi \circ \rho \circ \theta$ i.e. One round the state remains linear.

12. Input to the step χ of second round are all linear.

13. Since we have hash of only 4 lanes, assume any random value as value of 5th lane.

14. We can directly inverse these 320 bits through $\chi^{-1} \circ \iota^{-1}$ from given modified hash value.

15. Then we can apply the same technique as mentioned in section 6.3 for this structure also.

16. We linearize the $C[x][y][z]$ term as in (14).

17. As mentioned in **Improved preimage attacks on 3-round Keccak-384 and Keccak-512.**
18. We can match $2\left\lfloor\frac{t-5}{8}\right\rfloor$ bits of a given hash value if we have t variables.
19. For Keccak-256, we have $t = 5 * 64 = 320$ variables.
20. Match bits = 78
21. Attack complexity = $2^{256-78} = 2^{178}$.
22. If correct, there is a small improvement of 2^{14}.

2.10 EuroCrypt'19 Keccak attack :

Preimage Attacks on Round-reduced Keccak -224/256 via an Allocating Approach

1. Key takeaways :
2. Try to use two message blocks instead of one, so as to satisfy some initial conditions by xor-ing the second message block into the output state of the first block.
3. Keep the structure simple

2.11 Try : Preimage Attack on Round-reduced Keccak-384 via an Allocating Approach

Note : Refer mainly, `https://eprint.iacr.org/2019/248.pdf`

1. To meet this condition : (Theorem 1)
2. **Theorem 1 :** Let the messaged state be (a') in figure 5, i.e. bits in Row 0, 2 are unknown and bits in Row 1,3,4 are constants such that
   (a) $a_{x,1,z} = a_{x,3,z} = a_{x,4,z} \oplus 1$ and
   (b) $\bigoplus a_{x,4,z} = 0$
3. For Keccak-384, capacity = $384 * 2 = 768 = 12$ lanes
4. Which means that the last 12 lanes of bits can't be changed after second message block being XOR-ed.
5. So the last 12 lines in State (A) and (B) are identical. (refer Fig. 6)
6. The values of the first 13 lanes in state (B) can be adjusted by second message block, to make bits in state (B) meet condition (i) and (ii) in Theorem 1.
7. Here take state (A) of the form that variables :

(a) in row 4 : $e(0,4), e(1,4), e(2,4), e(3,4), e(4,4)$

(b) in row 3 : $e(0,3), e(1,3), e(2,3), e(3,3), e(4,3)$

8. It suffices to ensure :
9. $\bigoplus e_{x,4,z} = 0$
10. For $a_{x,3,z} \oplus 1 = a_{x,4,z}$. We ensure :
11. $e_{0,3,z} \oplus 1 = e_{0,4,z}$, $e_{1,3,z} \oplus 1 = e_{1,4,z}$, $e_{2,3,z} \oplus 1 = e_{2,4,z}$
12. $e_{3,3,z} \oplus 1 = e_{3,4,z}$, $e_{4,3,z} \oplus 1 = e_{4,4,z}$
13. The above equations are referred as equation (3).
14. To make bits in state (B) meet condition in theorem 1, we need $64*5+1 = 321$ equations to hold.
15. Attack for keccak-384 via allocating approach, consists of 2 stages:
    (a) Precomputation stage : Find a first message block such that equation (3) hold for output bits of 1st block.
    (b) Online stage : Construct an algebraic system using the structure in Theorem 1 for a given hash value, and solve this system for a second message block.
16. Attack has 3 parts :
    (a) Find a 1st message block such that equation (3) holds in precomputation stage.
    (b) Find a 2nd message block such that the state (B) meets condition in theorem 1 and the outputs of the second block equal the given hash value in online stage.
    (c) At last, check how to deal with the paddings.
17. Part 1 : Finding a first message block
18. Fix lanes (0,0), (0,1), (0,2), (2,0), (2,1), (2,2) as variables.
19. Fix lane (1,0), (3, 0) as 1 lane. Rest are 0 lanes
20. The state **doesnt** remain linear after 2 rounds.
21. To find the 1st msg block satisfy equation (3).
22. In the messaged state of 1st round, bits of 6 lanes are set as unknowns.
23. Which means there are $6 * 64 = 384$ unknowns.
24. During this procedure to avoid propagation by θ in first round, $2*64 = 128$ linear constraints are added to the system by assuming sum of linear columns as constants.

25. The state becomes quadratic after 2 rounds.

26. By equation (3) we obtain another 321 quadratic equations.

27. Overall equations : $128 + 321 = 449$ in 384 unknowns.

28. As stated in section 4.1 of this paper by Li-Sun, we use the following methods to create more linear equations : (The details are in reference with Fig 7 of the same paper)

    (a) Let $p_{i,j}$ be the linear representation (polynomial) of bits in state (c) of 2nd round and

    (b) $e_{i,j}$ represents the lane after 2 rounds of 1st msg block, these are quadratic variables, because before the second χ the full state was linear.

    (c) By χ we have :

    (d)
    $$e_{3,4} = p_{3,4} \oplus (p_{4,4} \oplus 1) \cdot p_{0,4}$$

    (e)
    $$e_{4,4} = p_{4,4} \oplus (p_{0,4} \oplus 1) \cdot p_{1,4}$$

    (f) By Equation (3) of same paper we have the following equations :

    (g)
    $$e_{3,4} \oplus e_{4,4} = 1$$

    (h) Hence Equation(5)
    $$p_{3,4} \oplus (p_{4,4} \oplus 1) \cdot p_{0,4} \oplus p_{4,4} \oplus (p_{0,4} \oplus 1) \cdot p_{1,4} = 1$$

    (i) Similarly, Equation (6)
    $$p_{4,4} \oplus (p_{0,4} \oplus 1) \cdot p_{1,4} \oplus p_{4,3} \oplus (p_{0,3} \oplus 1) \cdot p_{1,3} = 1$$

    (j) If the values of pair $(p_{0,3}, p_{0,4})$ are enumerated then both Equation (5) and (6) are linearized in each slice.

    (k) Out of the 449 equations only 128 are linear due to θ

    (l) Get 256 linear equations from equation 5 and 6 by enumeration for 64 slices.

    (m) So no. of linear equations and unknowns match, so we can expect a solution.

    (n) Actuall complexity = $2^{449-384} = 2^{65}$

29. The probability of existence of a solution is 2^{-65}.

30. The whole complexity of finding 1st msg block consists of 2 parts : Complexity 2^{65} of ensuring the system has a solution, and complexity constant of solving the system.

31. Part 2 : Finding a second message block
32. By part 1, we get an initial state of 2nd block satisfying equation (3).
33. For 2nd round, keep structure similar to one proposed by Guo-Song in Linear structure approach.
34. Linear constraints by 1st θ : $2 * 64$ (on all columns). To avoid propagation by θ in 1st round of the second msg.
35. Based on hash value, we can setup 320 linear equations.
36. The system has :
37. No. of linear equations : 2*64 + 320 = 7*64
38. No. of unknowns = 64*6
39. To enusre this system has a solution we enumerate $2^{64*7-64*6} = 2^{64}$
40. So overall it will add up to 2^{66}
41. This is a rough bigger idea, of how to think. We need to ensure that the 2 rounds for second block are done properly.
42. We can also make state (B) similar to the one used in (Fig 8 of paper).

Key Observation :

In Guo-Liu-Song, Fig. 13 the (allowed) linear pattern after 32 rounds is the same as in Li-Sun, Fig. 9 (forbidden, in our view) but complexity seems 2^{192} rather than 2^{161}.

2.12 Preimage Attack on 3R, Keccak-256 using Allocating Approach

Note : I refer to : `https://eprint.iacr.org/2019/248.pdf`

1. Use the (updated image) Fig. 9
2. To find a first message block, we set
3. $2 * 64 + 3 * 4 = 5 * 64 = 320$ linear equations
4. By Assuming sum of linear constants in state (a) and (d)
5. The state after 3rd χ will be quadratic
6. We need $e_{2,3,z} \oplus 1 = e_{2,4,z}$, $e_{3,3,z} \oplus 1 = e_{3,4,z}$, $e_{4,3,z} \oplus 1 = e_{4,4,z}$,
7. and $\bigoplus_{x,z} e_{x,4,z} = 1$
8. Total $3 * 64 + 1 = 193$ quadratic equations to meet conditions in Theorem 1.
9. The no. of unknowns is $6 * 64 = 384$
10. So, this system consists of $320 + 193 = 513$ equations
11. Probability of existence of a solution is $= \frac{1}{2^{512-384}} = 2^{-129}$
12. That is we need to enumerate 2^{129} sum values of linear columns in state (d) to ensure the system has a solution.
13. To solve : We need to enumerate the values of pair $(p_{0,3}, p_{0,4})$ in 16 slices and obtain 64 linear equations.
14. Then, we obtain 64 new linear equations and already has 320 equations. So total = 384 linear equations.
15. and then the system can be solved with a constant time complexity.
16. Thus, the complexity of finding a first msg block $= 2^{129+32} = 2^{161}$.
17. 2^{32} is due to : we need to enumerate 2 variables in 16 slices i.e. 32 variables
18. To solve a second message block, the procedure is same as that of Keccak-224
19. Except that we obtain $4 * 64 = 256$ linear equations from the hash value.

20. The System of this stage consists of : $5 * 64 + 2 * 6402 + 256 = 702$ linear equations
21. and we have 10 variable lanes $= 64 * 10$ unknowns.
22. We need to try $= 2^{702-640} = 2^{62}$ sum values of the linear columns in the state(a) of the 2nd round, to ensure that there is a solution to this system.
23. Complexity of this stage is 2^{62}

2.13 2R, Keccak-384, Internal diff

1. We start with structure of Fig. 9 initial msg in Guo-liu-song paper.
2. To increase no. of freedome degrees we take lanes (3,0), (3,1)
3. To keep θ constant, we are left with 5 (2 + 2 + 1) freedom degrees.
4. Since we are performing experiment based on rotation index = 32, we assume initial msg symmetry.
5. Hence, left with 5*32 bits of freedom degrees.
6. We go forward with $\pi \circ \rho \circ \theta$ and get a linear state, where 2 linear lanes are adjacent. (State is still symmetric).
7. After, χ one quadratic lane is produced. Replace this with a new linear variable of 32-bit symmetry. Since we are following rotation index property. So we artificial degree of freedom = 32
8. So till now we have 5*32=160 actual degrees of freedom and 32 artificial.
9. The hash state has 6 lanes. 5 lanes can be directly inverted by χ and ι. But is this symmetric?
10. From the characteristic found of i=32, 1.5 Rounds K-384: We observe that only 3 lanes are not completely symmetric and have 3 such bits(1 from each).
11. We initially guessed that the msg was symmetric for i=32.
12. There are some symmetric hashes for this should work. (With some probability?)
13. Complexity increase due to probabilistic 6th lane linear conversion will be $= (p)^{32}$, $p = 0.75^{-1}$ due to linear for 6th lane.
14. Complexity $= 2^{6*32-5*32} * (p)^{32}$
15. The above steps is based on what all clarity I have about the method.
16. This should work because for lanes of our interest i.e. first 6 lanes there is hardly any difference.

2.14 Preimage attack on Keecak-224, Keccak-256

1. This is in reference to Guo et al work "Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak"
2. As seen in Fig. 13, 14 of section 6.2, the attacks for Keccak-256 and Keccak-224 resp.
3. Lets see Keccak-224 in more detail, so the attack mentioned in section 6.2 has the following structure as shown in Figure 14.
4. Here the input to the step χ of the third round are all linear.
5. So, if we consider the state input to χ of third round be A, output of χ be B and output of fourth round θ be C.
6. Then we have the following relations :
7. $$B[x][y][z] = A[x][y][z] \oplus (A[x+1][y][z] \oplus 1) \cdot A[x+2][y][z]$$
8. $$C[x][y][z] = B[x][y][z] \oplus \oplus_{y'=0}^{4} B[x-1][y'][z] \oplus \oplus_{y'=0}^{4} B[x+1][y'][z-1]$$
9. $A[x+1][y][z]$ is a common bit in both $B[x][y][z]$ and $B[x-1][y][z]$.
10. We linearize $C[x][y][z]$ by guessing $B[x+1][y'][z-1]$ for all $0 \leq y' \leq 4$ and $B[x-1][y'][z]$ for all y' other than y . These all are 9 guesses.
11. Also, after guessing $A[x+1][y][z]$, hence a total of $9+1 = 10$ bits are guessed.
12. Guessing $A[x+1][y][z]$ makes $B[x][y][z] \oplus B[x-1][y][z]$ term linear.
13. So after these 10 guesses $C[x][y][z]$ is linearized.
14. Hence the method for matching hash bits by linearizing the terms corresponding to these hash bits is applied here.
15. In this we linearize the output of θ of 4th round by guessing bits input to θ. We need 10 guesses to match 1-bit.
16. By this method we obtain $1 + 10$ linear equations by 10 guesses and we are able to match 1 bit of hash value corresponding to $C[x][y][z]$.
17. Hence we can match $\lfloor \frac{t}{11} \rfloor$ bits of the hash value if we have t variables.
18. Why this works ?

19. For Keccak-224 we can't invert the hash value through χ^{-1} as its possible for for Keccak-512, but we can set up the equations such as $a_0 = b_0$ for $b_1 = 1$ according to (6) equation of Guo et. al. paper. (For inversion of hash through χ^{-1} we can think of other ideas too, like randomly setting the remaining hash bits in row0, so as to get full row bits or maybe some other method)

20. The relations obtained after invert operation by χ^{-1}, so the each bit of the state just after final χ is a sum of 11 bits of the output of the third round (due to θ).

21. $\pi \circ \rho$ just permutate the positions of bits and ι only adds a constant, hence they don't increase or introduce any non-linear term. Hence after linearizing $C[x][y][z]$ we can match the hash bits.

22. Time Complexity by above method , Keccak-224 has 127 variables

23. No. of matched bits $= \lfloor \frac{127}{11} \rfloor = 11$

24. Attack complexity $= 2^{224-11} = 2^{213}$.

25. As seen in Improved preimage attacks on 3-round Keccak-384 and Keccak-512 by Guo et. al in section 6.3 . It is possible to cut down the time complexity if we elaborately choose linearly dependent ones, since there will be more degrees of freedom for guessing more linear combinations to match more bits of the hash value.

26. So we start with following two equations, B represents the state after third round χ

27. $$B[x][y][z] = A[x][y][z] \oplus (A[x+1][y][z] \oplus 1) \cdot A[x+2][y][z]$$

    and

    $$B[x-1][y][z] = A[x-1][y][z] \oplus (A[x][y][z] \oplus 1) \cdot A[x+1][y][z]$$

28. By guessing $A[x+1][y][z]$ we make both of the above equations linear.

29. Hence we guess for $0 \leq y \leq 4$ $A[x+1][y][z]$. Similarly for $B[x+1][y][z-1], B[x+2][y][z-1]$ we guess $0 \leq y \leq 4$ $A[x+3][y][z-1]$.

30. $$C[x][y][z] = B[x][y][z] \oplus \oplus_{y'=0}^{4} B[x-1][y'][z] \oplus \oplus_{y'=0}^{4} B[x+1][y'][z-1]$$

    and

    $$C[x+1][y+1][z] = B[x+1][y+1][z] \oplus \oplus_{y'=0}^{4} B[x][y'][z] \oplus \oplus_{y'=0}^{4} B[x+2][y'][z-1]$$

31. These 10 bits are guessed not only make $C[x][y][z]$ linear, but $C[x+1][y+1][z]$ has only one quadratic term $B[x+1][y+1][z]$ left and after guessing that $C[x+1][y+1][z]$ is also linear.

32. We can match 2 bits by setting up 13 (10 + 1 + 2) linear equations.

33. Then we consider another two equations :

34. $$C[x+2][y+2][z-1] = B[x+2][y+2][z-1] \oplus \oplus_{y'=0}^{4} B[x+1][y'][z-1] \oplus \oplus_{y'=0}^{4} B[x+3][y'][z-2]$$

    and

    $$C[x+3][y+3][z-1] = B[x+3][y+3][z-1] \oplus \oplus_{y'=0}^{4} B[x+2][y'][z-1] \oplus \oplus_{y'=0}^{4} B[x+4][y'][z-2]$$

35. Here we can set up another 8 linear equations and match two more bits of the hash value by guessing 6 more bits.

36. So in general, if there are t variables then we can match $2\lfloor \frac{t-5}{8} \rfloor$.

37. So, if we apply the same technique to 4 rounds Keecak-224, Keccak-256

38. Then we have 127 and 64 variables resp, we can match $2\lfloor \frac{t-5}{8} \rfloor$ bits of a hash value if we have t variables.

39. For Keecak-224 : matched bits = 30 for $t = 127$

40. Time Complexity : $2^{224-30} = 2^{194}$

41. And similarly for Keccak-256, Time complexity = $2^{256-14} = 2^{242}$

42. Another Idea : Maybe some better relations can be drawn between variables and hash values to match more bits. Like for symmetric cases, guesses for $A[x][y][z]$ would also work for $A[x][y][z+i]$. So good possibility for more matched bits from lesser variables. Would give good reductions in time complexities for 4 rounds of Keecak-224,256.