Probabilistic Linearization: Internal Differential Collisions in up to 6 Rounds of SHA-3[⋆]

Zhongyi Zhang[1,2][0009−0008−0491−3006], Chengan Hou[1,2][0009−0009−5618−6979], and Meicheng Liu[1,2(✉)][0000−0002−5259−1848]

[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, People's Republic of China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, People's Republic of China
{zhangzhongyi0714,houchengan,liumeicheng}@iie.ac.cn

Abstract. The SHA-3 standard consists of four cryptographic hash functions, called SHA3-224, SHA3-256, SHA3-384 and SHA3-512, and two extendable-output functions (XOFs), called SHAKE128 and SHAKE256. In this paper, we study the collision resistance of the SHA-3 instances. By analyzing the nonlinear layer, we introduce the concept of maximum difference density subspace, and develop a new target internal difference algorithm by probabilistic linearization. We also exploit new strategies for optimizing the internal differential characteristic. Furthermore, we figure out the expected size of collision subsets in internal differentials, by analyzing the collision probability of the digests rather than the intermediate states input to the last nonlinear layer. These techniques enhance the analysis of internal differentials, leading to the best collision attacks on four round-reduced variants of the SHA-3 instances. In particular, the number of attacked rounds is extended to 5 from 4 for SHA3-384, and to 6 from 5 for SHAKE256.

Keywords: Hash Functions · SHA-3 · Collision Attacks · Internal Differentials · Linearization

1 Introduction

The Keccak hash function [2], a creation of Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche [3], emerged victorious in the SHA-3 (Secure Hash Algorithm-3) competition conducted by the National Institute of Standards and Technology (NIST) in the United States. In 2015, NIST released the final version of the SHA-3 standard [8]. The SHA-3 family consists of four cryptographic hash functions, called SHA3-224, SHA3-256, SHA3-384 and

⋆ Supported by the National Natural Science Foundation of China (Grant No. 62122085 and 12231015), the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB0690000 and the Youth Innovation Promotion Association of Chinese Academy of Sciences.

SHA3-512. Additionally, the SHA-3 family includes two Extendable-Output Functions (XOFs), called SHAKE128 and SHAKE256, capable of generating digests of variable lengths. The SHA-3 function, to say, KECCAK, employs a sponge construction that accommodates messages of varying lengths for hash function inputs. The message undergoes padding and is then divided into uniform-sized message blocks. The 1600-bit initial state of KECCAK is XORed with the first message block, followed by 24 rounds of the KECCAK-f permutation applied to update the state and XORing of subsequent message blocks until absorption is complete. Finally, a final 24-round KECCAK-f is applied to the state, and selected state bits are extracted as the resulting digest. Since its introduction in 2008, KECCAK has emerged as a crucial hash function, undergoing extensive security analysis, including evaluations of preimage resistance and collision resistance [1,4,6,7,11,14,15,9,10,17].

A collision attack is to find a pair of distinct messages that produce identical digests. In the literature, differential and internal differential cryptanalysis are the two main cryptanalytic tools for security evaluation of SHA-3 against collision resistance. In 2012, Dinur, Dunkelman and Shamir [5] introduced practical attack strategies targeting 4-round KECCAK-224 and KECCAK-256 using differential cryptanalysis. Their work involved the development of a target difference algorithm (TDA) to establish a link between a 1-round connector and a 3-round high probability differential characteristic. Building on Dinur *et al.*'s framework, Qiao *et al.* [14] expanded the connection of 2-round connectors and 3-round differential characteristic through the application of linearization techniques, and demonstrated actual collisions for 5-round SHAKE128. Further advancements were made in [15,9], where connectors were improved to 3-round connectors using non-full Sbox linearization techniques, resulting in practical collision attacks on 5-round SHA3-224 and SHA3-256. In [12], Huang *et al.* introduced a SAT-based connector to address the issue of insufficient degrees of freedom and proposed a collision attack on SHA3-384 with a complexity of $2^{59.64}$. In [10], Guo *et al.* employed SAT-based automatic search tools and enhanced connector construction algorithms to present the first quantum collision attacks on SHA-3 instances, including 6-round SHA3-224 and SHA3-256, as well as the classical collisions of 6-round SHAKE128.

In 2013, Dinur, Dunkelman and Shamir [6] employed generalized internal differentials in the analysis of SHA-3 for the first time, and conducted practical collision attacks on 3-round KECCAK-384 and KECCAK-512 as well as theoretical attacks on 4-round KECCAK-384 and 5-round KECCAK-256. In the 5-round attack of KECCAK-256, Dinur *et al.* developed an analogous variant of the TDA, called target internal difference algorithm (TIDA). Recently in 2023, Zhang, Hou and Liu [17] improved the target internal difference algorithm and proposed the conditional internal differentials, which directly selects messages that pass through the first two rounds of internal differential characteristic by imposing linear conditions on the initial message space. With the help of these methods, Zhang *et al.* presented collision attacks for all the six SHA-3 functions reduced to up to 5 rounds, including the first collision attack on 4 rounds and 5 rounds of SHAKE256 and the best collision attack on 4-round SHA3-512.

The target difference and internal difference algorithms have demonstrated their core force in collision attacks on reduced SHA-3. Most of these algorithms are designed by setting up and solving linear equations through linearizing the Sbox, of which all the linearization techniques are developed in a deterministic way. Concretely speaking, in the existing target difference algorithms [5,14,15,9] and target internal difference algorithms [6,17], the input difference of each active Sbox is constrained in a two-dimensional affine subspace by introducing 3 linear equations to establish a linear equation system with respect to the input differences, based on which another linear system is built with respect to the values of the states. This results in too many equations to apply TDA or TIDA to SHA-3 instances with large capacity and high security strength (such as SHA3-384, SHA3-512, and even SHAKE-256). Another method is to set up and solve nonlinear equations combining with linear ones, *e.g.*, [12]. This method depends on the ability of the solver. It is challenging to evaluate the complexity in a general case, and it is unclear how to extend to more rounds.

At present, the best known collisions on SHA3-384, SHA3-512 and SHAKE-256 achieve 4 rounds, 4 rounds and 5 rounds respectively, while the best known collisions on SHA3-224, SHA3-256 and SHAKE-128, the three SHA-3 instances that have collision resistance within 128 bits, reach 5 rounds, 5 rounds and 6 rounds respectively, which are exactly one round more. A natural question that arises in this context is whether we could extend the attacks on the SHA-3 instances with stronger collision resistance to more rounds.

Our Contribution. In this paper, we focus on the collision resistance of the SHA-3 instances, especially including those with high security strength, starting from the above question. We first propose a method called probabilistic linearization by introducing the concept of maximum difference density subspace. Then we generalize the target internal difference algorithm by probabilistic linearization, in which we set up much less equations. We also figure out the expected size of collision subsets in internal differentials, which is more accurate than before. In particular, for attacking more rounds of the SHA-3 instances, we construct new internal differential characteristic by exploiting specific properties. These techniques enhance the internal differential cryptanalysis, and lead to the development of theoretical attacks on all the six SHA-3 variants up to 6 rounds. The details of our techniques and results are summarized as follows.

We introduce the *maximum difference density subspace* to constrain the input difference of each active Sbox in an affine subspace of a high dimension (more than two) with maximum probability. The linear system established in this probabilistic way has much less equations, and it requires much less messages of the first block to satisfy the inner part of the system. The cost is that the solution of the system is not necessarily the input difference. We need to create multiple systems and determine whether their solutions sets contain input differences. Overall, the complexity is lower than the previous method of using two-dimensional affine subspace. By probabilistic linearization, we design a new TIDA to link an internal differential characteristic starting from the second round to the initial state of `SHA-3`, also called 1-round internal connector. The application of the new TIDA makes it possible to improve the collision attacks on round-reduced `SHA-3`.

We describe two guidelines for choosing the target internal difference in TIDA, and extend the internal differential characteristic by an extra round, for two `SHA-3` instances, by exploiting their specific properties.

The collision subsets are the output subsets of the last round in internal differentials used to find collisions. In the discussion of a variant of birthday attack [17], the following conclusion was proved. If there are 2^k collision subsets, each of which has a size of 2^m, then during the collision searching stage, we need to search $2^{(m-k)/2}$ states in each subset to find a collision with a probability of 0.4 (the success probability of normal birthday attack), and the time complexity is $2^{(k+m)/2}$. In the previous works [6,17], the size of the collision subset was estimated by the length of the state that affected the output before the last χ mapping. This results in the same time complexity required for collisions of 512 bits and 640 bits, while for the 512-bit case the adversary expects a lower cost than that of 640 bits. In this work, we estimate the size of collision subset by calculating the probability of a collision in the digests rather than the intermediate states input to the last nonlinear layer χ, and thereby obtain a tighter upper bound on the complexity of collision search. The accuracy of our estimation is also confirmed by experiments.

With the help of these techniques, we propose the first collision attack on 5-round `SHA3-384` with a complexity of $2^{170.73}$, as well as the first collision attack on 6-round `SHAKE256` with a complexity of $2^{232.29}$. We also revisit the collision attacks on 4-round `SHA3-512`, 5-round `SHA3-224`/`SHA3-256`/`SHAKE128` and 5-round `SHAKE256` in [17], and obtain a new complexity of $2^{225.29}$, $2^{96.67}$ and $2^{163.28}$ respectively. Our results are listed in Table 1 with a comparison of the related previous work.

Organization. The rest of the paper is organized as follows. In Section 2, we describe the `SHA-3` hash function. In Section 3, we list the notations used in this paper and review the internal differentials. In Section 4, we propose the technique of probabilistic linearization, followed by a description and analysis of the new target internal difference algorithm. Section 5 presents the framework of our attacks, followed by detailed explanations over our techniques. The results of our attacks are given in Section 6. We conduct experiments for verifying our attacks in Section 7, and conclude the paper in Section 8.

2 Description of SHA-3

In this section, we give a brief description of the sponge construction and the `SHA-3` hash function, *i.e.*, the Keccak hash function. Subsequently, the security strengths of `SHA-3` instances are given.

Target	Rounds	Complexity	Attack method	Reference
SHA3-224	5	2^{105}	Internal differential	[17]
	5	$2^{96.67}$	Internal differential	Section 6.3
	5	Practical	Differential	[15]
SHA3-256	5	2^{105}	Internal differential	[17]
	5	$2^{96.67}$	Internal differential	Section 6.3
	5	Practical	Differential	[9]
SHA3-384	4	2^{147}	Internal differential	[6]
	4	2^{76}	Internal differential	[17]
	4	$2^{59.64}$	Differential	[12]
	5	$\mathbf{2^{170.73}}$	Internal differential	Section 6.1
SHA3-512	3	Practical	Internal differential	[6]
	4	2^{237}	Internal differential	[17]
	4	$\mathbf{2^{225.29}}$	Internal differential	Section 6.3
SHAKE128	5	2^{105}	Internal differential	[17]
	5	$2^{96.67}$	Internal differential	Section 6.3
	5	Practical	Differential	[14]
	6	$2^{123.5}$	Differential	[10]
SHAKE256	4	2^{76}	Internal differential	[17]
	5	2^{185}	Internal differential	[17]
	5	$\mathbf{2^{163.28}}$	Internal differential	Section 6.3
	6	$\mathbf{2^{232.29}}$	Internal differential	Section 6.2

Table 1: Comparison of the best collision attacks against the SHA-3 family

2.1 The Sponge Function

The sponge construction is a framework for constructing hash functions based on permutations. The sponge construction proceeds in two phases: absorbing phase and squeezing phase, as shown in Figure 1. The message is firstly padded by appending a bit string of 10*1, where 0* represents a shortest string of 0's so that the length of padded message is multiple of r, and cut into r-bit blocks. The b-bit internal state is initialized to be all zeros. In absorbing phase, each message block is XORed into the first r bits of the current state, and then applying a fixed permutation to the entire b-bit state. The sponge construction switches to the squeezing phase after all message blocks are processed. In this phase, the first r bits of the state are returned as output and the permutation is applied in each iteration. This process is repeated until all d bits digest are produced.

2.2 The Keccak Hash Function

The Keccak permutation has 24 rounds, which operates on the 1600-bit state that can be viewed as a 3-dimensional array of bits. One bit of the state at position (x, y, z) is noted as $A[x][y][z]$, where $0 \leq x, y < 5$ and $0 \leq z < 64$. The designers of Keccak defined the following naming conventions: $A[\cdot][y][z]$ is a

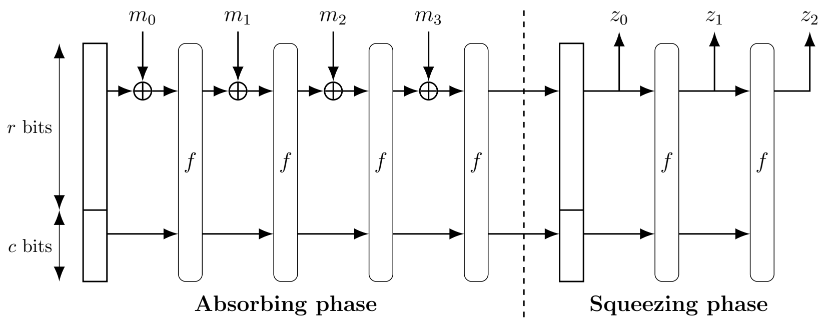

Fig. 1: The sponge construction

row, $A[x][\cdot][z]$ is a column, and $A[x][y][\cdot]$ is a lane; $A[x][\cdot][\cdot]$ is a sheet, $A[\cdot][y][\cdot]$ is a plane, and $A[\cdot][\cdot][z]$ is a slice.

There are five mappings in each round of the permutation:

$$\theta : A[x][y][z] \leftarrow A[x][y][z] + \sum_{y'=0}^{4} A[x-1][y'][z] + \sum_{y'=0}^{4} A[x+1][y'][z-1].$$

$$\rho : A[x][y][z] \leftarrow A[x][y][z+T(x,y)], \text{where } T(x,y) \text{ is a predefined constant.}$$

$$\pi : A[x][y][z] \leftarrow A[x'][y'][z], \text{where } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \end{pmatrix}.$$

$$\chi : A[x][y][z] \leftarrow A[x][y][z] + (\neg(A[x+1][y][z])) \wedge A[x+2][y][z].$$

$$\iota : A \leftarrow A + RC[i_r], \text{where } RC[i_r] \text{ is the round constants.}$$

The addition and multiplication are in $GF(2)$. Since we analyse round-reduced variant with at most 6 rounds, we only give the first five round constants: 0000000000000001, 0000000000008082, 800000000000808a, 8000000080008000, 000000000000808b, 0000000080000001 (given in hexadecimal using the little-endian format).

Function	Rate Size	Capacity Size	Output Size	Security Strengths in Bits		
				Collision	Preimage	2nd Preimage
`SHA3-224`	1152	448	224	112	224	224
`SHA3-256`	1088	512	256	128	256	256
`SHA3-384`	832	768	384	192	384	384
`SHA3-512`	576	1024	512	256	512	512
`SHAKE128`	1344	256	d	$\min(d/2, 128)$	$\geq \min(d, 128)$	$\min(d, 128)$
`SHAKE256`	1088	512	d	$\min(d/2, 256)$	$\geq \min(d, 256)$	$\min(d, 256)$

Table 2: Specifications and security strengths of `SHA-3` functions

2.3 Instances of SHA-3

The four instances of `SHA-3` family named `SHA3-d` are defined from Keccak $[c]$ by appending a two-bit suffix '01' to the message, where $b = 1600$, $c = 2d$ and $d \in \{224, 256, 384, 512\}$. After that, the padding of Keccak is applied. `SHAKE128` and `SHAKE256` are two instances with the capacity $c = 256$ or 512 and any output length d, and the original message M is appended with an additional 4-bit suffix '1111' before applying the padding rule, for any output length. In our attack on `SHAKE256`, the output length is 512 bits. The suffixes "128" and "256" indicate the security strengths that these two functions can generally support. We summarize specifications and security strengths of the `SHA-3` functions in Table 2.

3 Notations and Review of Internal Differentials

In this section, we first enumerate some notations used in this paper. Then we review the concepts of squeeze attack and internal differentials, as well as conditional internal differentials. Finally, the framework of collision attack using internal differentials is summarized, together with the complexity analysis.

3.1 Notations

The addition operation of the state is performed on $GF(2)$ or the linear space over $GF(2)$. We summarize the major notations to be used in this paper here.

n_r	Number of attacked rounds
c	Capacity of a sponge function
r	Rate of a sponge function
b	Width of a Keccak permutation in bits, $b = r + c$
d	Length of the digest in bits
p	Number of minimum fixed bits in the initial state due to padding
i	Period of a symmetric state
$\theta, \rho, \pi, \chi, \iota$	The five mappings that comprise a round.
L	Composition of θ, ρ, π and its inverse denoted by L^{-1}
$\mathtt{R}^j(\cdot)$	Keccak permutation reduced to the first j rounds
$\mathtt{S}(\cdot)$	5-bit Sbox operating on each row of Keccak state
$\delta_{in}, \delta_{out}$	5-bit input and output differences of an Sbox
$\overline{M}$	Padded message of M. Note that $\overline{M}$ is the second block in our attack
$M_0\|\|M_1$	Concatenation of strings M_0 and M_1
α_{j-1}	Input internal difference of the j-th round function with period 32
β_{j-1}	Input internal difference of χ in the j-th round with period 32
$\delta_{in} \to \delta_{out}$	δ_{in} is the internal difference input to χ and δ_{out} the output difference
$\Delta(\cdot)$	Internal difference of one state
$\mathcal{A}_{j-1}$	Bit value vector with period 32 input to the j-th round
$\mathcal{B}_{j-1}$	Bit value vector with period 32 input to χ of the j-th round
$\mathtt{E}[\cdot]$	Expectation of one random variable

3.2 Squeeze Attack

Dinur *et al.* [6] used a specific method in collision attack on Keccak by focusing on a specific subset of outputs to find hash function collisions. It describes how, by limiting outputs to a smaller subset and considering the probabilities of inputs leading to these outputs, one can potentially find collisions more efficiently than traditional methods. This strategy is called a *squeeze attack*.

To motivate this attack, assume that the hash function H maps a set S of possible inputs into a set D of possible outputs. By the birthday paradox, we have to try a subset $S' \subseteq S$ of size $|D|^{1/2}$. We consider a subset D' of D, where $|D'| = q|D|$, and the probability of picking an input in S' whose output is in D' is p. To find a collision in D', the number of inputs in S' we have to try is $(q^{1/2}/p)|D|^{1/2}$. If we can exploit some non-random behavior of the hash function in order to find sets S' and D' for which $p^2 > q$, we can get an improved collision finding algorithm.

Zhang *et al.* [17] proposed a variant of birthday attack, which is essentially a squeeze attack. Assume that the hash function H maps 2^k input subsets $S_1, ..., S_{2^k}$ into output subsets $D_1, ..., D_{2^k}$ (called collision subsets) and is a random function when it is confined to any set S_j, where $S_j (j = 1, ..., 2^k)$ and $D_j (j = 1, ..., 2^k)$ are both pairwise disjoint respectively, $|S_j| = 2^l$, $|D_j| = 2^m (m > 2l)$. For the randomly selected input x in the union of all S_j (denoted as S'), assume that we can determine which output subset $H(x)$ belongs to, but cannot determine the input subset corresponding to x. So the probability of $H(x)$ in subset D_j is 2^{-k} for any $j \in \{1, ..., 2^k\}$. According to the results derived in [17], the total number of inputs required to find a collision is $2^{(k+m)/2}$. Take 4-round `SHA3-512` as an example (Fig. 2). Assume that H maps a set S of possible inputs into a set D of possible outputs and $S' = \bigcup_{j=1}^{2^k} S_j$, $D' = \bigcup_{j=1}^{2^k} D_j$. $|S| = 2^{1600}$, $|D| = 2^{512}$, $|S'| = 2^{252}$, $|D_j| = 2^{296.58}$. There are 2^{156} output subsets D_j, and the size of their union D' is $2^{156+296.58} = 2^{452.58}$. By using conditional internal differentials, the probability of transition from S' to D' is 1. The expected number of inputs to find a collision is $2^{226.29}$.

3.3 Internal Differentials

The internal differential collision attack is essentially a squeeze attack. Internal differential was initially developed by Peyrin [13] in the cryptanalysis of the Grøstl hash function. This method was later generalized by Dinur *et al.* [6] in collision attacks on Keccak. Similar to the case of standard differential analysis, the adversary's goal is to find several internal differential characteristics with high transition probability by tracking the differences between different parts of the internal state through the cryptographic function. The difference between standard differential and internal differential is that the input of the former is multiple message pairs, while the input of latter is multiple messages. Another difference is that in standard differential analysis, the adversary can check whether a collision occurred after each input of a message pair. In internal differential analysis, the adversary has to input enough messages to enter the

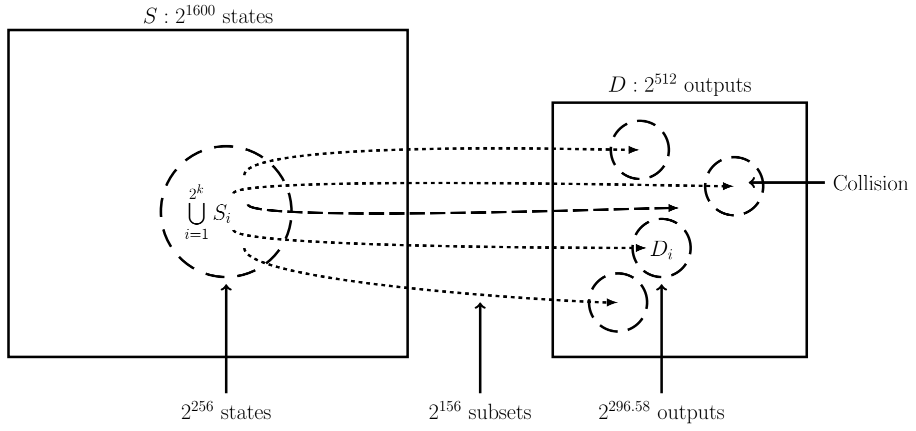

Fig. 2: A squeeze attack

characteristic, and search for each collision subset at the end of the characteristic until a collision is found. It can be seen that internal differential analysis is a squeeze attack, the collision subset is a subset of the output set D, and the transition probability of the internal differential characteristic is the probability p that the input enter the output subset.

In KECCAK, a state is called a *symmetric state* if it has period i in the z-axis. This means that for all (x, y, z), there is some positive integer i less than 64 such that state A satisfies $A[x][y][z] = A[x][y][(z+i) \bmod 64]$. An interesting property of KECCAK is that after applying any of the θ, ρ, π, χ operations to a symmetric state, the new state is still a symmetric state and maintains its period.

In this paper, we set $i = 32$, and a symmetric state consists of two repetitions of slices 0-31. Each sequence of slices (0-31, 32-63) is called a *consecutive slice set* or *CSS* in short. For a state A, we can express it as $A = (\hat{A}, \hat{\hat{A}})$, where $\hat{A}$ and $\hat{\hat{A}}$ are the first and second CSS's respectively, called *CSS form* in this paper.

Note that all round constants are not periodic, so the ι operation will introduce a difference between the two CSS's, and this difference will propagate through other operations. To characterize the difference, the internal difference is defined as follows. The set $\{v+u|u \text{ is symmetric}\}$ obtained by adding all symmetric states to a single state v is called *internal difference*, recorded as $[i, v]$. If $v = \mathbf{0}$, the subset $[i, \mathbf{0}]$ is called *zero internal difference*, and other internal differences $[i, v]$ are cosets of $[i, \mathbf{0}]$. The state v is called the *representative state*. We choose v satisfying $v[x][y][z] = 0$ $(z \in 32, 33, ..., 63)$ as the *canonical representative state*. For a state v, we refer to its corresponding canonical representative state of the internal difference as its internal difference, denoted by $\Delta(v)$. Then, an internal differential for round function $\mathtt{R}$ is a pair of internal differences (α_1, α_2), and its probability is defined as $\Pr(\Delta(\mathtt{R}(v)) = \alpha_2|\Delta(v) = \alpha_1)$.

Similarly to the standard differential characteristics, an internal differential characteristic is defined as the propagation of internal differences through round function. The internal differential transition of the j-th round in the characteristic is denoted by $\alpha_{j-1} \xrightarrow{L} \beta_{j-1} \xrightarrow{\chi} \alpha_j^{\star} \xrightarrow{\iota} \alpha_j$, where $j \geq 1$. An internal differential characteristic for n_r-round collision attack extends before the χ operation of the $(n_r - 1)$-th round. Non-zero internal differences will generate multiple internal differences after the χ operation. The output subset of each internal difference after the last round function is called *collision subset*. In the framework of our collision attack, collision search is carried out in each collision subset.

3.4 Conditional Internal Differentials

The conditional internal differential was developed by Zhang, Hou and Liu in [17] as a technique to find messages conforming the first two rounds internal differential characteristic. For a known 1.5-round internal differential characteristic,

$$\alpha_0 \xrightarrow{L} \beta_0 \xrightarrow{\iota\circ\chi} \alpha_1 \xrightarrow{L} \beta_1,$$

let the state with period 32 in the internal difference be expressed as follows:

$$v^{(0)} = \alpha_0 + \mathcal{A}_0, v^{(0.5)} = \beta_0 + \mathcal{B}_0, v^{(1)} = \alpha_1 + \mathcal{A}_1, v^{(1.5)} = \beta_1 + \mathcal{B}_1,$$

where $v^{(j)}, \alpha_j$ and $\mathcal{A}_j$ are the state, internal difference and symmetric state before the $(j+1)$-th θ operation respectively, $v^{(j+0.5)}, \beta_j$ and $\mathcal{B}_j$ are the state, internal difference and symmetric state before the $(j+1)$-th χ operation respectively. Since $\mathcal{A}_j$ and $\mathcal{B}_j$ each have two identical CSS's, they have the following vector form (determined by their first CSS):

$$\hat{\mathcal{A}}_j = (a_0^{(j)}, \ldots, a_{799}^{(j)}), \hat{\mathcal{B}}_j = (b_0^{(j)}, \ldots, b_{799}^{(j)}).$$

So the CSS forms of symmetric states $\mathcal{A}_j$ and $\mathcal{B}_j$ are $(\hat{\mathcal{A}}_j, \hat{\mathcal{A}}_j)$ and $(\hat{\mathcal{B}}_j, \hat{\mathcal{B}}_j)$ respectively. Since the second CSS of the internal difference is a zero vector, taking α_0 as an example, its separable form is symbolized as $(0, \alpha_0)$. To ensure a deterministic passage through the χ operations of the first two rounds, we should find $\hat{\mathcal{A}}_0$ such that $\hat{\mathcal{B}}_0$ and $\hat{\mathcal{B}}_1$ satisfy the respective *differential transition conditions* (denoted as $E_0(\hat{\mathcal{B}}_0)$ and $E_1(\hat{\mathcal{B}}_1)$). $E_0(\hat{\mathcal{B}}_0)$ and $E_1(\hat{\mathcal{B}}_1)$ can be transformed into $E_0'(\hat{\mathcal{A}}_0)$ and $E_1'(\hat{\mathcal{A}}_1)$ by linear operation ($L(\mathcal{A}_0) = \mathcal{B}_0, L(\mathcal{A}_1) = \mathcal{B}_1$). Since $\iota \circ \chi((\hat{\mathcal{B}}_0, \hat{\mathcal{B}}_0 \oplus \beta_0)) = \iota \circ \chi(u^{(0.5)}) = u^{(1)} = (\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_1 \oplus \alpha_1)$, we know $\hat{\mathcal{A}}_1 = \iota \circ \chi(\hat{\mathcal{B}}_0)$, and $a_j^{(1)}$ is equal to $b_j^{(0)} \oplus (b_{j+32}^{(0)} \oplus 1) \cdot b_{j+64}^{(0)}$ up to a constant. We regard $b_{j+32}^{(0)}$ or $b_{j+64}^{(0)}$ as a variable x, and $a_j^{(1)}$ can be transformed into a linear function about $\hat{\mathcal{B}}_0$ by assigning a value to x. Generally, we set all bits $b_{j+32}^{(0)}$ (or $b_{j+64}^{(0)}$) corresponding to the bits $a_j^{(1)}$ appearing in $E_1'(\hat{\mathcal{A}}_1)$ as intermediate variables $\{x_t\}_{t \in I}$ (I is an index set). System $E_1'(\hat{\mathcal{A}}_1)$ can be transformed into a system $E_1''(\hat{\mathcal{A}}_0)$ on $\hat{\mathcal{A}}_0$ by assigning values to all intermediate variables. Noting that each x_t is actually a linear function about $\hat{\mathcal{A}}_0$, so assigning values to all x_t is equivalent to adding linear equations to the initial space of $\hat{\mathcal{A}}_0$. The set of these equations is recorded as system E_2. Then, by solving the linear system $E_0' \bigcup E_1'' \bigcup E_2$ and

XORing each solution with α_0, we can get the messages conforming 2-round internal differential characteristic.

The intermediate variables $\{x_t\}_{t\in I}$ may be linearly dependent. We call the variables in the maximal linearly independent system of $\{x_t\}_{t\in I}$ are the *free intermediate variables.* During a collision attack, it is often necessary to iterate through all possible values of free variables in order to gather sufficient initial messages. Consequently, we must judiciously choose intermediate variables to minimize the number of free variables, thus reducing the overall complexity when solving linear systems at this stage.

Definition 1. [17] *Given the non-zero input difference $\delta_{in} = (\delta_0, \dots, \delta_4)^T$ of the 5-bit* Keccak *Sbox, the output difference δ_{out} is determined by q $(2 \le q \le 4)$ linear conditions with respect to the actual input $x = (x_0, \dots, x_4)^T$. The q linear conditions $\{L_t(x)\}_{t=0}^{q-1}$ (without constant terms) are called differential transition conditions. Equivalently, $\delta_{out} = \mathtt{S}(x) \oplus \mathtt{S}(x \oplus \delta_{in}) = C \cdot x \oplus \eta$, where $C \in \mathbb{F}_2^{5\times 5}$ is a matrix $(rank(C) \in \{2,3,4\})$ and $\eta \in \mathbb{F}_2^5$ is a constant vector. It can be easily verified that C and η can be represented by δ_{in} as*

$$C = \begin{pmatrix} & \delta_2 & \delta_1 & & \\ & & \delta_3 & \delta_2 & \\ & & & \delta_4 & \delta_3 \\ \delta_4 & & & & \delta_0 \\ \delta_1 & \delta_0 & & & \end{pmatrix}, \eta = \mathtt{S}(\delta_{in}) = \begin{pmatrix} \delta_0 \oplus (\delta_1 \oplus 1)\delta_2 \\ \delta_1 \oplus (\delta_2 \oplus 1)\delta_3 \\ \delta_2 \oplus (\delta_3 \oplus 1)\delta_4 \\ \delta_3 \oplus (\delta_4 \oplus 1)\delta_0 \\ \delta_4 \oplus (\delta_0 \oplus 1)\delta_1 \end{pmatrix}.$$

Take $\delta_{in} = 0x03$ as an example, the output difference is

$$\delta_{out} = \begin{pmatrix} 0&0&1&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&1 \\ 1&1&0&0&0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

The differential transition conditions are $\{l_0 = x_4, l_1 = x_2, l_2 = x_0 + x_1\}$.

Definition 2 (Transition Condition Number [17]). *Given an internal differential characteristic, for the input internal difference β_{j-1} of the j-th χ operation, the rank of the set of all differential transition conditions obtained from β_{j-1} is called the transition condition number (denoted as k_j).*

Property 1. [17] If the transition condition number of β_{j-1} is k, there are at most 2^k possible output internal differences and the lower bound of transition probability is 2^{-k}.

3.5 Collision Attacks Using Conditional Internal Differentials

Utilizing internal differentials and conditional internal differentials, Zhang *et al.* [17] select 2-block messages as inputs and propose collision attacks on 5-round SHA-3. As shown in Fig. 3, given a 2.5 round internal differential characteristic starting from the second round, their attack framework can be summarized into three parts:

1. TIDA Stage: Calculate in advance the number of initial messages needed to achieve a collision, aiming for a probability that matches that of the birthday attack, which is assumed to be 2^n. Continue to run the targeted internal differential algorithm until accumulating a sufficient quantity of messages. The output of the target internal difference algorithm includes the first block M_0, the initial input difference v_0, and an affine subspace W_1 of the second block M_1. Within this subspace W_1, the values of M_1 can ensure that v_0 legally propagate to the target internal difference v_1.
2. Selecting Messages Stage and Collecting Messages Stage: Utilize conditional internal differential techniques to select initial messages from subspace W_1 that satisfy all or part of the second round differential transition conditions. Subsequently, calculate the states and internal differences of these messages after four round functions. Following the fifth round function, organize the outputs into respective collision subsets, which are distinguished by the values of their internal differences observed after the fourth round.
3. Brute-force Searching Stage: Using hash table techniques, perform an exhaustive search in each collision subset sequentially until two states with the same digest value are found and their respective preimages are recovered.

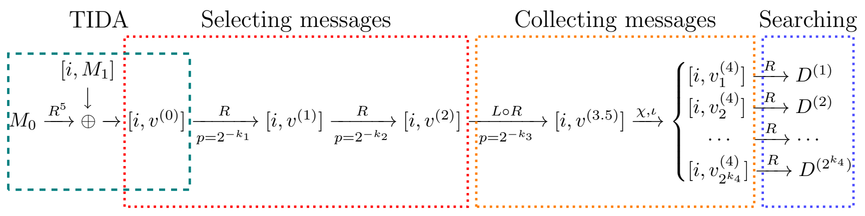
TIDA
Selecting messages
Collecting messages
Searching
$[i, M_1]$
$M_0 \xrightarrow{R^5} \oplus \rightarrow [i, v^{(0)}]$
$\xrightarrow[p=2^{-k_1}]{R} [i, v^{(1)}] \xrightarrow[p=2^{-k_2}]{R} [i, v^{(2)}]$
$\xrightarrow[p=2^{-k_3}]{L\circ R} [i, v^{(3.5)}] \xrightarrow{\chi,\iota}$
$[i, v_1^{(4)}] \xrightarrow{R} D^{(1)}$
$[i, v_2^{(4)}] \xrightarrow{R} D^{(2)}$
$\cdots \xrightarrow{R} \cdots$
$[i, v_{2^{k_4}}^{(4)}] \xrightarrow{R} D^{(2^{k_4})}$

Fig. 3: The framework of 5-round collision attack

The Complexity. The time complexity is determined by the complexity of the following parts.

1. The total complexity of TIDA stage, denoted by 2^{d_1}. It is calculated by multiplying the average time of TIDA and the number of times TIDA is run. Assume that the time to run TIDA once is 2^t, and the average size of the space W_1 output by TIDA is 2^d. Then the complexity of this part is $2^t \cdot \max\{2^{n-d}, 1\}$.
2. The complexity of Selecting Messages Stage and Collecting Messages Stage, denoted by 2^{d_2}. Assuming that q second round differential transition conditions are satisfied, then $2^{d_2} = 2^{n-q}$.
3. The complexity of Brute-force Searching Stage, denoted by 2^s. Assuming that the probability of the given differential characteristic is 2^{-k}, then $2^n = 2^{s+k}$.

Thus, the final complexity of collision attack is summarized in Equation (1),

$$2^{d_1} + 2^{d_2} + 2^s = 2^t \cdot \max\{2^{k+s-d}, 1\} + 2^{k+s-q} + 2^s. \tag{1}$$

In our attack strategy, we employ probabilistic linearization to accelerate TIDA, and thereby reduce the time complexity 2^t of the TIDA stage (see also Section 4). Furthermore, we use a probabilistic approach to estimate an upper bound for the size of the collision subset, effectively reducing the complexity 2^s of Brute-force searching stage (see also Section 5.3). We also provide new constructions of internal differential characteristics for some SHA-3 variants, including 5-round SHA3-384 and 6-round SHAKE256, and thus reduce the complexity of selecting messages stage and collecting messages stage (see also Section 5.2).

4 Probabilistic Linearization and Target Internal Difference Algorithm

In this section, we first review the target difference and internal difference algorithms which are the core algorithms for finding collisions of reduced `SHA-3` in the literature, and then describe a generalized target internal difference algorithm by the method of probabilistic linearization.

4.1 Connector and Connectivity Problem

In the differential collision attacks on reduced `SHA-3` [5,14,15,9], an n_1-round connector and an n_2-round differential characteristic are combined to find collisions for (n_1+n_2) rounds. Here an *n_1-round connector* is a procedure which produces a large number of one-block message pairs satisfying three requirements:

- The last $(c+p)$-bit difference input to the first round is zero;
- The last $(c+p)$-bit value of the initial state is fixed;
- The output difference after n_1 rounds should be equal to the target difference, *i.e.*, the input difference of the n_2-round differential characteristic.

The construction of n_1-round connector with the target difference α_{n_1} is essentially to find many solutions of the nonlinear system on M_1 and M_1',

$$\mathtt{R}^{n_1}(\overline{M_1}||0^c) \oplus \mathtt{R}^{n_1}(\overline{M_1'}||0^c) = \alpha_{n_1}. \tag{2}$$

Dinur *et al.* [5] initially introduced the *target difference algorithm* (TDA) to link a differential characteristic to the initial message space, which is a 1-round connector based on the linearization of the Keccak Sbox. Qiao *et al.* [14], Song *et al.* [15], and Guo *et al.* [9] further constructed the 2-round connector and 3-round connector respectively by the full and non-full Sbox linearization. In [6], Dinur *et al.* extended the technique of TDA to *target internal difference algorithm* (TIDA) for constructing 1-round connector of internal differentials. In [17], Zhang *et al.* redesigned TIDA for conditional internal differentials. All the connectors in these works were constructed by adding enough linear equations to the nonlinear system (2) or its internal-differential version such that the system is simplified into a linear one. A side effect of these methods is a rapid increase in the size of the linear system. For the `SHA-3` variants with a large capacity, *e.g.*, `SHA3-384` and `SHA3-512`, the initial message space output by TIDA is too small each time, and TIDA runs slowly, making it impossible to launch a collision attack.

In the differential collision attack on 4-round `SHA3-384`, Huang *et al.* [12] constructed 1-round connector by solving the nonlinear system on M_1 and M_1',

$$\mathtt{R}(\mathtt{R}^4(M_0||0^c) \oplus (\overline{M_1}||0^c)) \oplus \mathtt{R}(\mathtt{R}^4(M_0'||0^c) \oplus (\overline{M_1'}||0^c)) = \alpha_1. \tag{3}$$

In this connector, Huang *et al.* used two-block messages and observed that a linear system is implied by (3) on the difference of the inner part of middle states $\mathtt{R}^4(M_0||0^c)$ and $\mathtt{R}^4(M_0'||0^c)$ generated by (M_0, M_0'). Specifically, they grouped 10 differences out of all 5-bit output differences into one category and showed that their input differences always satisfy a linear relationship. Then the authors used a SAT solver to solve the 1-round connectivity problems when the middle states generated by the first block pairs (M_0, M_0') satisfy these linear conditions. In this attack, the solutions of connectivity problem are not lost by adding those linear conditions. The SAT-based connector makes the attack very efficient, and it brings the fastest collision attack on 4-round `SHA3-384`. On the other hand, this method highly depends on the ability of the solver. It is difficult to evaluate the complexity in a general case, and it is unclear how to extend to more rounds.

Inspired by the aforementioned works, we propose a *probabilistic linearization* method for the connector construction, specially demonstrated in internal differential cryptanalysis: to find 2-block message $M_0||M_1$ satisfying the internal connectivity problem.

Definition 3. *In a collision attack on n_r-round* `SHA-3`*, given the first block M_0 and a 1-round input difference α_1, the internal connectivity problem is to find a second block M_1, if exists, such that*

$$\Delta(\mathtt{R}(\mathtt{R}^{n_r}(M_0||0^c) \oplus (\overline{M_1}||0^c))) = \alpha_1. \tag{4}$$

We generalize the target internal difference algorithm by probabilistic linearization to solve the internal connectivity problem. Its essence is to transform the system (4) into two linear systems E_Δ and $E_{\Delta(L(\alpha_0))\to\alpha_1^\star}$, in a probabilistic way rather than in a deterministic way. The system E_Δ is called the *input difference system* (Def. 4), and its solution space contains correct input internal differences with a probability of $p_1^\star$. That is, solving $1/p_1^\star$ systems of E_Δ's gives an input internal difference α_0 propagated to α_1 on average. The system $E_{\Delta(L(\alpha_0))\to\alpha_1^\star}$ is *differential transition system* (Def. 5), used to determine whether the input internal difference α_0 can be propagated to α_1 for legal messages. This method neither radically reduces the solution set of (4), nor depends on the solver of nonlinear system.

The definitions of input difference system and differential transition system are described as follows.

Definition 4 (Input Difference System). *In internal differentials of* `SHA-3`*, given the characteristic starting from the second round, the linear system with respect to the input difference α_0 of the first round is the input difference system, regarded as E_Δ. The last $(c/2+p)$ bits of α_0 and the last $(c+p)$ bits of the input state of the second block are defined as padding and inner bits (or inner bits for short), which are known but can not be controlled. After applying Gaussian elimination to E_Δ, the equations related only to the inner bits are called the inner part (or inner system) of E_Δ, denoted as $E_\mathcal{C}$.*

Example 1. Let's consider the state of two bits in each lane. We set the rate part to the first 15 lanes, totaling 30 bits, and the remaining 20 bits are the capacity part. The internal difference α_0 and β_0 can be represented by 25-bit states $(x_0,...,x_{24})$ and $(y_0,...,y_{24})$ respectively, where $(x_{15},...,x_{24})$ are inner bits. When the output internal difference $\alpha_1^\star = (0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,1,0,0,0,1)$, there are 4 active Sboxes and 1 non-active Sbox, and the output differences of the 5 Sboxes are in sequence: 0x04, 0x00, 0x03, 0x08, 0x11. According to the method of [17], each active Sbox adds 3 equations to the system, and each non-active Sbox adds 5 equations to the system, resulting in a total of 17 equations related to β_0, denoted as E'_Δ. We substitute $\beta_0 = \Delta(L(\alpha_0))$ into system E'_Δ, transforming it into a system relating to α_0, and obtain system E_Δ after performing Gaussian elimination. As shown in system (5), the internal system $E_\mathcal{C}$ of E_Δ is $\{x_{17}+x_{23}+x_{24}=1, x_{18}+x_{23}=0, x_{19}+x_{21}+x_{24}=0, x_{22}+x_{23}+x_{24}=0\}$, which only depends on inner bits.

$$E'_\Delta : \begin{cases} y_2 = 1, \\ y_3 = 0, \\ y_4 = 0, \\ y_5 = 0, \\ y_6 = 0, \\ y_7 = 0, \\ y_8 = 0, \\ y_9 = 0, \\ y_{12} = 0, \\ y_{10}+y_{13} = 0, \\ y_{11}+y_{13} = 1, \\ y_{15} = 0, \\ y_{18} = 1, \\ y_{19} = 0, \\ y_{21} = 0, \\ y_{20}+y_{22} = 1, \\ y_{20}+y_{24} = 1. \end{cases} \Rightarrow E_\Delta : \begin{cases} x_0+x_5+x_{15}+x_{16}+x_{20}+x_{23}+x_{24} = 0, \\ x_1+x_{16}+x_{21} = 0, \\ x_2+x_{16}+x_{21}+x_{23}+x_{24} = 1, \\ x_3+x_{23} = 0, \\ x_4+x_{24} = 0, \\ x_6+x_{11}+x_{24} = 0, \\ x_7+x_{21}+x_{23}+x_{24} = 1, \\ x_8+x_{23} = 0, \\ x_9+x_{24} = 0, \\ x_{10}+x_{16} = 0, \\ x_{12}+x_{23}+x_{24} = 1, \\ x_{13}+x_{23} = 0, \\ x_{14}+x_{16}+x_{21}+x_{24} = 0, \\ x_{17}+x_{23}+x_{24} = 1, \\ x_{18}+x_{23} = 0, \\ x_{19}+x_{21}+x_{24} = 0, \\ x_{22}+x_{23}+x_{24} = 0. \end{cases} \quad (5)$$

Definition 5 (Differential Transition System). *Given an input difference β and its output difference α after χ, the linear system composed of all differential transition conditions and their values of the constant terms is called the differential transition system from β to α, regarded as $E_{\beta\to\alpha}$.*

4.2 Probabilistic Linearization of the Sbox for Input Difference

In this section, we analyze the 5-bit Keccak Sbox and apply probabilistic linearization on the input difference set of each active Sbox. Specifically, when building the input difference system, we select the t-dimensional affine subspace that contains the most input differences (denoted as the maximum difference density subspace), which is equivalent to adding $(5-t)$ linear equations to the system for each active Sbox. Since not all elements in the maximum difference density subspace are input differences of the corresponding Sbox, the solutions of input difference system may not propagate to the target difference $\alpha_1^\star$. Therefore we need to solve multiple input difference systems and search the solution set of each system to obtain an input difference. Although the probabilistic linearization cannot obtain the input difference deterministically, from the overall perspective of TIDA, this will result in lower complexity. Before delving into the details of Keccak Sbox, we first introduce a property of 5-dimensional linear space.

Property 2. Any t-dimensional affine subspace U of the 5-dimensional $\mathbb{F}_2^5$ can be regarded as the kernel of a particular equation within $\mathbb{F}_2^5$. Namely, for the t-dimensional affine subspace $U \subset \mathbb{F}_2^5$, there exists $(l_0^{(j)}, l_1^{(j)}, l_2^{(j)}, l_3^{(j)}, l_4^{(j)}, q^{(j)}) \in \mathbb{F}_2^6$, such that

$$U = \left\{ (x_0, ..., x_4) \;\middle|\; \begin{array}{c} \sum_{j=0}^{4} l_j^{(1)} \cdot x_j = q^{(1)}, \\ \vdots \qquad , \\ \sum_{j=0}^{4} l_j^{(5-t)} \cdot x_j = q^{(5-t)} \end{array} \right\} \triangleq \mathrm{Ker}\left(\textstyle\sum_{j=0}^{4} l_j^{(1)} \cdot 2^j, q^{(1)} | \dots | \sum_{j=0}^{4} l_j^{(5-t)} \cdot 2^j, q^{(5-t)}\right)$$

where $(l_0^{(j)}, ..., l_4^{(j)}) \neq 0$, for $1 \leq j \leq 5-t$.

From the above property, we infer that the number of 4-dimensional affine subspaces of the 5-dimensional space is 62. In the following, we sometimes use $\mathrm{Ker}(\sum_{j=0}^{4} l_j \cdot 2^j, q)$ or $(\sum_{j=0}^{4} l_j \cdot 2^j, q)$ to refer to a certain 4-dimensional subspace.

Given a non-zero output difference δ_{out} of the Keccak Sbox, for the t-dimensional affine space U, the proportion of the input differences of δ_{out} in U is called the *difference density* of U with respect to δ_{out}, recorded as $\mathrm{P}(U, \delta_{out})$. That is to say, $\mathrm{P}(U, \delta_{out}) = \#\{\delta \in U | \delta \to \delta_{out}\}/|U|$.

For example, all input differences δ_{in}'s of $\delta_{out} = $ 0x05 and the input differences in space Ker(0x01, 1) are listed as follows, and the difference density is 7/16.

δ_{in}	0x04	0x06	0x07	0x0f	0x11	0x16	0x17	0x19	0x1b	0x1d
Ker(0x01,1)			✓	✓	✓		✓	✓	✓	✓

For δ_{out}, the t-dimensional affine subspace with maximum difference density is called the *maximum difference density subspace* (to say max density subspace) of δ_{out}. Note that there may be more than one max density subspace. Table 3 records all max density subspaces with $\delta_{out} = 0x05$ and their average transition condition number $\mathtt{E}[k_U]$. The average transition condition number of an affine subspace U is the average of transition condition numbers of all input differences of δ_{out} in U. We call the table containing all max density subspaces and corresponding average transition condition numbers the *max density subspace table* (MDST). For $\delta_{out} \in \{0x01, 0x02, 0x04, 0x08, 0x10, 0x03, 0x06, 0x0c, 0x18, 0x11\}$, we find that it has only one 4-dimensional max density subspace, and the subspace contains all input differences. In order to use the greedy algorithm to build a probabilistic input difference system with smaller inner part for 5-round `SHA3-384` collision attack, we also record the 4-dimensional affine subspace whose difference density is second only to the max density subspace in MDST of this type of δ_{out}.

U	(0x01,1)	(0x04,1)	(0x09,0)	(0x11,0)	(0x06,0)	(0x0c,1)	(0x14,1)	(0x13,0)	(0x0e,0)
$\mathtt{E}[k_U]$	27/7	24/7	24/7	24/7	25/7	24/7	24/7	24/7	25/7

Table 3: The max density subspace of $\delta_{out} = 0x05$

We now show the establishment process of linear system E_Δ. The first step is to select an 800-dimension subspace (named W) to which the initial internal difference $\alpha_0 = (\Delta_\mathcal{R}, \Delta_\mathcal{C})$ propagates through the linear layer L, that is, $W = L(\Delta_\mathcal{R}, \Delta_\mathcal{C})$. For each non-active Sbox involved in $\alpha_1^\star$, it is still constrained by five linear equations. For an active Sbox, given the output difference δ_{out}, we select a max density subspace $\mathrm{Ker}(\sum_{j=0}^{4} l_j \cdot 2^j, q)$ of δ_{out} and add the equation $l_0 \cdot x_0 + \cdots + l_4 \cdot x_4 = q$ into E_Δ. The other details of establishment are shown in Procedure PIDS. Assuming that E_Δ is consistent and $\Delta(L^{-1}(\beta_0))$ is a solution to E_Δ, the probability $p_1^\star$ of β_0 being the input difference of $\alpha_1^\star$ is determined by the product of the corresponding difference density of each selected affine subspace.

We can also try to construct the system E_Δ using 3-dimensional affine subspaces. In this scenario, the difference density of max density subspace will increase, but it will not exceed twice the difference density in 4-dimension subspace. Furthermore, when constructing system E_Δ, each active Sbox introduces two equations, leading to an expansion of the inner part within E_Δ. Considering both factors, we prioritize the utilization of 4-dimensional affine subspace in our attack. In the collision attack on 5-round `SHA3-384`, we design the Procedure GreedyPIDS according to the greedy algorithm to construct the system E_Δ with smaller inner part, and the algorithm is shown in Supplementary Material A. We have discussed in detail the advantages of adding more linear conditions to the active Sbox in Supplementary Material F.

Procedure PIDS($\alpha_1^\star$, MDST)

```
Input: Internal difference $\alpha_1^\star$ and MDST.
Output: Probabilistic input difference system $E_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C})$, probability $p_1^\star$.
1  Set $E_\Delta = \emptyset$, $p_1^\star = 1$ and $W = L(\Delta_\mathcal{R}, \Delta_\mathcal{C})$.
   /* $W$ is a variable vector $(w_0, ..., w_{799})$, $w_i = w_i(\Delta_\mathcal{R}, \Delta_\mathcal{C})$ is a linear
      function about $\Delta_\mathcal{R} = (\delta_0, ..., \delta_{799-p-c/2}), \Delta_\mathcal{C} = (\delta_{800-p-c/2}, ..., \delta_{799})$   */
2  for $j = 0 \to 160$ do
3      Get the output difference $\delta_{out}$ of the $j$-th Sbox from $\alpha_1^\star$.
4      if $\delta_{out} = 0$ then
5          $E_\Delta = E_\Delta \cup \{w_{5j}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = 0, \ldots, w_{5j+4}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = 0\}$.
6      else
7          Select one subspace $\mathrm{Ker}(\sum_{j=0}^{4} l_j \cdot 2^j, q)$ of $\delta_{out}$ from MDST[$\delta_{out}$].
8          $E_\Delta = E_\Delta \cup \{l_0 \cdot w_{5j}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) + \cdots + l_4 \cdot w_{5j+4}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = q\}$.
9          $p_1^\star = p_1^\star \cdot \mathrm{P}(\mathrm{Ker}(\sum_{j=0}^{4} l_j \cdot 2^j, q), \delta_{out})$.
10     end
11 end
12 return $(E_\Delta, p_1^\star)$
```

4.3 An Improved Target Internal Difference Algorithm

After obtaining the input difference β_0 of $\alpha_1^\star$ by solving $1/p_1^\star$ systems E_Δ, another question is that β_0 may not be propagated to $\alpha_1^\star$ for any legal message. This question is equivalent to determining whether the differential transition system $E_{\beta_0 \to \alpha_1^\star}$ is consistent with respect to M_1. In our work, we modify the TIDA in conditional internal differential cryptanalysis [17] to obtain two-block message $M_0||M_1$ that satisfies Eq.(4) by solving the input difference systems E_Δ and the differential transition systems $E_{\beta_0 \to \alpha_1^\star}$ sequentially. The procedure of TIDA is shown in Algorithm 1.

For the target internal difference $\Delta_T = \alpha_1$, we begin by running the Procedure PIDS to set the system $E_\Delta = E_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C})$, where $\Delta_\mathcal{R}$ represents the first $(r/2 - p)$ bits of α_0 and $\Delta_\mathcal{C}$ represents the inner bits of $\Delta(\mathtt{R}^{n_r}(M_0))$. Next, we apply Gaussian elimination to the system E_Δ and then extract the inner system $E_\mathcal{C}$. The first block M_0 is randomly selected until linear system E_Δ is consistent. Randomly select a solution α_0 of $E_\mathcal{R}$ and calculate its corresponding internal difference β_0 after the linear layer. If β_0 is an input difference of $\alpha_1^\star$ and the differential transition system $E_{\beta_0 \to \alpha_1^\star}$ is consistent, then get a solution M_1 of $E_{\beta_0 \to \alpha_1^\star}$. The 2-block message $M_0||M_1$ satisfies the internal connectivity problem Eq.(4). If none of the solutions for $E_\mathcal{C}$ satisfy the conditions mentioned above, we will proceed by randomly selecting the first message block M_0 until we find a solution for M_1. Most of M_0 are filtered out before solving the system $E_\mathcal{R}$ because they do not satisfy system $E_\mathcal{C}$. Consequently, only a small subset of M_0 results in the input difference system and the differential transition system being consistent.

Remark 1. For collision attack on 6-round `SHAKE256`, we only randomly select one solution instead of searching for all solutions for each system $E_\mathcal{R}$.

Algorithm 1: TIDA

```
Input: Target internal difference $\alpha_1$, target number of rounds $n_r$, and MDST
Output: the first block $M_0$, the value subspace $W_1$ of the second $M_1$, initial
        internal difference $\alpha_0$
1  Set $E_\Delta = \emptyset, \alpha_1^\star = \alpha_1 \oplus RC[1]$ and $p_1^\star = 1$.
2  $(E_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C}), p_1^\star) = \mathrm{PIDS}(\Delta_T, p_1^\star)$.
3  Reduce $E_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = E_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) \cup E_\mathcal{C}(\Delta_\mathcal{C})$.
4  Set $W_1$ to NULL.
5  do
6      Set $\Delta_c^\star = (\delta'_{800-p-c/2}, ..., \delta'_{799})$.
7      do
8          Randomly select $M_0$ and compute $\Delta(\mathtt{R}^{n_r}(M_0))$.
9          for each integer $j \in [800 - p - c/2, 800)$ do
10             $\delta'_j = \Delta(\mathtt{R}^{n_r}(M_0))[j]$. // the $j$-th bit of $\Delta(\mathtt{R}^{n_r}(M_0))$
11         end
12     while $\Delta_c^\star$ is not a solution of $E_\mathcal{C}$;
13     Solve $E_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C}^\star)$ and obtain its solution space $U_\mathcal{C}$.
14     do
15         Randomly choose and delete a solution $\alpha_0$ of $U_\mathcal{C}$.
16         $\beta_0 = \Delta(L(\alpha_0))$.
17         if $\beta_0$ is the input difference of $\alpha_1^\star$ then
18             Obtain the differential transition system $E_{\beta_0 \to \alpha_1^\star}$. // as defined
                  in Def. 5
19             Solve the linear system $E_{\beta_0 \to \alpha_1^\star}(\mathtt{R}^{n_r}(M_0) \oplus (\overline{X}||0^c))$ on $X$, and get
                its solution space $W_1$ if it has solutions.
20         end
21     while $W_1$ is NULL and $U_\mathcal{C} \neq \emptyset$;
22 while $W_1$ is NULL;
23 return $(M_0, W_1, \alpha_0)$
```

Analysis of TIDA. Next we explain why the output of TIDA satisfies the internal connectivity problem and analyze the complexity of the algorithm. Assume that for a given first message block M_0, $\Delta(\mathtt{R}^{n_r}(M_0))[800 - p - c/2, ..., 799]$ is a solution of system $E_\mathcal{C}(\Delta_\mathcal{C})$.

We rewrite Eq.(4) as $\Delta(\mathtt{R}(\Delta_H \oplus \mathcal{S})) = \alpha_1$, where Δ_H is the internal difference of $\mathtt{R}^{n_r}(M_0) \oplus M_1$, $\mathcal{S}$ is the symmetric state $\mathtt{R}^{n_r}(M_0) \oplus M_1 \oplus \Delta_H$. Namely, $\mathtt{R}^{n_r}(M_0) \oplus M_1 = (\hat{\mathcal{S}}, \hat{\mathcal{S}} \oplus \Delta_H)$. Then $\Delta_H[800 - p - c/2, ..., 799]$ is in the solution set of $E_\mathcal{C}$, and the freedom degrees of $\Delta_H[0, ..., 799 - p - c/2]$ and $\mathcal{S}$ are both $(800-p-c/2)$. Eq.(4) having solutions for M_1 is equivalent to $\Delta(\mathtt{R}(\Delta_H \oplus \mathcal{S})) = \alpha_1$ having solutions for Δ_H and $\mathcal{S}$.

For the system E_Δ we selected, there are s independent linear equations, including s_1 equations in $E_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C})$ and s_2 equations in $E_\mathcal{C}(\Delta_\mathcal{C})$. Given $\Delta_\mathcal{C}^\star = \Delta_H[800 - p - c/2, ..., 799]$, let $\Delta_H^\star$ be the Δ_H that satisfies all equations in $E_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C}^\star)$, the degrees of freedom of $\Delta_H^\star$ are $d = (800 - p - c/2 - s_1)$. Since the value space of $\Delta_H^\star$ is a subset of the value space of Δ_H, the solutions of

$\Delta(\mathtt{R}(\Delta_H^\star \oplus \mathcal{S})) = \alpha_1$ with respect to $(\Delta_H^\star, \mathcal{S})$ must be the solutions of $\Delta(\mathtt{R}(\Delta_H \oplus \mathcal{S})) = \alpha_1$.

Set $\tilde{\mathcal{S}}$ is a symmetric state with 800 degrees of freedom, then $\Delta(\mathtt{R}(\alpha_0 \oplus \tilde{\mathcal{S}})) = \alpha_1$ having solutions for $\tilde{\mathcal{S}}$ is equivalent to $\Delta(L(\alpha_0))$ being the input difference of α_1. Therefore, $\Delta(\mathtt{R}(\Delta_H^\star \oplus \tilde{\mathcal{S}})) = \alpha_1$ having solutions for $(\Delta_H^\star, \tilde{\mathcal{S}})$ means that there is at least one input difference of α_1 in the value space of $\Delta(L(\Delta_H^\star))$. The value space of $\Delta_H^\star$ is actually the solution spaces of $E_{\mathcal{R}}(X, \Delta_{\mathcal{C}}^\star)$ on X, denoted by $U_{H,\mathcal{C}}$, so the value space of $\Delta(L(\Delta_H^\star))$ is $\Delta(L(U_{H,\mathcal{C}}))$. If there is an input difference β_0 of α_1, which satisfies the differential transition system $E_{\beta_0 \to \alpha_1^\star}$ is consistent, then $\Delta(\mathtt{R}(\Delta_H^\star \oplus \mathcal{S})) = \alpha_1$ has solutions with respect to $(\Delta_H^\star, \mathcal{S})$. Let $N_{H,\mathcal{C}}$ be the set of containing all input differences of α_1 in space $\Delta(L(U_{H,\mathcal{C}}))$, the above derivation can be summarized as the following conclusions:

$$\begin{aligned}
&\text{For given } U_{H,\mathcal{C}}, \text{ there exits } \beta_0 \in N_{H,\mathcal{C}} \text{ such that } E_{\beta_0 \to \alpha_1^\star} \text{ is consistent.}\\
\Longrightarrow &\Delta(\mathtt{R}(\Delta_H^\star \oplus \mathcal{S})) = \alpha_1 \text{ having solutions for } (\Delta_H^\star, \mathcal{S}).\\
\Longrightarrow &\Delta(\mathtt{R}(\Delta_H \oplus \mathcal{S})) = \alpha_1 \text{ having solutions for } (\Delta_H, \mathcal{S}).\\
\Longleftrightarrow &\Delta(\mathtt{R}(\mathtt{R}^t(M_0||0^c) \oplus (\overline{M_1}||0^c))) = \alpha_1 \text{ having solutions for } M_1.
\end{aligned}$$

Therefore the output of Algorithm 1 satisfies the internal connectivity problem.

Assume that, on average, the probability of $N_{H,\mathcal{C}} \neq \emptyset$ is denoted as p_1. Under the condition that $N_{H,\mathcal{C}} \neq \emptyset$, the average probability of there being an input difference $\beta_0 \in N_{H,\mathcal{C}}$, which makes $E_{\beta_0 \to \alpha_1^\star}$ consistent, is represented as p_2. The complexity of TIDA can be upper-bounded by $2^{s_2 + log_2(p_1^{-1}) + log_2(p_2^{-1})}$. Below we will make a preliminary estimate of the value of p_1, and the value of p_2 will be calculated in Section 6.1 and Section 6.2.

Let U_0 be the set of all input differences of α_1 before linear layer, and U_1 be the solution space of $E_\Delta(\tilde{\Delta})$, where $\tilde{\Delta}$ is an internal difference with 800 degrees of freedom, $\#U_1 = 2^{800-s}$. Then $U_2 = U_0 \bigcap U_1$ consists of the input differences of α_1 in U_1. From the above definition, $\Delta_H^\star \in U_1$, the freedom degrees of $\Delta_H^\star$ are d and $U_{H,\mathcal{C}}$ is the value space of $\Delta_H^\star$. In other words, U_1 is divided into $2^{p+c/2-s_2}$ subspaces of size 2^d according to the value of the inner bits $\Delta_{\mathcal{C}}^\star$, and $U_{H,\mathcal{C}}$ is one of the subspaces. Set $\Delta_H^\star(j)$ is an internal difference in $U_{H,\mathcal{C}}$. Noting that there is a unique solution of E_Δ corresponding to each element of U_2, if $\Delta_H^\star(j)$ is randomly chosen from U_1, then the probability that it falls into U_2 is $p_1^\star = \#U_2/\#U_1$.

Record u_j as

$$u_j = \begin{cases} \{\Delta(L(\Delta_H^\star(j)))\} & , \Delta_H^\star(j) \in U_2 \\ \emptyset & , \Delta_H^\star(j) \notin U_2. \end{cases}$$

Then $N_{H,\mathcal{C}} = \bigcup_{\Delta_H^\star(j) \in U_{H,\mathcal{C}}} u_j$, the probability $p_1 > p_1^\star$. In the analysis of 6-round `SHAKE256`, we use $p_1^\star$ instead of p_1 to evaluate the complexity of TIDA. For 5-round `SHA3-384`, we determine the value of p_1 through experiments.

5 Applications to Collision Attacks on Reduced SHA-3

In this section, we first give an overview of our collision attacks, followed by two directions for improvements: the selection of α_1 in the internal connectivity problem and the estimation of the complexity of searching stage. Finally, we introduce two guidelines for constructing characteristics and a new method for bounding the size of collision subset in response to the two directions.

5.1 The Framework of the Attack

Following the framework of Dinur *et al.* [6] and Zhang *et al.*'s work [17], our collision attack consists of three parts, *i.e.*, an internal connector linking the internal differential characteristic with the initial value, a high probability internal differential characteristic and several collision subsets generated by the characteristic for searching collisions. Given an internal differential characteristic covering $(n_r - 2.5)$ rounds starting from the second round, there are three stages in our n_r-round collision attacks, as depicted in Figure 4. Next, we overview the three stages.

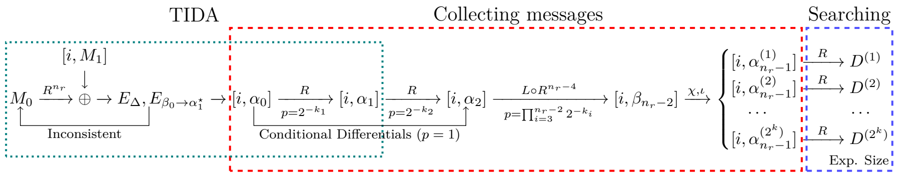
TIDA
Collecting messages
Searching
$[i, M_1]$
$M_0 \xrightarrow{R^{n_r}} \oplus \rightarrow E_\Delta, E_{\beta_0\rightarrow\alpha_1^\star} \rightarrow [i,\alpha_0] \xrightarrow[p=2^{-k_1}]{R} [i,\alpha_1] \xrightarrow[p=2^{-k_2}]{R} [i,\alpha_2] \xrightarrow[p=\prod_{i=3}^{n_r-2} 2^{-k_i}]{L\circ R^{n_r-4}} [i,\beta_{n_r-2}] \xrightarrow{\chi,\iota}$
Inconsistent
Conditional Differentials $(p = 1)$
$[i,\alpha^{(1)}_{n_r-1}] \xrightarrow{R} D^{(1)}$
$[i,\alpha^{(2)}_{n_r-1}] \xrightarrow{R} D^{(2)}$
$[i,\alpha^{(2^k)}_{n_r-1}] \xrightarrow{R} D^{(2^k)}$
Exp. Size

Fig. 4: The framework of n_r-round collision attack

TIDA stage. In this stage, for the target internal difference α_1, we establish the input difference system E_Δ by probabilistic linearization and filter out the first message blocks M_0 that make E_Δ consistent. After that, select α_0 from the solution space of each E_Δ that can be legally propagated to α_1.
Collecting messages stage. For each $\beta_0 = \Delta(L(\alpha_0))$ obtained in the previous stage, we solve the differential transition systems $E_{\beta_0\rightarrow\alpha_1^\star}$ and $E_{\beta_1\rightarrow\alpha_2^\star}$ and get several subspaces of the second block M_1 passing the first 2 rounds by conditional internal differentials. After that, compute after $(n_r - 1)$ rounds functions from the subspaces and store these outputs into different sets.
Searching stage. In this stage, perform brute-force search on the outputs of each set after one round function, continuing until a collision is found. In order for collisions to be found, we need to estimate the size of each collision subset in advance to determine the number of states that need to be searched.

Note that the time complexity of the TIDA stage is the number of runs of Algorithm 1 multiplied by the time of a single run. We find that the choice of α_1 directly affects these two aspects. This leads to the first question of this section, how to choose α_1 to reduce the complexity of the TIDA stage. It will be answered in Section 5.2.

Another question is how to more accurately estimate the number of states required in searching stage. This is because the initial messages generated in the first two stage are stored in the collision subsets after going through internal differential characteristic. The number of states required during searching stage directly determines the complexity of collecting message stages. We will address this question in Section 5.3.

5.2 Constructing Internal Differential Characteristics

For 5-round `SHA3-384` and 6-round `SHAKE256`, we extend the characteristic starting from the second round forward by one extra round. When α_1 is fixed, the internal differential characteristic is determined. We choose α_1 according to two guidelines as follows:

- **Guideline 1**: The probability of first round differential transition should not be too small.
- **Guideline 2**: The inner part of system E_Δ should not have too many equations.

If we violate Guideline 1, there will not be enough messages to enter the collision subset during the collecting messages stage. Unless we run TIDA multiple times, by identifying multiple initial input differences α_0 to generate a sufficient number of initial messages. However, this will result in increased complexity of the TIDA stage.

If Guideline 2 is not satisfied, the process of generating the first message block M_0 will consume more time complexity, potentially causing the overall complexity of TIDA to surpass the boundary of birthday attack and hinder the effectiveness of our collision attack.

In order to follow Guideline 1, we try to find an input internal difference β_0 of $\alpha_1^\star$ with high transition probability. For 5-round `SHA3-384`, the number of initial messages we need to complete the collision attack is $2^{k_1+151.38}$, where $2^{151.38}$ is the time complexity of searching stage (see Section 6.1). When β_0 is determined, the size of initial message space is $2^{800-384-4} = 2^{412}$, so k_1 have to satisfy $k_1 \leq 412 - 151.38 = 260.62$. The contribution of each active Sbox to k_1 is between 2 and 4. Assuming an average of 3.5, there should be approximately $\lceil 261/3.5 \rceil = 75$ active Sboxes in α_1. For 6-round `SHAKE256`, the size of initial message space is $2^{800-256-6} = 2^{538}$, k_1 is upper-bounded by $538 - k_2 - 233.29 = 273.71$, where $2^{233.29}$ is the time complexity of searching stage (see Section 6.2). Therefore the number of active Sboxes in α_1 is approximately $\lceil 273/3.5 \rceil = 78$.

As for Guideline 2, since there are not many internal differences α_1 that meet the above conditions, we can start from α_1 with the least active Sboxes to check whether this guideline is fulfilled.

Among the characteristics we searched by MILP, there are 77 active Sboxes in Characteristic 1 and 65 active Sboxes in Characteristic 2. The average transition condition number of β_0 can be upper-bounded by accumulating $\mathtt{E}[k_U]$ in MDST. In Characteristic 2, $\mathtt{E}[k_1] \leq \sum \mathtt{E}[k_U] = 224.71 \leq 225$. In Characteristic 1, the rank of $E_{\beta_0 \to \alpha_1}$ is equal to the total number of equations minus the number of equations related only to the inner bits. The average number of equations of inner part is 12.65, $\mathtt{E}[k_1] = \sum \mathtt{E}[k_U] - 12.65 = 267.88 - 12.65 \leq 256$.

5.3 The Expected Size of Collision Subset

In the searching stage using birthday attack [17], the size of collision subset determines its complexity. If there are 2^k collision subsets, each of which is 2^m in size. We need to search $2^{(m-k)/2}$ states in each subset to find a collision with a probability of 0.4, time complexity is $2^{(k+m)/2}$. In [17], taking the digest lengths of 512 bits and 640 bits as an example, the sizes of corresponding collision subset are both estimated to be 2^{320}, resulting in the same search complexity. But when $d = 512$, the complexity of collision searching stage is lower. In this section, we use a probabilistic method to bound the size of the collision subset.

The collision subset is the output of the last round $\iota \circ \chi$, and since ι does not affect the size of the collision subset, it can be disregarded in this section. A property of χ is its independent application to each plane of the state, specifically, mapping each plane to itself. Let X_δ and X'_δ be the two 10-bit vectors with the same internal difference $\delta = (\delta_0, \cdots, \delta_4)$, each of which contains only two rows, *i.e.*, five lanes with length 2. With these two vectors as the input of χ, below we show an observation on Keccak Sbox which describes the probability of collision of the output vectors in the first j lanes on average for $1 \leq j \leq 5$.

Observation 1 *Let χ_j be the first j components of the 5-bit* Keccak *Sbox, $1 \leq j \leq 5$. Given $\delta \in \mathbb{F}_2^5$, let $X_\delta = (\hat{X}, \hat{X} + \delta)$ and $X'_\delta = (\hat{X}', \hat{X}' + \delta)$ with $\hat{X}, \hat{X}' \in \mathbb{F}_2^5$, and $Y_j = (\chi_j(\hat{X}), \chi_j(\hat{X} + \delta))$ and $Y'_j = (\chi_j(\hat{X}'), \chi_j(\hat{X}' + \delta))$. Let $P_\delta^{(j)}$ be the probability of collision between Y_j and Y'_j, that is,*

$$P_\delta^{(j)} = \Pr_{\hat{X}, \hat{X}'}\{Y_j = Y'_j | Y_j = (\chi_j(\hat{X}), \chi_j(\hat{X} + \delta)), Y'_j = (\chi_j(\hat{X}'), \chi_j(\hat{X}' + \delta))\}.$$

Then the geometric mean of the probability that Y_j and Y'_j are equal is $P_j = (\prod_\delta P_\delta^{(j)})^{1/32}$, the arithmetic mean is $P'_j = (\sum_\delta P_\delta^{(j)})/32$, and

$$P_1 = 0.297, P_2 = 0.118, P_3 = 0.052, P_4 = 0.035, P_5 = 0.031;$$

$$P'_1 = 0.313, P'_2 = 0.125, P'_3 = 0.055, P'_4 = 0.036, P'_5 = 0.031.$$

The above observation is obtained through calculations over all possible cases. Building upon this, we now consider the lanes with length 64. Given the internal difference $\Delta = (\Delta_0, \cdots, \Delta_4)$, where each Δ_j has 32 bits. Randomly select two distinct states from the internal difference $[32, \Delta]$, and let P_Δ be the probability of collision in the first three lanes after applying the operation χ. Then $P_\Delta = \prod_{j=0}^{31} P_{\delta^{(j)}}^{(3)}$, where $\delta^{(j)} = (\Delta_0[j], \cdots, \Delta_4[j])$. When Δ is randomly selected, P_Δ

can be estimated as $\overline{P_\Delta} = P_3^{32}$. The probability that no collision is found after performing t birthday attacks in $[32, \Delta]$ is

$$P = (1 - P_\Delta)(1 - P_\Delta)^2 \cdots (1 - P_\Delta)^{t-1} \approx e^{-t^2 P_\Delta/2}. \tag{6}$$

$N_\Delta = P_\Delta^{-1}$ is called the *expected size* of the collision subset corresponding to $[32, \Delta]$. When $t = N_\Delta^{1/2}$, the probability of finding a collision is consistent with the success probability of the birthday attack. If there are w internal differences $\{[32, \Delta^{(j)}]\}_{j=0}^{w-1}$, the probability of performing t birthday attacks in each corresponding collision subset without finding a collision is

$$P = \prod_{j=0}^{w-1} e^{-t^2 P_{\Delta^{(j)}}/2} = e^{-t^2 \sum_{j=0}^{w-1} P_{\Delta^{(j)}}/2}. \tag{7}$$

When w is too large to calculate P, we use $\overline{P_\Delta}^{-1}$ to estimate the expected size of the collision subset corresponding to each $[32, \Delta^{(j)}]$, which is called the *geometric expected size*. The complexity estimated by the geometric expected size is $(w\overline{P_\Delta}^{-1})^{1/2}$. Correspondingly, the *arithmetic expected size* is defined as $\overline{P'_\Delta}^{-1}$, where $P'_\Delta = P_j'^{32}$. Using arithmetic expected size to estimate complexity will result in lower attack boundaries. It can be seen that the geometric (arithmetic) expected size of the collision subset is related to the length of the digest. Table 4 lists the size of the collision subset with an output length within 640 bits.

collision length	G.E. size[†]	A.E. size[‡]	collision length	G.E. size	A.E. size
1 lane	2^{56}	$2^{53.70}$	6 lanes	2^{216}	$2^{213.70}$
2 lanes	$2^{98.64}$	2^{96}	7 lanes	$2^{258.64}$	2^{256}
3 lanes	$2^{136.58}$	$2^{134.16}$	8 lanes	$2^{296.58}$	$2^{294.16}$
4 lanes	$2^{154.64}$	$2^{153.93}$	9 lanes	$2^{314.64}$	$2^{313.93}$
5 lanes	2^{160}	2^{160}	10 lanes	2^{320}	2^{320}

[†] Geometric expected size
[‡] Arithmetic expected size

Table 4: The expected size for collision length no more than 640 bits

If $t = (\sum_{j=0}^{w-1} P_{\Delta^{(j)}})^{-1/2}$, the above attack is equivalent to the birthday attack, with a time complexity of wt. Since $P_{\Delta^{(j)}} = \prod_{i=0}^{31} P^{(3)}_{\delta^{(j,i)}}$, where $\Delta^{(j)} = (\Delta_0^{(j)}, \cdots, \Delta_4^{(j)}), \delta^{(j,i)} = (\Delta_0^{(j)}[i], \cdots, \Delta_4^{(j)}[i])$, there is the following inequality:

$$w \cdot t \leq w(w(\prod_{j=0}^{w-1}\prod_{i=0}^{31} P^{(3)}_{\delta^{(j,i)}})^{1/w})^{-1/2} \approx w(w(\overline{P_\Delta}^{w})^{1/w})^{-1/2} = (w\overline{P_\Delta}^{-1})^{1/2}. \tag{8}$$

When $w \cdot 32$ internal differences $\delta^{(j,i)}$ traverse each of the 5-bit differences exactly w times, the approximate equality sign in the previous inequality becomes an equality sign. It can be seen from (8) that the complexity estimated by geometric expected size is generally larger than the real complexity.

We performed a simulation on the probability of a collision during the searching stage for the case of the lane length 32, using the geometric expected size as the size of each collision subset. In each experiment, we randomly select 2^{12} internal differences with period of 16, and the goal is to find a collision in the first two lanes by searching the 2^{12} corresponding collision subsets. The geometric expected size is $2^{49.32}$, and the number of states required to generate a collision is $2^{(49.32+12)/2} = 2^{30.66}$. We randomly select $2^{18.66}$ states in each internal difference and calculate their outputs after χ mapping to store them in the corresponding collision subset. Finally, the collision subsets are searched sequentially until a collision is found. Out of 128 experiments, collisions were found 86 times. The probability of finding a collision is 0.67, which exceeds the theoretical probability of 0.4 shown in [17]. This shows that the complexity of searching stage calculated by the geometric expected size is larger than but close to the real complexity.

6 Details of the Attacks

In this section, we present the details and theoretical results of our collision attack on 4-round `SHA3-512`, 5-round `SHAKE128/SHA3-224/SHA3-256`, 5-round `SHAKE256`, 5-round `SHA3-384` and 6-round `SHAKE256`. For 4-round `SHA3-512`, 5-round `SHAKE128/SHA3-224/SHA3-256` and 5-round `SHAKE256`, we follow the attack framework in [17] and use the expected size of collision subset in the searching stage to more accurately bound the complexity. For 5-round `SHA3-384` and 6-round `SHAKE256`, the collision attacks are performed in the framework in Section 5.1. Given an internal differential characteristic covering $(n_r - 2.5)$ rounds starting from the second round, the steps of a collision attack on n_r-round `SHA-3` are as follows:

1. Calculate the number N of required initial messages based on the characteristic. Run the Procedure PIDS to construct the input difference system E_Δ for α_0.
2. According to the system E_Δ, run Algorithm 1 to obtain the first block M_0 and the initial internal difference α_0.
3. Construct differential transition systems of the first two rounds and solve them to get initial messages by conditional internal differentials. If the number of initial messages is less than N, go back to Step 2.
4. Pick an arbitrary message and calculate its internal difference of after (n_r-1) rounds. If the internal differential characteristic satisfied, store the message into the corresponding subset. Otherwise, discard the message and go back to Step 4 until collect enough states.
5. Choose an unselected subset.
   (a) Pick a state and store its output after the n_r-th round in a hash table (along with its initial message).
   (b) If a collision is found, stop and output it. Else if all state are chosen and there is no collision, go back to Step 5. Otherwise, go back to (a).

6.1 A Collision Attack on 5-round SHA3-384

For 5-round `SHA3-384`, we use the internal differential characteristic given in Characteristic 1 in Supplementary Material B. The transition condition numbers are $(k_2, k_3, k_4) = (25, 18, 16)$. For $\alpha_1^\star$, there are 77 active Sboxes and 83 non-active Sboxes.

During the TIDA stage, we combine the Procedure PIDS and the greedy algorithm to design the Procedure GreedyPIDS. The input difference system E_Δ contains 492 equations, of which the inner part contains 97 equations. The output probability $p_1^\star$ of Procedure GreedyPIDS is $2^{-76.50}$. Continuing the discussion in Section 4.3, we find that there are 45 active Sboxes involving only variables in the inner bits of $\tilde{\Delta}^\star$, where $\tilde{\Delta}^\star$ is $\tilde{\Delta}$ that satisfies all equations in $E_\Delta(\tilde{\Delta})$. Noting that for the randomly selected $\Delta_H^\star$, the value of $\Delta(L(\Delta_H^\star))$ on these 45 Sboxes are all random values with 4 degrees of freedom. We can calculate that the probability that the projection of $\Delta(L(\Delta_H^\star))$ on these 45 Sboxes is the input difference is $2^{-45.70}$. The degrees of freedom of $\Delta_H^\star$ are $800 - 388 - 395 = 17$, which means that there are 17 free variables in $\Delta_H^\star$. The remaining 32 active Sboxes will be affected by these 17 free variables and the inner bits of $\Delta_H^\star$ at the same time. Since each active Sbox is independent of each other, we can randomly select several $\Delta(L(\Delta_H^\star))$ to estimate the probability that the other 32 Sboxes will obtain the input difference at the same time. We search the value spaces of 2^{24} randomly selected $\Delta_H^\star$ and find that there are 34 $N_{H,\mathcal{C}}$ contained input differences which simultaneously satisfied 32 Sboxes. Therefore the value of p_1 can be estimated as $2^{-45.70} \cdot 34/2^{-24} = 2^{-64.61}$. For p_2, we randomly choose 2^{28} input differences β_0 and find that there are $2^{17.42}$ consistent systems $E_{\beta_0 \to \alpha_1^\star}$. So $p_2 = 2^{17.42-28} = 2^{-10.58}$. The number of the first blocks M_0 required at this stage is $2^{97+64.61+10.58} = 2^{172.19}$, of which $2^{75.19}$ M_0 enter the step of solving system $E_{\mathcal{R}}(\Delta_{\mathcal{R}}, \Delta_{\mathcal{C}}^\star)$.

The dimension of the solution space $U_{H,\mathcal{C}}$ of each $E_{\mathcal{R}}$ is $(800 - 492 + 97 - 388) = 17$, so there are $2^{92.19}$ internal differences β_0 that need to be judged as the input difference of $\alpha_1^\star$. We can record the input differences of 77 active Sboxes in 77 tables in advance. For each β_0, determine whether its projection on each active Sboxes is in the corresponding table. The complexity of this step is $2^{92.19} \cdot 77 \cdot 2^{q_1} = 2^{98.46+q_1}$. Where 2^{q_1} is the complexity of calculating the projection of β_0 on an Sbox and looking up the table, which is significantly lower than the complexity of a 5-round `SHA3-384` operation. The complexity of solving the differential transition systems can be ignored, because only about $2^{10.58}$ input differences β_0 enter this step.

In collecting messages stage and searching stage, we use conditional internal differential to select the initial messages that can pass the first round function, which only required solving the differential transition system $E_{\beta_0 \to \alpha_1^\star}$. The complexity of this step can be ignored. The average of all collision subsets' expected size is $2^{200.75}$ for $d = 384$. We need to choose $2^{25+18+(200.75+16)/2} = 2^{151.375}$ initial messages from the solution space of $E_{\beta_0 \to \alpha_1^\star}$, which confirm the internal differential characteristic of the first round. From the discussion in Section 5.2, we know that the size of the solution space of $E_{\beta_0 \to \alpha_1^\star}$ is 2^{412-k_1}, where $412 - \mathtt{E}[k_1] \geq$

$412 - 256 = 156$. Therefore, running the TIDA once on average can obtain enough initial messages. The total complexity is $2^{172.19} + 2^{98.46+q_1} + 2^{151.375} = 2^{172.19}$.

Furthermore, we removed 7 redundant conditions and added a linear condition to each of the other 7 active Sboxes to ensure that no new redundant conditions were generated, resulting in a new input difference system $\hat{E}_{\Delta}$. The maximum density subspace selected for each active Sbox can be found in Supplementary Material C. In this new attack, the size of inner part is reduced from 97 to 90, and the value of p_1 is $2^{-70.15}$. The total complexity of the attack is $2^{170.73}$.

6.2 A Collision Attack on 6-round SHAKE256

In this section, we present a collision attack on 6-round `SHAKE256`. Our attack uses internal differential characteristic given in Characteristic 2 in Supplementary Material B, which covers 3.5 rounds starting from the second round. The transition condition numbers are $(k_2, k_3, k_4, k_5) = (31, 25, 20, 83)$. For $\alpha_1^{\star}$, there are 65 active Sboxes and 95 non-active Sboxes.

In TIDA stage, each active Sbox provides 1 linear equation, each non-active Sbox provides 5 linear equations. There are 540 equations in system E_{Δ} and 59 equations in the inner part. The output probability $p_1^{\star}$ of Procedure PIDS is $2^{-57.22}$, and the average number of equations related only to the inner bits in $E_{\beta_0 \to \alpha_1}$ is 2. We take $2^{-57.22}$ and 2^{-2} as the values of p_1 and p_2 respectively in Section 4.3. Therefore, the number of the first blocks M_0 required at this stage is $2^{59+57.22+2} = 2^{118.22}$, of which $2^{59.22}$ M_0 enter the step of solving system $E_{\mathcal{R}}(\Delta_{\mathcal{R}}, \Delta_{\mathcal{C}}^{\star})$. Since we only randomly select one solution of each $E_{\mathcal{R}}$, the complexity of determining the input difference of $\alpha_1^{\star}$ and solving the differential transition system can be ignored.

In collecting messages stage, we need to collect $2^{k_3+k_4} \cdot 2^{q_2}$ initial messages, which conforming the first two rounds internal differential characteristic, where 2^{q_2} is the complexity of the searching stage. Since period $i = 32$ is half the lane size, we can put the first CSS of two completely symmetric states in each state for computation. In addition, there are $800 - 262 - k_1 = 538 - k_1$ variables, denoted by $\{x_j\}_{j \in J}$, after solving the linear system $E_{\beta_0 \to \alpha_1^{\star}}$. Fixing the values of free intermediate variables $\{x_t\}_{t \in I}$ gives at most 231 equations on $\{x_j\}_{j \in J}$, and therefore there are at most 231+31 equations in total together with the 31 differential transition conditions from β_1 to $\alpha_2^{\star}$. On average, by solving such linear system of up to 262 equations we obtain 2^{276-k_1} solutions. In order to collect 2^{45+q_2} states in searching stage, we need to randomly assign all free variables $2^{45+q_2-276+k_1} = 2^{k_1+q_2-231}$ times, and solve the linear system $E_{\beta_1 \to \alpha_2^{\star}}$ each time to get the initial message. The complexity of this step is $2^{k_1+q_2-231} \cdot 2^{q_3}$, where $2^{q_3} < 262^2 \times 317 = 2^{24.38}$ is the time complexity of solving a linear system containing about 262 equations and 317 variables in terms of bit operations. The complexity of collecting messages stage is $2^{k_1+q_2+q_3-231} + 2^{45+q_2-1}$, where $\mathtt{E}[k_1] \leq 225$.

In searching stage, there are 2^{83} different internal differences after the χ mapping of the fifth round. These internal differences actually compose a 83-

dimensional affine space (denoted as V) over $\mathbb{F}_2$. The projection of the affine space $L \circ \iota(V)$ on the first 10 lanes consists of internal differences which are projected to their first 10 lanes, and its dimension is 80. So there are 2^{80} different collision subsets. The geometric expected size of collision subset is $2^{296.58}$, arithmetic expected size is $2^{294.16}$. To find a collision, the number of outputs we need to search is $2^{q_2} = 2^{(296.58+80)/2} = 2^{188.29}$. The total complexity is $2^{118.22} + 2^{k_1+q_3-42.71} + 2^{45+188.29-1} = 2^{232.29}$.

6.3 Collision Attacks on 4-round SHA3-512 and 5-round SHA3-224/SHA3-256/SHAKE128/SHAKE256

For 4-round `SHA3-512`, we use the 2.5-round internal difference characteristic given in Characteristic 2 in [17]. The transition condition numbers of the characteristic are $(k_1, k_2, k_3) = (16, 16, 170)$. For $k_3 = 170$, there are 2^{170} different internal differences after the χ mapping of the third round. The dimension of the projection of the affine space composed of 2^{170} internal differences on the first 10 lanes is 156. For $d = 512$, we collect the messages with the same internal difference in the first 10 lanes into a set. So there are 2^{156} different collision subsets. We randomly select 2^{30} different collision subsets and calculate the average of their expected size, which is $2^{296.06}$. This is closer to the geometric expected size $2^{296.58}$ compared to the arithmetic expected size $2^{294.16}$ for the collision length of 8 lanes. We use the geometric expected size to estimate complexity.

Since the size of collision subset is bounded by $2^{296.58}$, we need to choose $2^{(296.58+156)/2} = 2^{226.29}$ messages, which conforming the first two rounds internal differential characteristic. With the conditional differential technique, we can directly select the initial message that can pass the first two rounds functions at the cost of negligible time complexity. We calculate the CSS of two messages simultaneously as in Section 6.2. The time complexity of calculating the output of all messages after 4-round permutation is bounded by $2^{226.29-1} = 2^{225.29}$.

For 5-round `SHA3-224/SHA3-256/SHAKE128/SHAKE256`, we use internal differential characteristic given in Characteristic 3 in [17]. The transition condition numbers of the characteristic are $(k_2, k_3, k_4) = (21, 18, 16)$. We evaluated the average expected sizes of 2^{16} collision subsets, which are $2^{134.34}$ for $d = 256$ and $2^{276.56}$ for $d = 512$. We also use conditional differential technique and calculate the CSS of two messages simultaneously to reduce time complexity. For an output length of 256 bits, the complexity of our collision attack is $2^{18+(16+134.34)/2-1} = 2^{96.67}$. When the output length is 512 bits, the complexity is $2^{18+(16+276.56)/2-1} = 2^{163.28}$.

6.4 Summary of Collision Attacks

We summarize our collision attacks in Table 5. For 4-round `SHA3-512` and 5-round `SHAKE256`, we use the same characteristic in [17]. For 5-round `SHA3-384`, the last two rounds of its characteristic is the same as characteristic of 5-round `SHAKE256`. We slightly change the input difference of the second χ mapping to obtain a target internal difference with less active Sboxes. For 6-round `SHAKE256`, we use MILP to search for characteristic starting from the third round with the objective of minizing $(k_3 + k_4 + k_5/2)$. Then, the characteristic are extended one round forward to obtain the output difference of the first χ with fewer active Sboxes.

Target	n_r	i	k_1	k_2	k_3	k_4	k_5	Complexity ($\log_2$)
`SHA3-512`	4	32	16	16	170	-	-	225.29
`SHA3-224/SHA3-256/SHAKE128`	5	32	-	21	18	16	-	96.67
`SHA3-384`	5	32	-	25	18	16	-	170.73[†]
`SHAKE256`	5	32	-	21	18	16	-	163.28
`SHAKE256`	6	32	-	31	25	20	83	232.29

[†] Inspired by the reviewers, we reduced the time complexity from $2^{172.19}$ to $2^{170.73}$

Table 5: The parameters of characteristics and complexities

7 Experiments

In order to illustrate the probabilistic linearization technique, and to validate the new attack boundary, we implemented an actual collision attack on a reduced version of Keccak, called Keccak$[240, 160, 5, 96]$. The internal state size of Keccak$[240, 160, 5, 96]$ is 400 bits, with the first 240 bits being the rate part and the remaining 160 bits constituting the capacity part. The underlying permutation is reduced to 5 rounds, and the length of the digest is set to 96 bits.

In our collision attack on this Keccak instance, the basic framework aligns with the analysis of 5-round `SHA3-384` as shown in Section 6.1. We select 2-block messages for the collisions. Initially, we find a 2.5-round internal differential characteristic from round 1 to round 3.5 through MILP, see also Characteristic 3 in Supplementary Material B. The parameters are $(k_2, k_3, k_4) = (14, 8, 7)$, and there are 18 active Sboxes and 22 non-active Sboxes for $\alpha_1^\star$. In the TIDA stage, we set 1 linear equation for each active Sbox and 5 linear equations for each non-active Sbox, and obtain $18+22\times5 = 128$ equations in input difference system. For this system, 21 equations are in the inner part. In other words, for each random message of the first block M_0, it has a solution with a probability $p_0 = 2^{-21}$. By applying the Procedure PIDS, the probability $p_1^\star$ that a randomly selected difference in its solution space happens to be the input difference turns out to be $2^{-15.30}$. This means that we randomly generate at most $2^{21+15.30} = 2^{36.30}$ times for the first block M_0, and the input difference to the nonlinear operation of the first round can be obtained. In the experiment, we randomly selected 2^{30} M_0 and obtained 2^9 input difference systems with solutions, which verifies $p_0 = 2^{-21}$. Among them, about 2^2 input difference systems E_Δ have solution spaces that contain input differences. Therefore the real value of the probability p_1 is 2^{-7} and it is much higher than its theoretical lower bound $p_1^\star = 2^{-15.30}$ (see also the analysis of TIDA in Section 4.3). Then, as shown in steps 15 to 20 of Algorithm 1, for each input difference β_0 in the input difference system E_{Δ}, we calculate its differential transition system $E_{\beta_0 \to \alpha_1^\star}(\mathtt{R}^5(M_0) \oplus M_1)$ and solve it on M_1. On average, for about $2^2 \sim 2^3$ trials of input difference systems E_{Δ}, there is one correct input difference β_0 such that the corresponding differential transition system $E_{\beta_0 \to \alpha_1^\star}$ has solutions. The size of the solution space W_1 for the system $E_{\beta_0 \to \alpha_1^\star}$ is $2^{240/2-k_1}$, where $\mathtt{E}(k_1) = 62$. In the experiments, we ran Algorithm 1 multiple times and found a space W_1 of size 2^{66}. This solution space alone can provide enough initial messages to generate collisions.

In the collecting stage and searching stage, we use the techniques of conditional internal differentials to satisfy all the differential transition conditions in the first round together with 9 differential transition conditions in the second round. The total time complexity is $2^{k_2-1-9+k_3+(k_4+54)/2} = 2^{42.5}$, where $2^{54} = 2^{216/4}$ is the geometric expected size of the collision subset (see also Table 4). If we use arithmetic expected size to predict time, the complexity is $2^{k_2-1-9+k_3+(k_4+53.425)/2} = 2^{42.21}$. On a desktop with an Intel Core i9-13900KF processor, it takes about 17 hours to find a 96-bit collision, and consumes about 48G of memory. The total number of the searched messages is around 2^{43}, which matches the theoretical complexity. A concrete example is given in Supplementary Material E.

8 Conclusions

In this paper, we presented collision attacks on up to 6 rounds against four `SHA-3` variants by 1-round internal connectors. We equivalently transform the construction of internal connectors into the internal connectivity problem, and reduce this nonlinear problem to two types of linear systems: input difference system and differential transition system. The TIDA is redesigned to construct and solve these linear systems by probabilistic linearization method. In addition, we search for new internal differential characteristics and more accurately estimate the time complexity of searching stage. Compared to previous ways of constructing connectors, the probabilistic linearization method neither loses too many input differences, nor depends on the solver of nonlinear system. For 4-round `SHA3-512` and 5-round `SHAKE256`, our collision attack outperforms the best known attacks, and the collision attacks on 5-round `SHA3-384` and 6-round `SHAKE256` are presented for the first time.

The further work includes bounding the complexity of this algorithm and building new input difference system for other `SHA-3` instances with higher security strength, such as `SHA3-512`. Additionally, in some cases it may be helpful to use solvers such as CryptoMiniSAT [16] or develop fast solvers to directly solve the connectivity problem.

We stress that our attack does not threaten the security of the full `SHA-3`.

Acknowledgements. We thank the anonymous reviewers from CRYPTO 2024 for their thoughtful comments and insightful questions. Inspired by the reviewers, we were able to optimize the collision attack on 5-round `SHA3-384`.

References

1. Bernstein, D.J.: Second preimages for 6 (7?(8??)) rounds of keccak. NIST mailing list (2010)
2. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Keccak. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 313–314. Springer (2013), https://doi.org/10.1007/978-3-642-38348-9_19
3. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Keccak sponge function family main document. Submission to NIST (Round 2) **3**(30), 320–337 (2009)
4. Chang, D., Kumar, A., Morawiecki, P., Sanadhya, S.K.: 1st and 2nd Preimage Attacks on 7, 8 and 9 Rounds of Keccak-224,256,384,512. In: SHA-3 workshop (2014)
5. Dinur, I., Dunkelman, O., Shamir, A.: New attacks on keccak-224 and keccak-256. In: Canteaut, A. (ed.) Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7549, pp. 442–461. Springer (2012), https://doi.org/10.1007/978-3-642-34047-5_25
6. Dinur, I., Dunkelman, O., Shamir, A.: Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In: Moriai, S. (ed.) Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8424, pp. 219–240. Springer (2013), https://doi.org/10.1007/978-3-662-43933-3_12
7. Dinur, I., Dunkelman, O., Shamir, A.: Improved practical attacks on round-reduced keccak. J. Cryptol. **27**(2), 183–209 (2014). https://doi.org/10.1007/s00145-012-9142-5
8. Dworkin, M.J.: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015). https://doi.org/10.6028/nist.fips.202
9. Guo, J., Liao, G., Liu, G., Liu, M., Qiao, K., Song, L.: Practical Collision Attacks against Round-Reduced SHA-3. J. Cryptol. **33**(1), 228–270 (2020). https://doi.org/10.1007/s00145-019-09313-3
10. Guo, J., Liu, G., Song, L., Tu, Y.: Exploring SAT for cryptanalysis: (quantum) collision attacks against 6-round SHA-3. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13793, pp. 645–674. Springer (2022), https://doi.org/10.1007/978-3-031-22969-5_22
11. Guo, J., Liu, M., Song, L.: Linear structures: Applications to cryptanalysis of round-reduced keccak. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 249–274 (2016), https://doi.org/10.1007/978-3-662-53887-6_9
12. Huang, S., Ben-Yehuda, O.A., Dunkelman, O., Maximov, A.: Finding collisions against 4-round SHA-3-384 in practical time. IACR Trans. Symmetric Cryptol. **2022**(3), 239–270 (2022). https://doi.org/10.46586/tosc.v2022.i3.239-270
13. Peyrin, T.: Improved differential attacks for ECHO and grøstl. In: Rabin, T. (ed.) Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference,

Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6223, pp. 370–392. Springer (2010), https://doi.org/10.1007/978-3-642-14623-7_20

14. Qiao, K., Song, L., Liu, M., Guo, J.: New collision attacks on round-reduced keccak. In: Coron, J., Nielsen, J.B. (eds.) Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10212, pp. 216–243 (2017), https://doi.org/10.1007/978-3-319-56617-7_8
15. Song, L., Liao, G., Guo, J.: Non-full Sbox Linearization: Applications to Collision Attacks on Round-Reduced Keccak. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10402, pp. 428–451. Springer (2017), https://doi.org/10.1007/978-3-319-63715-0_15
16. Soos, M., Nohl, K., Castelluccia, C.: Extending SAT solvers to cryptographic problems. In: Kullmann, O. (ed.) Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5584, pp. 244–257. Springer (2009), https://doi.org/10.1007/978-3-642-02777-2_24
17. Zhang, Z., Hou, C., Liu, M.: Collision attacks on round-reduced SHA-3 using conditional internal differentials. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14007, pp. 220–251. Springer (2023), https://doi.org/10.1007/978-3-031-30634-1_8

Supplementary Material

A Greedy Probabilistic Input Difference System

Procedure GreedyPIDS($\alpha_1^\star$, MDST)

```
Input: Internal difference $\alpha_1^\star$ and MDST.
Output: Linear equation system $E_\Delta$, probability $p_1^\star$.
1  Set $E_\Delta = \emptyset, E'_\Delta = \emptyset, E_\mathcal{C}^\star = \emptyset, p_1^\star = 1$ and $W = \Delta(L(\Delta_\mathcal{R}, \Delta_\mathcal{C}))$.
   /* $W$ is a variable vector $(w_0, ..., w_{799})$, $w_i = w_i(\Delta_\mathcal{R}, \Delta_\mathcal{C})$ is a linear
      function about $\Delta_\mathcal{R} = (\delta_0, ..., \delta_{799-p-c/2}), \Delta_\mathcal{C} = (\delta_{800-p-c/2}, ..., \delta_{799})$   */
2  for each integer $j \in [0, 160)$ do
3      Get the output difference $\delta_{out}$ of the $j$-th Sbox from $\alpha_1^\star$.
4      if $\delta_{out} = 0$ then
5          $E_\Delta = E_\Delta \cup \{w_{5j}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = 0, \ldots, w_{5j+4}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = 0\}$.
6      end
7  end
8  Reduce $E_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = E_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) \cup E_\mathcal{C}(\Delta_\mathcal{C})$, $E_\Delta = E_\mathcal{R}$, $E_\mathcal{C}^\star = E_\mathcal{C}$.
9  for each integer $j \in [0, 160)$ do
10     Get the output difference $\delta_{out}$ of the $j$-th Sbox from $\alpha_1^\star$.
11     if $\delta_{out} \neq 0$ then
12         do
13             $r_1 = rank(E_\Delta)$.
14             Select and delete a subspace Ker$(\sum_{j=0}^{4} l_j \cdot 2^j, q)$ from MDST[$\delta_{out}$].
15             $E'_\Delta = E_\Delta \bigcup \{l_0 \cdot w_{5j}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) + \cdots + l_4 \cdot w_{5j+4}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = q\}$.
16             Reduce $E'_\Delta(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = E'_\mathcal{R}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) \cup E'_\mathcal{C}(\Delta_\mathcal{C})$, $E'_\Delta = E'_\mathcal{R}$.
17             $r_2 = rank(E'_\Delta)$.
18             if $r_2 > r_1$ then
19                 $E_\Delta = E'_\Delta$.
20             else if MDST[$\delta_{out}$] $= \emptyset$ then
21                 $E_\mathcal{C}^\star = E_\mathcal{C}^\star \bigcup \{l_0 \cdot w_{5j}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) + \cdots + l_4 \cdot w_{5j+4}(\Delta_\mathcal{R}, \Delta_\mathcal{C}) = q\}$.
22             end
23         while $r_2 = r_1$ and MDST[$\delta_{out}$] $\neq \emptyset$;
24     end
25 end
26 $E_\Delta = E_\Delta \bigcup E_\mathcal{C}^\star$.
27 return $(E_\Delta, p_1^\star)$
```

B Internal Differential Characteristics

In this section, we provided the internal differential characteristics (labeled as Characteristic 1-3) which we use in our collision attacks. The internal difference

$[i, v]$ is represented by its canonical representative state defined in Section 3.3. Each state is given as a matrix of 5×5 lanes of 64 bits, order from left to right, where each lane is given in hexadecimal using the little-endian format. The symbol '-' is used in order to denote a zero 4-bit value.

```
|----------a---4b|--------8184--2-|--------8496--31|--------8782---c|--------8382--1a|
|--------8-a---49|---------194--68|--------8-94--31|--------8682---4|--------8382--1a|
|--------8-a---49|---------184--6-|--------8484--31|--------8782---4|--------8382--1a|  R1
|--------8-a----9|---------18---6-|--------8494--31|--------8382---4|--------8382--1a|
|--------8-8---49|---------184--6-|--------849---35|--------8782---4|--------838---18|
```

$\downarrow L$

```
|--------8------2|------------8--1|--------8-------|------------8---|--------8---8---|
|--------8-------|----------------|----------------|--------8-------|--------8---8---|
|--------------81|---------------1|----------------|----------------|--------------8-|
|----------------|----------------|----------------|----------------|----------------|
|------------8---|------------8---|----------------|------------8---|----------------|
```

$\downarrow \chi \ \ (p = 2^{-25})$

```
|--------8------2|---------------1|----------------|----------------|------------8---|
|--------8-------|----------------|----------------|----------------|------------8---|
|--------------8-|---------------1|----------------|----------------|----------------|
|----------------|----------------|----------------|----------------|----------------|
|------------8---|----------------|----------------|----------------|----------------|
```

$\downarrow \iota$

```
|--------8---8-8-|---------------1|----------------|----------------|------------8---|
|--------8-------|----------------|----------------|----------------|------------8---|
|--------------8-|---------------1|----------------|----------------|----------------|  R2
|----------------|----------------|----------------|----------------|----------------|
|------------8---|----------------|----------------|----------------|----------------|
```

$\downarrow L$

```
|--------8---8-8-|----------------|----------------|----------------|----------------|
|----------------|---------------8|-------------4--|----------------|----------------|
|---------------2|----------------|----------------|----------------|---------------2|
|-------------4--|---------------8|-------------4--|----------------|----------------|
|----------------|----------------|----------------|----------------|----------------|
```

$\downarrow \chi \ \ (p = 2^{-18})$

```
|--------8---8-8-|----------------|----------------|----------------|----------------|
|---------------8|---------------8|-------------4--|----------------|----------------|
|---------------2|----------------|----------------|----------------|----------------|
|----------------|---------------8|-------------4--|----------------|----------------|
|----------------|----------------|----------------|----------------|----------------|
```

$\downarrow \iota$

```
|---------------a|----------------|----------------|----------------|----------------|
|---------------8|---------------8|-------------4--|----------------|----------------|
|---------------2|----------------|----------------|----------------|----------------|  R3
|----------------|---------------8|-------------4--|----------------|----------------|
|----------------|----------------|----------------|----------------|----------------|
```

$\downarrow L$

```
|---------------a|------------8---|----------------|----------------|----------------|
|----------------|----------------|--------------1-|-----------1----|----------------|
|----------------|-----------1----|----------------|----------------|----------------|  R3.5
|----------------|--------------8-|----------------|---------2------|----------------|
|----------------|----------------|----------------|----------------|----------------|
```

The characteristic has a period of $i = 32$ for the 5-round attack on `SHA3-384`, as described in Section 6.1.

Characteristic 1: The 1-3.5 round internal differential characteristic with probability 2^{-43} and $k_4 = 16$.

```
|--------c2-----2|--------381--4-2|--------58-2-2-2|--------3----2--|--------3-2----d|
|--------42-----1|--------38---4-2|--------58-2---2|--------38-----8|--------3-2----5|
|--------4-1----1|--------381--4-2|--------58-----2|--------38---2-8|--------32-----d|  R1
|---------2-----5|--------781----2|--------58-----2|--------38---2-8|--------3-2----d|
|--------42-----1|--------781--4-2|--------5c-----2|--------38---2-8|--------3-2----d|
```

$$\downarrow L$$

```
|--------8------3|---------------1|----------------|----------------|----------------|
|--------8-8-----|----------8-----|--------1-8-----|----------8--8--|----------8-----|
|----------------|----------8-----|----------------|----------------|----------------|
|----------------|----------------|----------------|----------------|----------------|
|------------8-8-|---------------1|--------1------1|-------------88-|---------------1|
```

$$\downarrow \iota \circ \chi \ \ (p = 2^{-31})$$

```
|--------8---8-8-|---------------1|----------------|----------------|----------------|
|--------8-------|----------8-----|--------1-------|-------------8--|----------------|
|----------------|----------8-----|----------------|----------------|----------------|  R2
|----------------|----------------|----------------|----------------|----------------|
|------------8-8-|---------------1|--------1-------|-------------8--|----------------|
```

$$\downarrow L$$

```
|--------8---8-8-|---------------8|----------------|----------------|----------------|
|----------------|----------------|----------------|----------------|---------2------|
|---------------2|---------------4|----------------|----------------|---------2-----2|
|----------------|---------------8|---------------2|----------------|---------------8|
|----------------|---------------4|----------------|----------------|---------------4|
```

$$\downarrow \iota \circ \chi \ \ (p = 2^{-25})$$

```
|---------------2|---------------8|----------------|----------------|----------------|
|----------------|----------------|----------------|----------------|---------2------|
|---------------2|---------------4|---------------2|----------------|---------2------|  R3
|----------------|---------------8|---------------2|----------------|----------------|
|----------------|---------------4|----------------|----------------|----------------|
```

$$\downarrow L$$

```
|---------------2|----------------|------------1---|----------------|----------------|
|----------------|------------2---|--------------1-|-----------1----|----------------|
|--------------1-|----------------|----------------|----------------|----------------|
|----------------|----------------|------------1---|-----------1----|----------------|
|----------------|----------------|---------------1|----------------|--------------1-|
```

$$\downarrow \iota \circ \chi \ \ (p = 2^{-20})$$

```
|------------8--2|----------------|------------1---|----------------|----------------|
|--------------1-|------------2---|--------------1-|-----------1----|----------------|
|--------------1-|----------------|----------------|----------------|--------------1-|  R4
|----------------|----------------|------------1---|-----------1----|----------------|
|----------------|----------------|--------------11|----------------|--------------1-|
```

$$\downarrow L$$

```
|------------c--2|---------a------|---------1------|----------2---2-|--------4--5----|
|--------1-------|----------4---1-|-----------2--8-|--------1-------|--------2----4-2|
|-----------1----|-----------8-4--|---------2------|---------1---4--|---------------1|  R4.5
|--------2----8--|-----------4-1--|---------2------|--------18------|---------1------|
|-------------c--|----------8---8-|----------8--a--|----------8-----|-----------2----|
```

The characteristic has a period of $i = 32$ for the 6-round attack on `SHAKE256`, as described in Section 6.2.

Characteristic 2: The 1-4.5 round internal differential characteristic with probability 2^{-76} and $k_5 = 83$.

```
|---2|---6|---4|---1|----|
|---8|----|---4|---1|----|
|---8|---2|---1|---1|----|  R1
|---8|---2|---4|----|--2-|
|---a|---2|---4|---1|--2-|
```

$\downarrow L$

```
|---a|--2-|--28|--2-|---8|
|----|----|----|----|----|
|---8|----|----|--2-|---8|
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow \chi \ \ (p = 2^{-14})$

```
|---a|----|----|--2-|----|
|----|----|----|----|----|
|---8|----|----|--2-|----|
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow \iota$

```
|---8|----|----|--2-|----|
|----|----|----|----|----|
|---8|----|----|--2-|----|  R2
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow L$

```
|---8|----|----|----|----|
|---2|----|--4-|----|----|
|----|----|--4-|----|----|
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow \chi \ \ (p = 2^{-8})$

```
|---8|----|----|----|----|
|---2|----|--4-|----|----|
|----|----|--4-|----|----|
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow \iota$

```
|---2|----|----|----|----|
|---2|----|--4-|----|----|
|----|----|--4-|----|----|  R3
|----|----|----|----|----|
|----|----|----|----|----|
```

$\downarrow L$

```
|---2|----|---2|----|----|
|----|----|----|----|----|
|----|--1-|----|----|----|  R3.5
|----|--2-|----|----|----|
|----|----|----|----|----|
```

The characteristic has a period of $i = 8$ for the 5-round attack on Keccak$[240, 160]$, as described in Section 7.

Characteristic 3: The 1-3.5 round internal differential characteristic with probability 2^{-22} and $k_4 = 7$.

C Maximum Difference Density Subspace for SHA3-384

In.	δ_{out}	U	P	In.	δ_{out}	U	P
0	0x04	(0x04, 1)	9/16	1	0x11	(0x05, 1)	9/16
2	0x08	(0x08, 1)	9/16	3	0x19	(0x01,1),(0x06,0)	5/8
4	0x14	(0x04,1),(0x06,0)	5/8	5	0x06	(0x14, 1)	9/16
6	0x01	(0x01, 1)	9/16	17	0x1c	(0x04,1),(0x1b,0)	5/8
18	0x06	(0x14, 1)	9/16	20	0x04	(0x04, 1)	9/16
21	0x01	(0x01, 1)	9/16	23	0x1f	(0x07,1),(0x1e,0)	6/8
24	0x1a	-	12/32	25	0x18	(0x12, 1)	9/16
26	0x0c	(0x04, 1)	6/16	31	0x1e	(0x16, 0)	9/16
32	0x05	(0x04, 1)	7/16	33	0x10	(0x04, 1)	6/16
34	0x08	(0x08, 1)	9/16	35	0x13	(0x10,1),(0x0f,0)	5/8
36	0x14	(0x04,1),(0x19,0)	5/8	37	0x06	(0x14, 1)	9/16
38	0x03	(0x0a, 1)	9/16	49	0x18	(0x12, 1)	9/16
50	0x06	(0x02, 1)	6/16	52	0x06	(0x14, 1)	9/16
53	0x01	(0x04, 0)	6/16	55	0x1f	(0x07, 1)	8/16
56	0x12	(0x02, 1)	7/16	57	0x18	(0x12, 1)	9/16
58	0x08	(0x02, 1)	6/16	63	0x1d	(0x0d, 0)	9/16
64	0x05	(0x11,0),(0x0e,0)	5/8	65	0x10	(0x02, 0)	6/16
66	0x08	(0x08, 1)	9/16	67	0x11	(0x05, 1)	9/16
68	0x14	-	10/32	69	0x06	(0x14, 1)	9/16
70	0x03	(0x01, 1)	6/16	81	0x18	(0x12, 1)	9/16
82	0x06	(0x02, 1)	6/16	85	0x01	(0x01, 1)	9/16
87	0x1f	(0x07, 1)	8/16	88	0x1a	-	12/32
89	0x18	(0x12, 1)	9/16	90	0x0c	(0x09, 1)	9/16
95	0x1d	-	11/32	96	0x05	(0x04, 1)	7/16
97	0x10	(0x04, 1)	6/16	98	0x08	(0x04, 1)	6/16
99	0x11	(0x05, 1)	9/16	100	0x14	(0x04, 1)	7/16
101	0x06	(0x02, 1)	6/16	102	0x02	(0x02, 1)	9/16
113	0x18	(0x12, 1)	9/16	114	0x04	(0x04, 1)	9/16
116	0x04	(0x04, 1)	9/16	117	0x01	(0x01, 1)	9/16
119	0x1f	(0x13, 1)	8/16	120	0x1a	(0x02, 1)	9/16
121	0x18	(0x12, 1)	9/16	122	0x04	(0x04, 1)	9/16
127	0x1d	(0x0d, 0)	9/16	128	0x05	(0x06, 0)	7/16
130	0x0c	(0x09, 1)	9/16	131	0x11	(0x05, 1)	9/16
132	0x14	-	10/32	133	0x06	(0x14, 1)	9/16
134	0x03	(0x0a, 1)	9/16	145	0x08	(0x08, 1)	9/16
146	0x02	(0x02, 1)	9/16	148	0x04	(0x02, 1)	6/16
151	0x1f	(0x07, 1)	8/16	152	0x1a	-	12/32
153	0x18	(0x12, 1)	9/16	154	0x0c	(0x09, 1)	9/16
159	0x1d	-	11/32				

D Maximum Difference Density Subspace for SHAKE256

Index	δ_{out}	U	E$[k_U]$	Index	δ_{out}	U	E$[k_U]$
0	0x11	(0x05, 1)	30/9	1	0x07	(0x01, 1)	27/7
2	0x10	(0x10, 1)	30/9	3	0x10	(0x10, 1)	30/9
9	0x0c	(0x09, 1)	30/9	10	0x02	(0x02, 1)	30/9
17	0x04	(0x04, 1)	30/9	20	0x02	(0x02, 1)	30/9
21	0x10	(0x10, 1)	30/9	25	0x01	(0x01, 1)	30/9
27	0x06	(0x14, 1)	30/9	28	0x1e	(0x16, 0)	33/9
29	0x1a	(0x10, 1)	33/9	30	0x05	(0x01, 1)	27/7
31	0x01	(0x01, 1)	30/9	32	0x11	(0x05, 1)	30/9
33	0x06	(0x14, 1)	30/9	34	0x10	(0x10, 1)	30/9
35	0x08	(0x08, 1)	30/9	42	0x02	(0x02, 1)	30/9
49	0x04	(0x04, 1)	30/9	53	0x10	(0x10, 1)	30/9
57	0x01	(0x01, 1)	30/9	59	0x0e	(0x02, 1)	27/7
60	0x1e	(0x16, 0)	33/9	61	0x1a	(0x10, 1)	33/9
62	0x05	(0x01, 1)	27/7	64	0x11	(0x05, 1)	30/9
65	0x06	(0x14, 1)	30/9	66	0x10	(0x10, 1)	30/9
67	0x18	(0x12, 1)	30/9	73	0x08	(0x08, 1)	30/9
74	0x02	(0x02, 1)	30/9	84	0x03	(0x0a, 1)	30/9
89	0x10	(0x10, 1)	30/9	91	0x0e	(0x02, 1)	27/7
92	0x1e	(0x16, 0)	33/9	93	0x1a	(0x10, 1)	33/9
94	0x05	(0x01, 1)	27/7	96	0x11	(0x05, 1)	30/9
97	0x06	(0x14, 1)	30/9	98	0x11	(0x05, 1)	30/9
99	0x18	(0x12, 1)	30/9	105	0x08	(0x08, 1)	30/9
116	0x02	(0x02, 1)	30/9	117	0x10	(0x10, 1)	30/9
121	0x01	(0x01, 1)	30/9	123	0x0e	(0x02, 1)	27/7
124	0x1e	(0x16, 0)	33/9	125	0x1a	(0x10, 1)	33/9
126	0x06	(0x14, 1)	30/9	128	0x11	(0x05, 1)	30/9
129	0x06	(0x14, 1)	30/9	130	0x10	(0x10, 1)	30/9
131	0x18	(0x12, 1)	30/9	137	0x08	(0x08, 1)	30/9
138	0x02	(0x02, 1)	30/9	148	0x02	(0x02, 1)	30/9
149	0x10	(0x10, 1)	30/9	153	0x01	(0x01, 1)	30/9
154	0x04	(0x04, 1)	30/9	155	0x0e	(0x02, 1)	27/7
156	0x1e	(0x16, 0)	33/9	157	0x1a	(0x10, 1)	33/9
158	0x07	(0x01, 1)	27/7				

E An Example of the Collision

```
                |6a44|1a51|321-|23f2|66f6|
                |41-9|7d57|a1b2|7d8c|1a7a|
        M_0 =   |6d-f|c9aa|d211|b134|f229|
                |----|----|----|----|----|
                |----|----|----|----|----|
                          ↓R^5
                |169d|-51e|5252|7437|dc49|
                |629a|9762|e2--|8eae|5a93|
   R^5(M_0) =   |67-5|8-dc|c829|f4c4|-aea|
                |fbff|bb15|95ee|3b2f|7b41|
                |--93|54ba|a9ce|5e4a|4779|

       |b9a5|-d74|35f2|51be|f816|  |617d|86ff|df18|fa15|9876|
       |9a53|7251|e164|ebe5|482b|  |8a43|e4c7|ca4f|cdc3|482b|
M_1 =  |5--3|14a7|e3cd|-b83|8614|  |683b|2c9f|d2fc|c54d|8c1e|  = M_1'
       |----|----|----|----|----|  |----|----|----|----|----|
       |----|----|----|----|----|  |----|----|----|----|----|
                 ↓⊕R^5(M_0)                  ↓⊕R^5(M_0)
       |af38|-86a|67a-|2589|245f|  |77e-|83e1|8d4a|8e22|443f|
       |f8c9|e533|-364|654b|12b8|  |e8d9|73a5|284f|436d|12b8|
       |37-6|947b|2be4|ff47|8cfe|  |-f3e|ac43|1ad5|3189|86f4|
       |fbff|bb15|95ee|3b2f|7b41|  |fbff|bb15|95ee|3b2f|7b41|
       |--93|54ba|a9ce|5e4a|4779|  |--93|54ba|a9ce|5e4a|4779|
                 ↓R^5                        ↓R^5
       |5992|37b4|27ce|9981|b9eb|  |5992|37b4|27ce|9981|b9eb|
       |e7e5|81a7|eafc|9a8e|6ef8|  |e7e5|3197|37a5|-f1b|25b3|
       |e4a9|8b81|8264|187b|e9e9|  |9ed2|fdb-|2baf|4665|a9a9|
       |7826|9f-9|a72d|e5bf|3e62|  |ece4|e96d|d75f|--58|7e34|
       |-ba6|f6d7|db68|84ce|7744|  |8ba7|-cad|997a|--d1|6af4|
```

Collision: A collision $(M_0||M_1)$ and $(M_0||M_1')$ in KECCAK$[r = 240, c = 160, n_r = 5, d = 96]$.

F Advantages of Adding More Equations to One Sbox

Compared with adding one equation on the input difference for each active Sbox, the probability $p_1^\star$ will slightly increase without changing the total number of equations if the number of equations for each active Sbox is allowed to any number among 0, 1, 2, 3.[1] For simplification, we consider two scenarios: the groups of two active Sboxes and the groups of three.

Denoted by $p_{i,w}$ the difference density of Sbox i for adding w equations. All the 31 nonzero δ_{out}'s can be classified into 5 categories, according to their difference density profile $(p_{\cdot,0}, p_{\cdot,1}, p_{\cdot,2}, p_{\cdot,3})$, as listed in Table 6.

Category	$p_{\cdot,0}$	$p_{\cdot,1}$	$p_{\cdot,2}$	$p_{\cdot,3}$	$p_{\cdot,0}/p_{\cdot,1}$	$p_{\cdot,2}/p_{\cdot,1}$	$p_{\cdot,3}/p_{\cdot,1}$
C1	1.830075	0.830075	0.415037	0	-1	0.415038	0.830075
C2	1.540568	0.830075	0.192645	0	-0.710493	0.63743	0.830075
C3	1.415037	0.830075	0.192645	0	-0.584962	0.63743	0.830075
C4	1.540568	1	0.415037	0	-0.540568	0.584963	1
C5	1.678072	1.192645	0.678072	0	-0.485427	0.514573	1.192645

Table 6: The Difference Density Profile of 5 Categories of δ_{out}'s ($-\log_2$)

For a group of two active Sboxes, we compare one 5-dimensional and one 3-dimensional affine subspace with two 4-dimensional affine subspaces. Let

$$a_2 = \log_2(p_{i,0}/p_{i,1}) + \log_2(p_{j,2}/p_{j,1})$$

be the advantage of adding two equations to Sbox j and no equations to Sbox i. If $a_2 > 0$, then adding two equations to Sbox j (and no equations to Sbox i) is better than adding one equation to each of these two Sboxes. Table 7 lists the advantages of adding two equations to one Sbox greater than 0.

2 Eqs. / 0 Eqs.	C2 or C3	C4	C5
C3	0.052468		
C4	0.096862	0.044395	
C5	0.152003	0.099536	0.029146

Table 7: The Advantage a_2 of Adding Two Equations to One Sbox

For the case of three Sboxes, if three equations are imposed to one Sbox (to say k), we have the advantage

$$a_3 = \log_2(p_{i,0}/p_{i,1}) + \log_2(p_{j,0}/p_{j,1}) + \log_2(p_{k,3}/p_{k,1}).$$

If $a_3 > 0$, then adding three equations to Sbox k (and no equations to Sboxes i and j) is better. By an exhaustive calculation with taking Table 7 into account,

[1] The setting of the number of equations is suggested by the reviewers.

0 Eqs. \ 3 Eqs.	C5
(C4,C4)	0.111509
(C4,C5)	0.166650
(C5,C5)	0.221791

Table 8: The Advantage a_3 of Adding Three Equations to One Sbox

Category	Number of Equations
C1	1
C2	1,2
C3	0,1,2
C4	0,1,2
C5	0,1,2,3

Table 9: Candidates for Optimal Number of Equations

we can conclude that adding three equations to one Sbox (that is, full linearization of one Sbox for every three Sboxes) is optimal only in two cases: a) the number n_5 of δ_{out}'s from Category 5 (this is exactly the category of δ_{out}'s mentioned by the reviewer) is larger than the total number $(n_2 + n_3 + n_4)$ of δ_{out}'s from Category 2, 3, and 4, and it happens for at most $(n_5 - (n_2 + n_3 + n_4))/3$ triples (C5,C5,C5); b) all Sboxes have been handled but three Sboxes with one or two δ_{out}'s from Category 5 and the rest from Category 4, that is, a single triple (C4,C4,C5) or (C4,C5,C5). Their advantages are listed in Table 8.

According to Tables 7 and 8, the candidates for optimal number of equations for probabilistic linearization are listed in Table 9.

From the above analysis, it is evident that the probability of $p_1^\star$ can possibly be optimized by a factor of $2^{160\times 0.152/2} = 2^{12.16}$. However, it does not affect the complexity of our collision attacks on reduced SHA-3 in Section 6.

For the collision attack on 6-round SHAKE256, the original probability of $p_1^\star$ is used to evaluate the lower bound of p_1 and derive an upper bound on the complexity of TIDA stage, which is much lower than those of collecting messages and searching stages, and thus the optimized probability of $p_1^\star$ does not affect the total complexity.

For the collision attack on 5-round SHA3-384, given $\alpha_1^\star$ in Section 6.1, the probability of $p_1^\star$ is $2^{-69.48}$ using PIDS. It turns out to be $2^{-67.70}$ if using the optimized probabilistic linearization, improving $p_1^\star$ by a factor of around $2^{1.78}$. But the probability $p_1^\star$ is not the dominant term of the complexity either. For setting up an input difference system with inner part as small as possible, we adopt 4-dimensional affine subspaces with both maximal and submaximal difference density in GreedyPIDS (see also Line 11 of Page 17). As a result, while the probability $p_1^\star$ produced by GreedyPIDS decreases to $2^{-76.50}$ from $2^{-67.70}$, the size t_2 of the inner system is reduced to 97 from 116. In addition, $p_1^\star$ gives a lower bound of p_1, but a low probability $p_1^\star$ does not imply a low probability p_1. The probability p_1 turns out to be $2^{-64.61}$ as estimated in Page 26, which is markedly larger than $p_1^\star = 2^{-76.50}$.

In either case, the optimization of $p_1^\star$ is not our primary goal as mentioned previously. However, we further expand MDST with 3-dimensional affine subspaces in GreedyPIDS, and cut down the attack complexity to $2^{170.73}$, by a factor of $2^{-1.71}$, for 5-round SHA3-384. In this new attack, t_2 is reduced to 90 from 97, while $p_1^\star = 2^{-76.97}$ and $p_1 = 2^{-70.15}$. The other parameters are the same. We can see that the improved attack is achieved by optimizing the size of the inner system rather than $p_1^\star$. Still, we can add more equations to the input difference system without changing the inner part to increase $p_1^\star$, but the corresponding p_1 becomes smaller. In Supplementary Material C, we list the difference density subspaces used in the new collision attack on 5-round SHA3-384.