Internal Differential Boomerangs: Practical Analysis of the Round-Reduced Keccak-f Permutation

Jérémy Jean and Ivica Nikolić

Nanyang Technological University, Singapore
{JJean,inikolic}@ntu.edu.sg

Abstract. We introduce internal differential boomerang distinguisher as a combination of internal differentials and classical boomerang distinguishers. The new boomerangs can be successful against cryptographic primitives having high-probability round-reduced internal differential characteristics. The internal differential technique, which follow the evolution of differences between *parts of* the state, is particularly meaningful for highly symmetric functions like the inner permutation Keccak-f of the hash functions defined in the future SHA-3 standard. We find internal differential and standard characteristics for three to four rounds of Keccak-f, and with the use of the new technique, enhanced with a strong message modification, show practical distinguishers for this permutation. Namely, we need 2^{12} queries to distinguish 7 rounds of the permutation starting from the first round, and approximately 2^{18} queries to distinguish 8 rounds starting from the fourth round. Due to the exceptionally low complexities, all of our results have been completely verified with a computer implementation of the analysis.

Keywords: SHA-3, Keccak, internal differential, boomerang, practical-complexity distinguisher.

1 Introduction

The family of sponge functions Keccak [4] was one of the proposals for the hash function competition organized by NIST [30]. In 2012, Keccak was announced as the winner of this competition, and some hash functions from this family will officially become part of the SHA-3 standard [31], to complement the SHA-2 hash standard. As such, Keccak is among the most significant cryptographic primitives to date; its security is therefore of crucial importance.

In the past several years, Keccak has received significant amount of attention from the cryptographic community, both during the competition and after being announced as the winning algorithm. Analyses of round-reduced versions have been proposed for the hash function, for the underlying permutation, and for various secret-key schemes based on this permutation. So far, the best attacks on the hash function in the standard model reach five rounds [15, 16], while in the keyed model reach up to nine rounds [17]. For the underlying permutation, the best analysis in terms of complexity reaches six rounds and requires 2^{11} queries [23], while in terms of number of rounds, the best is on eight rounds and requires 2^{491} queries [18].

In this paper, we present distinguishers for round-reduced versions of the permutation Keccak-f used in Keccak based on a new analysis technique called *internal differential boomerang distinguishers*. We stress that we propose distinguishers on the round-reduced permutation: the paper does not target a keyed mode using it, while the technique may encourage follow-up works. From a high-level perspective, this technique resembles classical boomerangs, but in one part of the boomerang it uses internal differentials, which consider differences between *part of* a state, rather than a difference *between* two states. As a result, our boomerang produces *pairs* of state values that have specific input internal and output differences, while classical boomerangs produce *quartets* of inputs.

More precisely, on the one hand, the classical boomerang starts with an input pair that has a specific internal difference, and the corresponding outputs are computed. Then, a second output pair is produced by XORing a specific difference to both output values, and finally, these values are inverted to a second input pair, and it is checked if this pair has the same specific input difference. On the other hand, the internal differential boomerang distinguisher framework depicted in this paper is slightly different than this classical boomerang scenario since it considers *internal* differences, which ultimately produces pairs of inputs rather than quartets. Specifically, an input with particular *internal* difference generates an output to which we apply a specific output difference. The second output is then inverted to a second input, and one checks whether it has the given input *internal* difference.

For both these kinds of boomerangs, the time complexity required to generate either a right quartet or a right pair depends on the probability of the differentials (internal differentials or regular differentials) used in the two parts of the primitive. Furthermore, in internal differential boomerangs, the part of the primitive covered by the internal differential is passed twice, whereas the part covered by the standard differential only once (in classical boomerang, both of the parts are passed twice). Thus, our technique outperforms the classical boomerangs when high-probability internal differentials exist for several rounds of the primitive. We further give an evaluation of the time complexity required to generate right quartets and pairs for both types of boomerangs, and discuss the use of the message modification technique to greatly reduce this complexity when we have the ability to choose bits of intermediate state values.

Interestingly, Dinur et al. [15] collision attacks on `Keccak` can be seen as an instance of our boomerangs: as they perform only forward queries, their attacks are in fact amplified version of our boomerangs. Thus, the boomerangs presented here can be seen as a generalization of the work of Dinur et al.

We distinguish the round-reduced `Keccak-`f permutation by producing boomerang pairs. First, we find internal differential and standard differential characteristics that are used in the boomerangs. The characteristics span on three to four rounds and, as in some rounds the differences are truncated, have very high probabilities. We combine the characteristics according to the internal differential boomerang, and with the use of an enhanced message modification (which allows to pass deterministically the two low probability rounds in the middle of the boomerang), obtain boomerang pairs with low and practical complexity. We also provide a rigorous bound on the query complexity of producing such boomerang pairs in the case of a random permutation. As this complexity is much higher than what we need for round-reduced `Keccak-`f, we claim distinguishers.

Our internal characteristics depend on the round constants, thus we give distinguishers on the round-reduced `Keccak-`f permutation for two different cases: when the permutation starts[1] at round 0, and when it starts at round 3. In the first case,

[1]Note that while the draft FIPS 202 [31] defines the r-round-reduced versions of `Keccak-`f as the last r rounds of `Keccak-`f, this paper allows the reduced permutation to start at any round number.

we can distinguish the permutation reduced to 6 rounds with 2^5 queries, and 7 rounds with 2^{13} queries. In the second case, we can distinguish 7 rounds with $2^{10.3}$ queries, and 8 rounds with $2^{18.3}$ queries.

We emphasize that the whole analysis, due to its exceptionally low complexity, has been implemented and successfully verified. We refer the reader to the Appendix C for the outputs produced by our computer experiments. We also stress that our results do not threaten the security of the full-round `Keccak`-f permutation. A summary of previous analysis of `Keccak`, along with our new results, are given in Table 1 and Table 2.

Table 1: Summary of attacks on `Keccak`.

Rounds	Complexity	Type	Technique	Reference
2	2^{33}	Collision	Differential	[32]
2	2^{33}	Preimage	Differential	[32]
3	2^{25}	Near-Collision	Differential	[32]
4	2^{221}	Preimage	Rotational	[27]
4	2^{25}	Distinguisher	Differential	[32]
4	practical	Collision	Differential	[16]
4	practical	Collision	Differential	[22]
4	2^{506}	Preimage	Rotational	[27]
5	2^{115}	Collision	Int. differential	[15]
5	2^{35}	Key recovery (MAC)	Cube attack	[17]
5	practical	Near-Collision	Differential	[16]
6	2^{52}	Distinguisher	Differential	[13]
6	2^{36}	Key recovery (Stream)	Cube attack	[17]
8	2^{129}	MAC forgery	Cube attack	[17]
9	2^{256}	Keystream prediction	Cube attack	[17]

Table 2: Distinguishers of reduced-round versions of `Keccak`-f.

Rounds	Complexity	Type	Technique	Reference
5	2^8	Distinguisher	Rebound	[18]
6	$\mathbf{2^5}$	**Distinguisher**	**Internal Diff. Boomerang**	**Section 4**
6	2^{10}	Distinguisher	Zero-sum	[1, 10]
6	2^{11}	Distinguisher	Self-symmetry	[23]
6	2^{32}	Distinguisher	Rebound	[18]
6.5	unknown	Distinguisher	Cube tester	[17]
7	$\mathbf{2^{10}}$	**Distinguisher †**	**Internal Diff. Boomerang**	**Section 4**
7	$\mathbf{2^{13}}$	**Distinguisher**	**Internal Diff. Boomerang**	**Section 4**
7	2^{15}	Distinguisher	Zero-sum	[1, 10]
7	2^{142}	Distinguisher	Rebound	[18]
8	$\mathbf{2^{18}}$	**Distinguisher †**	**Internal Diff. Boomerang**	**Section 4**
8	2^{18}	Distinguisher	Zero-sum	[1, 10]
8	2^{491}	Distinguisher	Rebound	[18]
24	2^{1590}	Distinguisher	Zero-sum	[11]

†: Start from round 3.

Application of the internal differential boomerangs. The impact of this kind of boomerangs depends on the analyzed framework. When the subject of analysis is a block cipher, then the impact of the internal differential boomerangs is similar to that of the classical boomerangs, i.e. they immediately lead to distinguishers and possibly can be extended to key recovery attacks. On the other hand, in the framework of hash/compression functions and permutations, their significance depends on the quality of the internal differential and standard differential characteristics used to produce the boomerang pairs. For instance, if the input internal difference complies to the conditions of the input to the hash/compression function and the output difference has a low hamming weight, then an internal differential boomerang pair may lead to near collisions.

The internal differential boomerangs presented further in this paper only apply to the round-reduced Keccak-f permutation, but not to Keccak. This is due to the message modification used in the middle states, which results in inputs that do not comply to the inputs conditions to the sponge construction of Keccak where the values in the capacity part cannot be controlled. Similarly, it prevents applying the distinguishers to other keyed constructions, such as Keyak [6] and Ketje [5]. Therefore, our internal differential boomerangs only allow to distinguish round-reduced Keccak-f from a random permutation. However, their impact relate to Keccak since it adopts the hermetic sponge strategy as a design philosophy [3]. In its original formulation, this consists of using the sponge construction (providing security against generic attacks) and calling a permutation that should not have any properties (called structural distinguishers) besides having a compact representation. Our results disprove this requirement for the round-reduced Keccak-f permutation by showing a non-random behavior.

2 Description of Keccak-f

In this section, we give a partial description of the hash functions that will be defined in the future SHA-3 standard [31]. In particular, since the results in this paper only deal with the inner permutation (further denoted by Keccak-f), we do not recall the details of the sponge construction. For a complete description of this family of functions, we refer the interested reader to [4, 31].

The Keccak-f permutation works on a state of $b = 25 \times 2^l$ bits, where $b \in \{25, 50, 100, 200, 400, 800, 1600\}$, and has $n_r = 12 + 2l$ rounds. We count the rounds starting from zero. The results in this paper consider round-reduced versions of Keccak-f[1600], where the full permutation has $n_r = 24$ rounds. As introduced in [31], we define by Keccak-p a round-reduced version of the Keccak-f permutation, where its $n \geq n_r$ rounds are the n last ones of Keccak-f. In this paper, we leverage the restriction on the starting round number and further introduce the notation Keccak-$p_{i,n}$ to consider the n consecutive rounds of Keccak-f[1600] starting at round i; that is, rounds $i, \ldots, i + n - 1$. Using this notation, Keccak-f[1600] would be Keccak-$p_{0,24}$.

Each round of Keccak-$f[b]$ is composed of five steps: the first three (θ, π and ρ, in this order) are linear and further denoted together by $\lambda = \pi \circ \rho \circ \theta$, the fourth step is non-linear and denoted by χ, and the last step ι adds round-dependent constants $RC[i]$, $0 \leq i < n_r$, to break symmetries. Each step applies to different parts of the state, which is seen as a three-dimensional array of bits of dimension $5 \times 5 \times b$. A bit $S[x, y, z]$ in a state S is addressed by its coordinates (x, y, z), $0 \le (x, y) < 5$ and $0 \le z < b$. Furthermore, for fixed x, y and z, $S[x, y, \bullet]$ refers to a *lane* of b bits, and $S[\bullet, \bullet, z]$ to a *slice* of 25 bits.

We now discuss the details of each of the five steps on a given input state S:

The θ step operates on the slices of the state by performing the following operation at each coordinate (x, y, z):

$$S[x, y, z] \leftarrow S[x, y, z] \oplus \Big(\bigoplus_{y'=0}^{4} a[x-1, y', z] \Big) \oplus \Big(\bigoplus_{y'=0}^{4} a[x+1, y', z-1] \Big).$$

This linear step brings diffusion to the state. For instance, it expands a single bit difference to 11 bits, while the inverse step θ^{-1} expands it to about $b/2$ bits.

The ρ step rotates the bits inside each lane. The rotation constants are independent of the round numbers, and they are different for each of the 25 lanes (refer to [4] for the actual values).

The π step operates on each slice independently by permuting the 25 bits. Namely, at each coordinate (x, y, z), it applies:

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

This step mixes the lanes and thus brings an additional diffusion to the state.

The χ step is the only non-linear operation in a round and it applies the same 5-bit S-Box to each 5-bit row $S[\bullet, y, z]$ of the internal state. In total, $b/5$ independent S-Boxes are applied, that is 320 in the case of `Keccak-`f[1600]. The S-Box has maximal differential probability of $p_{max} = 2^{-2}$.

The ι step XORs the b-bit round-dependent constant $RC[i]$ at round i to the lane $S[0, 0, \bullet]$, $0 \le i < n_r$. The 24 constants used in `Keccak-`f[1600] are given in Appendix A.

3 The Internal Differential Boomerang Distinguisher

In this section, we introduce a new distinguisher called the *internal differential boomerang distinguisher*. As it combines internal differentials and the boomerang attack, we first give a brief overview of these two strategies, and then present the new technique.

3.1 The Internal Differential Attack

In the internal differential attack, introduced by Peyrin [34] for analysis of hash functions[2], the adversary observes the propagation of *the difference between the two halves of the same state* through the rounds of the cryptographic function/permutation. Similar to the case of classical differential analysis, the goal of the adversary is to

[2]Later, the technique has been applied to the case of block ciphers [19].

show that the propagation of some particular internal difference happens with an unusually high probability.

Let F be a permutation, and the n-bit state S is split into two halves S^H and S^L. With this notation, it follows that $|S^H| = |S^L|$ and $S = S^H \| S^L$. The internal difference $\delta(S)$ of the state S is computed as the XOR of its two halves, i.e. $\delta(S) = S^H \oplus S^L$. Then, an internal differential for F is a pair of internal differences $(\varDelta, \nabla)$, and its probability is defined as:

$$\Pr_S \Big(\delta(F(S)) = \nabla \Big| \, \delta(S) = \varDelta \Big).$$

In other words, this is the probability that a randomly chosen input state S with an internal difference $\varDelta$, after the application of F, will result in an output state with internal difference ∇. Similarly to the standard differential attacks, we can define an internal differential characteristic as the propagation of the internal differences through the rounds of the permutation. Obviously, to each such internal differential characteristic, we can associate a probability that this propagation holds as expected.

3.2 The Boomerang Attack

In classical boomerang attacks [36][3], the permutation F is seen as a composition of two permutations $F = g \circ f$, where each of them covers some rounds at the beginning and at the end of F. Even though a high-probability differential might not exist for F, if high-probability differentials do exist for the two permutations f and g, then one can attack F with the boomerang technique.

Let $\varDelta \rightarrow \varDelta^*$ be a differential for f that holds with a probability p and $\nabla \rightarrow \nabla^*$ be a differential for g that holds with a probability q. According to Figure 1, the adversary starts with a pair of inputs $(P_1, P_2) = (P_1, P_1 \oplus \varDelta)$ and, by applying F, produces a pair of corresponding outputs $(C_1, C_2) = (F(P_1), F(P_2))$. Then, the adversary produces a new pair of outputs $(C_3, C_4) = (C_1 \oplus \nabla^*, C_2 \oplus \nabla^*)$. For this pair, the adversary obtains the corresponding pair of inputs $(P_3, P_4) = (F^{-1}(C_3), F^{-1}(C_4))$. The main observation of the boomerang technique is that the difference $P_3 \oplus P_4$ would be $\varDelta$ with a probability of at least p^2q^2 because:

1. The difference $f(P_1) \oplus f(P_2)$ is $\varDelta^*$ with probability p.
2. The two differences $g^{-1}(C_1) \oplus g^{-1}(C_3)$ and $g^{-1}(C_2) \oplus g^{-1}(C_4)$ are both ∇ with probability q^2.
3. When 1. and 2. hold, then the difference $g^{-1}(C_3) \oplus g^{-1}(C_4)$ is $\varDelta^*$ (with probability pq^2), and therefore $f^{-1}(C_3) \oplus f^{-1}(C_4)$ is $\varDelta$ with probability p^2q^2.

The quartet of states (P_1, P_2, P_3, P_4) fulfilling the conditions $P_1 \oplus P_2 = P_3 \oplus P_4 = \varDelta$ and $F(P_1) \oplus F(P_3) = F(P_2) \oplus F(P_4)$ is called a *boomerang quartet*. As shown above, the quartet can be found in time equivalent to $(pq)^{-2}$ queries to the permutations. On the other hand, finding the boomerang quartet in the case of a random permutation requires about 2^n queries. Consequently, the boomerang approach yields a distinguisher for F as soon as the adversary can find the two differentials for f and g such that $(pq)^{-2} < 2^{-n}$, that is $pq > 2^{-n/2}$.

It has been shown in [8,9] that when F is a public permutation, a block cipher in the chosen-key attack framework, or a compression function, then the complexity

[3]The boomerang attack is closely related to higher-order differential techniques [21,24].

of producing the boomerang quartet can be reduced with the use of the message modification technique. That is, the adversary can choose particular state words to ensure that some probabilistic differential transitions hold with probability one. Consequently, some rounds can be passed deterministically, so that their probabilities do not contribute towards the total probability $(pq)^2$. The number of such free rounds depends on how efficiently the message modification can be applied. In general, the modification is used in the rounds around the boomerang switch, i.e. the last few rounds of f and the first few rounds of g.

3.3 The Internal Differential Boomerangs

In this section, we show that the internal differential attack can be used in the boomerang setting: we call this combined analysis *the internal differential boomerangs*. Although this new type of analysis shares similarity with the classical boomerangs based on standard differentials, we emphasize that there are a few differences between them. The first difference is in the number of differentials required to achieve the boomerang: the classical boomerang uses four differentials, whereas the internal differential boomerang works with only three. The second difference is in the type of differentials: the classical boomerang can use (almost) any two differentials for f and g, while for the internal differential boomerang, one of the differentials must have a special type.

Let F be a permutation that (similarly to the classical boomerang) is seen as a composition $F = g \circ f$. Let (Δ, Δ^*) be an *internal differential* for f that holds with probability p, and (∇, ∇^*) be a *standard differential* for g that holds with probability q, *where the input difference* ∇ *has an internal difference of zero*, i.e. $\delta(\nabla) = 0$. Then, the internal differential boomerangs can be described as:

1. Fix a random input P_1 with an internal difference Δ, i.e. $\delta(P_1) = \Delta$.
2. Produce the corresponding output $C_1 = F(P_1)$.
3. Produce another output C_2 such that $C_2 = C_1 \oplus \nabla^*$.
4. Produce the corresponding input $P_2 = F^{-1}(C_2)$.
5. Check if $\delta(P_2) = \Delta$. If it holds, output (P_1, P_2), otherwise go to 1.

The probability that the condition at step 5 holds is at least p^2q. This is based on a reasoning illustrated in Figure 1. Let $\nabla = \nabla^H \| \nabla^H$ and $\nabla^* = \nabla^{H*} \| \nabla^{L*}$ be the input and the output differences of the standard differential used in the function g. For a random input $P_1 = P_1^H \| (P_1^H \oplus \Delta)$, the output $X = f(P_1)$ will be $X^H \| (X^H \oplus \Delta^*)$ with probability p. Furthermore, for a pair of outputs (C_1, C_2) such that $C_1 \oplus C_2 = \nabla^* = \nabla^{H*} \| \nabla^{L*}$, after the inversion of g, the output pair (X, Y) will satisfy $X \oplus Y = \nabla = \nabla^H \| \nabla^H$ with probability q. Then,

$$Y = X \oplus \nabla = \left[X^H \| (X^H \oplus \Delta^*) \right] \oplus \left[\nabla^H \| \nabla^H \right] = Y^H \| (Y^H \oplus \Delta^*),$$

where $Y^H = X^H \oplus \nabla^H$. Therefore, the internal difference in Y is Δ^*, and after the inversion of f, it will become Δ with probability p. As a result, this algorithm outputs a pair of inputs with probability p^2q. We call such a pair an internal differential boomerang pair.

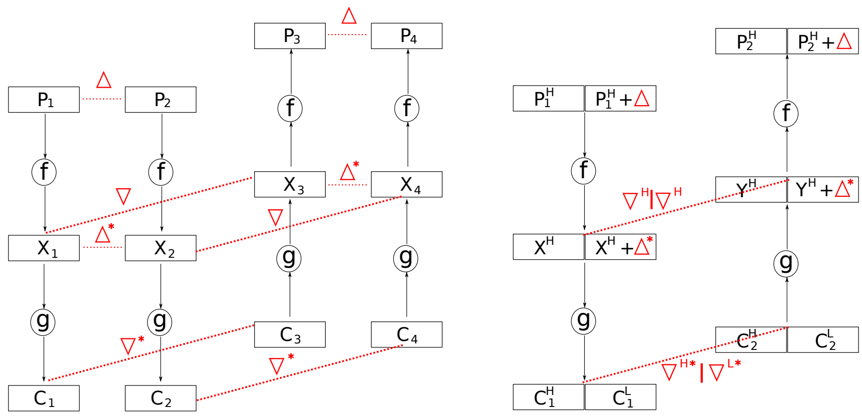

Figure 1: The classical boomerangs on the left, and the internal differential boomerangs on the right.

For a random n-bit permutation F, the pair can be found in around $2^{n/2}$ queries[4] to F. Therefore, the internal differential boomerang yields a distinguisher if $p^2q > 2^{-n/2}$. Recall that the same condition for the classical boomerangs is $pq > 2^{-n/2}$. Consequently, it is beneficial to use the internal differential boomerang technique over the classical boomerang strategy only if the internal differential for f has a much higher probability than a differential for f.

Given a public permutation (or a compression function) $F = g \circ f$, we can start the internal differential boomerang in any round of f (but not in g), and from there produce the pair of inputs and the pair of outputs. It is usually beneficial to start at the end of f and, with the use of the message modification technique, to pass a few rounds around the boomerang switch for free (deterministically). Then, the formula for the probability of the boomerang becomes $p_*^2 q_*$, where p_* and q_* are the differential probabilities of the non-linear parts of f and g respectively, that are passed probabilistically.

Dinur et al. collision attack. In [15], Dinur et al. present a collision attack on reduced variants of `Keccak` hash function by selecting message blocks in a small subspace[5] such that a high-probability characteristic might map them to a small subspace after a certain number of rounds of `Keccak`-f. More precisely, they find round-reduced internal characteristics and then they extend them for an additional 1.5 round. They call this extension *bounding the size of the output subset* and note that this is possible because the differences are quite sparse and the χ step has a slow diffusion.

We note that Dinur et al. collision attack is in fact based on the internal differential boomerangs presented in this paper. Their internal differential characteristics corresponds to the internal differential part of the boomerang, whereas the aforementioned extension is the standard differential part of the boomerang. Furthermore, Dinur et al. start the attack from the two inputs with specific internal differences and then check if the difference of the two outputs is as expected. This is precisely the variant

[4]In a random permutation, the boomerang will return P_2 with internal difference $\varDelta$ with a probability $2^{-n/2}$.

[5]A related subspace problem has been discussed in [25].

of the boomerang attack called amplified boomerang [20], where the attacker only makes forward queries. Thus, Dinur et al.'s collision attack succeeds as after the amplification in the middle, the remaining 1.5 rounds are passed according to any standard differential that at the output has no active bits among those that comprise the hash value.

Truncated differences. We further analyze the case when the input internal difference Δ and the output standard difference ∇^* of the boomerang are not fully determined, but are truncated. Namely, only some bits of these differences are determined, whereas the remaining bits can have any value. The lemma given below defines a lower bound on the complexity of finding such boomerang pair in the case of a random permutation. Note, in the lemma, we assume the output difference to be XOR difference, that is, the output difference is produced as an XOR of the two outputs.

Lemma 1. *For a random n-bit permutation π, the query complexity Q of producing an internal differential boomerang pair, with truncated input internal difference Δ determined in n_I bits and truncated XOR output difference ∇^* determined in n_O bits, satisfies:*

$$Q \geq \min(2^{n_I-2}, 2^{\frac{n_O}{2}-\frac{3}{2}}).$$

Proof. As the output difference is truncated XOR difference, ∇^* can be seen as a subset of $\{0,1\}^n$ (has zeros in n_O particular bits, while the remaining bits $n - n_O$ bits take all possible values). We may partition the set $\{0,1\}^n$ into output sets $O_1, \ldots, O_{n_O}$ such that $|O_i| = 2^{n-n_O}$ for $i = 1, \ldots, n_O$. Furthermore, we may partition the set $\{0,1\}^n$ into two input subsets I_G, I_B (good input, bad input), where I_G is composed of all x with internal difference $\delta(x) = \Delta$ and $I_B = \{0,1\}^n \setminus I_G$. Obviously, $|I_G| = 2^{n-n_I}$. Then, (x, y) forms a boomerang pair, iff $x, y \in I_G$ and $\pi(x), \pi(y) \in O_i$ for some i.

Let us define a game G_0: an adversary $\mathcal{A}$ has an access to a random permutation oracle $\pi : \{0,1\} \to \{0,1\}^n$ and its inverse π^{-1}, making a total of q queries to these two oracles. Starting from G_0, we will build a chain of games which are similar until *bad* is set (for details on this methodology, see [2]; our proof follows the proof given in [33] for the case of standard difference). In the games $G_k (k = 0, 1, 2)$, let E_k be the following event: $\mathcal{A}$ finds $x \neq y$ where $x, y \in I_G, \pi(x), \pi(y) \in O_i$, for some i while interacting with the game G_k. Further, we will show that

$$\Pr(E_0) \leq \frac{q^2}{2^n} + \frac{q^2}{2^{n_O}} + \frac{q}{2^{n_I}} \tag{1}$$

Before we give a formal proof, we remark that the intuition for this bound is as follows. The first term $\frac{q^2}{2^n}$ is the upper bound on the collision probability error due to the fact that we simplify the problem by replacing the random permutation π with a random function. The second term is the probability that two random outputs will collide on n_O bits. The third term is the probability that a random output, after the inversion will result in an input from I_G.

Let us define a game G_1 that is similar to G_0 except that the permutation π is replaced by a relation $P \subset \{0,1\}^n \times \{0,1\}^n$ that is injective and functional, but not necessary defined in the whole domain. According to the naming convention of [2], the relation P is called a partial permutation, whereas injectivity and functional conditions together are called permutation constraints. Initially, P is empty and through execution of G_1, its values are being sampled randomly with respect to the permutation constraints. Whenever $P(x)$ (respectively $P^{-1}(y)$) is needed, first it is checked if P (respectively P^{-1}) is defined on x (respectively y). If this is the case, then appropriate value is returned, otherwise $P(x)$ (respectively $P^{-1}(y)$) is sampled uniformly at random from $\overline{img(P)}$ (respectively $\overline{img(P^{-1})}$), where $\overline{img(P)}$ is complement of the image of P. Since the sampling is the same as in the game G_0, it follows that

$$\Pr(E_1) = \Pr(E_0). \tag{2}$$

Next, we define a game G_2, which is the same as G_1, except that the permutation constraint for P does not have to be fulfilled. That is, the values $P(x)$ (respectively $P^{-1}(y)$) are sampled at random from $\{0,1\}^n$, but the game stops immediately when the permutation constraint is not satisfied. Unless the permutation constraint is violated by the occurrence of a collision between a new output value return by P and a previous output value of P, or a collision between a new input P^{-1} and a previous input, the games G_1 and G_2 proceed identically. Since at each query there are at most q previous P (respectively P^{-1}) output values already defined, it follows that

$$|\Pr(E_2) - \Pr(E_1)| \leq \frac{q^2}{2^n}. \tag{3}$$

At this stage, we stop building chain of games and upper bound the probability $\Pr(E_2)$ directly. Among the q queries, the two queries required for the occurrence of the event E_2 can be either: 1) two queries to P, or 2) one of them is a query to P^{-1}. In the first case, the two queries to P on which the event E_2 occurred, are among the total q queries (either to P or to P^{-1}). Assume that all q queries were to P and for all of them belong to I_G (this only increases the probability of the event E_2, and we are looking for an upper bound of $\Pr(E_2)$). The answers $P(x)$ are sampled uniformly at random from $\{0,1\}^n$, thus two queries $P(x), P(y)$ will collide on some O_i (where $i = 1, \ldots, 2^{n_O}$) with a probability $\leq \frac{q^2}{2^{n_O}}$. In the second case, one of the queries on which E_2 occurred, is to P^{-1}. As the answers P^{-1} are sampled at random, the probability that a query P^{-1} belongs to I_G is $\frac{|I_G|}{|2^n|} = 2^{-n_I}$. Hence, even if all q queries were to P^{-1}, the probability that one of them is in I_G can be upper bounded by $\frac{q}{2^{n_I}}$. As a result, we get

$$\Pr(E_2) \leq \frac{q^2}{2^{n_O}} + \frac{q}{2^{n_I}}. \tag{4}$$

Therefore, from (2), (3) and (4), it follows that:

$$\Pr(E_0) = \Pr(E_1) \leq \frac{q^2}{2^n} + \Pr(E_2) \leq \frac{q^2}{2^n} + \frac{q^2}{2^{n_O}} + \frac{q}{2^{n_I}} \leq 2\frac{q^2}{2^{n_O}} + \frac{q}{2^{n_I}}, \tag{5}$$

where the last inequality comes from $n \geq n_O$.

Let us show that (5) implies the claimed complexity bound. As usual, we want $\Pr(E_0) = \frac{1}{2}$, hence, if each of the two terms at the right hand side of (5) has a value of $\frac{1}{4}$, then the complexity bound will follow. For the first term, this happens when $q = 2^{\frac{n_O}{2} - \frac{3}{2}}$, while for the second when $q = 2^{n_I - 2}$. This concludes the proof. $\square$

4 Distinguishers for the Round-Reduced `Keccak-`f Permutation

In this section, we present internal differential boomerang distinguishers on the round-reduced permutation `Keccak-`$f[1600]$, further denoted `Keccak-`$p_{i,n}$, where the starting round i and the number of rounds n is specified in the text for each case. In comparison to [31] where all the reduced variants simply called `Keccak-`p start at the first round, we relax this constraint by allowing the permutation to start at any number of round.

To describe our results, we first define the two differentials used in the boomerang: the internal differential used in the first rounds, and the standard differential used in the last rounds. Next, we show that a message modification can help to deterministically pass the two rounds that surround the boomerang switch. Finally, we present the actual distinguishers.

4.1 Internal Differential Characteristics

The 1600-bit state S of `Keccak` is composed of 25 lanes of 64 bits. The internal difference $\delta(S)$ of the state is defined as the XOR difference between the higher 32 bits and the lower 32 bits, for each lane – we stress out that the internal difference is defined precisely the same as in the work of Dinur et al. [15]. Hence, the internal difference is composed of 25 words of 32 bits, and can be seen as an 800-bit vector.

Let us scrutinize the behavior of the five round steps in regard to internal differences. The linear step θ may introduce an increase in the hamming weight of the internal difference, by a factor up to 11. The two steps ρ and π only permute the bits in the internal differences, but maintain their hamming weight. The non-linear step χ may increase the hamming weight of the internal difference. For instance, one-bit difference at the input (resp. output) of the S-box, may become a difference in more than 1 bit at the output (resp. input) of the S-box. However, a fixed 1-bit input difference can affect only up to three bits in the output difference, while a fixed 1-bit difference at the output of χ can affect up to 5 bits in the input difference. The ι step that XORs round constants can increase the hamming weight of the internal difference by *at most* the hamming weight of the rounds constant $\delta(RC[i])$, which are very sparse (see Appendix A for the actual values). Indeed, as already noted in [14, 16], the round constants used in `Keccak-`f play a crucial role in the existence of high-probability internal differential characteristics in the inner permutation.

Due to the good diffusion of the round function of `Keccak-`$f[1600]$, a state with low-weight internal difference can be transformed into a state with a high weight in a matter of a few rounds. To increase the number of rounds covered by the internal differential characteristic, while maintaining a high and practical probability, we use two approaches. First, we start in the middle of the characteristic with zero internal difference and pass one round with probability one. Second, we consider truncated characteristics (or differentials), i.e. the differences are not necessarily fully specified in all bits.

By the first approach, which is often used for constructing standard differential characteristics, the characteristics are built from inside out. First, a low-weight difference in some middle round of the characteristic is fixed, and then, by propagating the difference backwards and forwards, the input and the output differences of the characteristic are obtained. Therefore, the middle rounds of the characteristic have a high probability, while the rounds close to the input and to the output are of low probability. However, the low-probability rounds can be passed for free if we use a message modification or if we consider truncated characteristics, which is in fact the second approach.

The internal characteristic $\mathcal{I}_3$. Let us focus on the following 3-round internal differential characteristic $\mathcal{I}_3$, that starts at round 0, and that has been built with the first approach:

$$\begin{bmatrix}429\\800\end{bmatrix} \xleftarrow{\lambda^{-1}} \begin{bmatrix}\mathbf{1}\\800\end{bmatrix} \xleftarrow{\chi^{-1}} \begin{bmatrix}\mathbf{1}\\800\end{bmatrix} \xleftarrow{\iota_0^{-1}} \begin{bmatrix}0\\800\end{bmatrix} \xrightarrow{\lambda,\chi} \begin{bmatrix}0\\800\end{bmatrix} \xrightarrow{\iota_1} \begin{bmatrix}3\\800\end{bmatrix} \xrightarrow{\lambda} \begin{bmatrix}33\\800\end{bmatrix} \xrightarrow{\chi,\iota_2} \begin{bmatrix}?\\800\end{bmatrix}.$$

Round 0 Round 1 Round 2

The states are represented by the column vectors, where the upper number denotes the hamming weight of the internal difference, and the lower number gives the amount of bits in which the internal difference is fully determined. The numbers in **bold** around the χ step of round 1 represent active S-Boxes for that step, which is passed with a probability smaller than one. By ?, we represent an undetermined value.

The characteristic has been built by fixing a zero internal difference at the input of round 1. In the forward direction, there are no active S-Boxes in round 1, and the output difference is defined in all 800 bits after the linear step λ of round 2. The following steps χ and ι_2 produce some differences, but as we show later in Section 4.3, the value of this internal difference is irrelevant. In the backward direction, $RC[0]$ of ι_0 introduces only one bit difference, and thus the subsequent χ^{-1} has only one active S-Box. After the inversion of the linear layer, we can fully compute the internal difference at the input of the characteristic, so that each of the 800 bits are fully determined. Therefore, the whole 3-round internal characteristic has 34 active S-Boxes (probability 2^{-68}), and in the first two rounds has only a single active S-Box (probability 2^{-2}). The characteristic is fully specified in Appendix B.

The internal differential $\mathcal{ID}_4$. We can construct a longer characteristic by going backwards one additional round. However, in this round the hamming weight of the internal difference at the input of χ^{-1} would be high (in the above $\mathcal{I}_3$, the weight is 429). To avoid significant reduction of probability, we switch to truncated internal differences. That is, instead of trying to define completely the output difference of this χ^{-1} (that would be obtained with an extremely low probability), we specify the difference only in n_I bits out of 800 bits. The internal difference in each of these n_I specific bits can be either 0 or 1, but the probability of this event must be one. As a result, the probability of the first round of the characteristic would be one.

Once the truncated difference is fixed in n_I bits at the output of χ^{-1}, the remaining three *linear* steps of the round will keep the truncated property: π^{-1} and ρ^{-1} will only permute and rotate the truncated difference and thus at the output of these two steps still it will be defined in n_I bits, while at the output of θ^{-1} the internal difference will belong to a subspace of dimension $800 - n_I$. We note that with a minor modification of Lemma 1, the obtained input internal difference can be used to compare the query complexity to the generic case[6] Therefore, to simplify the presentation of the input

[6]That is, we use the subspace to claim distinguisher for the permutation. This is in line with our initial intention to show that the round-reduced permutation exhibits non-random properties.

internal difference, in the further analysis, we omit the three linear steps of the first round.

The number of bits n_I in which the truncated difference at the output of χ^{-1} is defined with probability one depends on the round constants RC_i. For instance, if we start with round 0, then there is no bits in which the truncated difference is determined, i.e. $n_I = 0$. Only if we start with round 3, the number n_I will be sufficiently large to claim later (according to Lemma 1) that the complexity of producing boomerang pairs for $\texttt{Keccak-}p_{3,n}$ is lower than the generic complexity, with $n \in \{7, 8\}$.

The resulting 4-round internal differential characteristic $\mathcal{I}_4$, that starts at round 3, is defined as:

$$\underbrace{\begin{bmatrix} \texttt{?} \\ 64 \end{bmatrix} \xleftarrow{\chi^{-1}} \begin{bmatrix} \texttt{398} \\ 800 \end{bmatrix} \xleftarrow{\iota_3^{-1}}}_{\text{Round 3}} \underbrace{\begin{bmatrix} \texttt{397} \\ 800 \end{bmatrix} \xleftarrow{\lambda^{-1}} \begin{bmatrix} \mathbf{5} \\ 800 \end{bmatrix} \xleftarrow{\chi^{-1}} \begin{bmatrix} \mathbf{5} \\ 800 \end{bmatrix} \xleftarrow{\iota_4^{-1}}}_{\text{Round 4}} \underbrace{\begin{bmatrix} 0 \\ 800 \end{bmatrix} \xrightarrow{\lambda,\chi} \begin{bmatrix} 0 \\ 800 \end{bmatrix} \xrightarrow{\iota_5}}_{\text{Round 5}} \underbrace{\begin{bmatrix} 2 \\ 800 \end{bmatrix} \xrightarrow{\lambda} \begin{bmatrix} 22 \\ 800 \end{bmatrix} \xrightarrow{\chi,\iota_2} \begin{bmatrix} \texttt{?} \\ \texttt{800} \end{bmatrix}}_{\text{Round 6}}.$$

The characteristic has been built by fixing a zero internal difference at the input of round 5. The forward propagation is similar to $\mathcal{I}_3$. Backwards, after the addition of the constant $RC[4]$, the weight of the internal difference is five. Hence, χ of round 4 has at most five active S-Boxes, that can be passed probabilistically and would result in a state with internal difference of weight five. Then, the linear steps λ^{-1} in round 4 and the addition of $RC[3]$ in round 3 increase the weight of the internal difference to 398. *In the following χ^{-1}, we switch to truncated differences.* Although the input difference has a weight of 398 (possibly, all 320 S-Boxes are active), at the output of χ^{-1}, the internal difference is 0 in 55 specific bits, and 1 in 9 other bits. In other words, $n_I = 55 + 9 = 64$ bits of the internal difference are defined deterministically and thus, the probability to pass this χ^{-1} is one. Note, the truncated characteristic in the first round holds with probability one only when moving backwards through the round.

The probability of the truncated internal differential characteristic $\mathcal{I}_4$ can be evaluated as follows: in round 3 the probability is 1, in round 4 there are 5 active S-Boxes, thus the probability is 2^{-10}, in round 6 there are no active S-Boxes, while in round 7 there are 22 active S-Boxes (probability is 2^{-44}). Hence, when going backwards through the rounds, the probability of the whole 4-round characteristic is 2^{-54}. Furthermore, the probability of the first three rounds is 2^{-10}.

Recall that the boomerangs can use differentials instead of characteristics. As the probability of a differential may be higher than the probability of a single characteristic, the complexity of producing boomerang pairs may be reduced. Therefore, let us build a 4-round differential $\mathcal{ID}_4$ by using the same approach as for $\mathcal{I}_4$. That is, for all of the characteristics that belong to $\mathcal{ID}_4$, we start at round 5 with zero internal difference. In the forward direction, we move deterministically through round 5 and at the input of χ in round 6, we have 22 active S-Boxes (i.e. all the characteristics are equally defined in this part of the differential). In the backward direction, all the characteristics are the same up to the input of χ^{-1} of round 4, but the five active S-Boxes in each of the characteristics results in different outputs. Then, for each of the outputs, we move through λ^{-1} of round 4, ι_3, χ^{-1} of round 3, and at the output of χ^{-1}, we check if the truncated difference is defined in the same 64 bits as $\mathcal{I}_4$. Therefore, all the characteristics of the differential $\mathcal{ID}_4$ have the same input truncated difference, and the same difference at the input of χ in round 6 (the output of this χ is irrelevant as before). We found experimentally the probability of $\mathcal{ID}_4$ for the first three rounds to be $2^{-4.6}$. This has to be compared to 2^{-10}, which is the probability of the first three rounds of the characteristic $\mathcal{I}_4$. The differential $\mathcal{ID}_4$ is fully specified in Appendix B.

4.2 Standard Differential Characteristics

Along with internal differential characteristics, the boomerang technique described in this paper uses standard differential characteristics. Recall that due to the special requirement of our boomerang, the standard characteristic cannot be of any form since it is connected to the two internal characteristics. This constraints the input difference ∇ of the standard characteristics to be symmetric, i.e. $\nabla = \nabla^H || \nabla^H$, or $\delta(\nabla) = \nabla^H \oplus \nabla^H = 0$, Note, the standard characteristic (unlike the internal characteristic) does not depend on the round number, hence further we omit ι_i from the description of the characteristic.

The standard characteristic that we use relies on the already-known concept of parity kernels, which allows to minimize the number of S-Boxes in two consecutive rounds of `Keccak-`f. This notion has been described in the submission document [4], and has been used in cryptanalytic results [13, 23, 32]. The behavior is possible due to two observations: first, a state-difference may be invariant of the θ step if there is an even number of active bits in each of the 320 column of the internal state; and second, an active S-Box in χ (or in χ^{-1}) leaves unchanged a 1-bit difference with probability 2^{-2}.

The 4-round standard differential characteristic $\mathcal{C}_4$ that we use in the boomerangs is defined as:

$$\underbrace{\begin{bmatrix} ? \\ 1600 \end{bmatrix} \xleftarrow{\chi^{-1}} \begin{bmatrix} ? \\ 1600 \end{bmatrix} \xleftarrow{\chi^{-1}}}_{\text{Round } i} \overbrace{\underbrace{\begin{bmatrix} 2+2 \\ 1600 \end{bmatrix} \xrightarrow{\lambda} \begin{bmatrix} 2+2 \\ 1600 \end{bmatrix} \xrightarrow{\chi}}_{\text{Round } i+1} \begin{bmatrix} 2+2 \\ 1600 \end{bmatrix}}^{\text{Kernel}} \underbrace{\xrightarrow{\lambda} \begin{bmatrix} 22+22 \\ 1600 \end{bmatrix} \xrightarrow{\chi}}_{\text{Round } i+2} \underbrace{\begin{bmatrix} ? \\ 1278 \end{bmatrix} \xrightarrow{\lambda\,\chi} \begin{bmatrix} ? \\ 118 \end{bmatrix}}_{\text{Round } i+3}.$$

The notations used in the characteristic are the same as before. With "$x + x$", we emphasize that the states are comprised of $2x$ active bits, but the actual difference is symmetric, which implies that there are x active bits in each half of the state, with equal differences.

This differential characteristic has been constructed by selecting a symmetric difference of hamming weight four at the input of round $i+1$ (note, this is the smallest possible weight of a symmetric parity kernel). In the backward direction, the step χ^{-1} has only 4 active S-Boxes, and results in a difference that is irrelevant as we further show in Section 4.3. In the forward direction, the selected 4-bit difference acts as a kernel and thus, after the λ step of round $i + 1$, results in a 4-bit difference. The same behavior of the following χ step is expected with probability 2^{-8}, so the input difference to round $i + 2$ still has a weight of four. The linear step in this round expands the difference to 44 active bits. Then, *we switch to truncated differences*. As a result, the difference in the following χ step is defined in 1278 bits, and after all the steps of round $i + 3$, the difference is still deterministically defined in 118 bits (78 zeros and 30 ones).

The differential characteristic $\mathcal{C}_4$ covers four full rounds of the permutation, and holds with probability 2^{-16} in the forward direction since there are a total of 8 active S-Boxes (four in each of the rounds i and $i+1$).

We can define a 3-round differential characteristic $\mathcal{C}_3$, which is basically the same as the first three rounds of $\mathcal{C}_4$, but we start truncating from χ at round $i+1$. That is, in $\mathcal{C}_3$, we begin with 4-bit difference at round $i+1$ and the backward round i is the same as $\mathcal{C}_4$. However, the 4-bit input difference at χ of round $i+1$ results in truncated output difference (with probability 1, instead of 2^{-8}), and after the steps λ and χ of round $i+2$, the truncated difference can still be determined in 1278 bits. Therefore, the probability of $\mathcal{C}_3$ in the forward direction is only 2^{-8} as it has only four active S-Boxes in the first round.

The two characteristics are fully specified in Appendix B.

4.3 Message Modification, Matching, and Neutral Bits

In our distinguishers, we start constructing the internal differential boomerang pairs from the middle by fixing some bits of the intermediate states, which allows to pass low-probability events similarly to the rebound technique [26]. We define in particular the *boomerang switch* as the "middle" where we start constructing the state pairs to be the location where the two internal differential characteristics (or internal differentials) meet with the standard differential characteristic (see Figure 2). Note that the two surrounding χ steps (denoted χ_{int} in the internal characteristic and χ_{std} in the standard characteristic on Figure 2) usually have very low differential probabilities. However, since we start in the middle, we can fix partial state values such that these two steps are passed deterministically. Namely, this message modification technique allows to go through these two non-linear steps χ_{int} and χ_{std} without considering their probability.

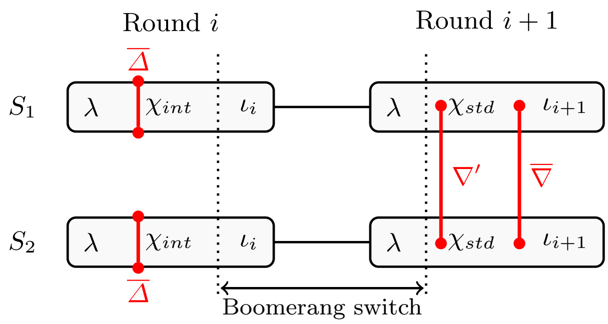

Figure 2: The boomerang switch: middle of distinguishing structure where the differentials on the two halves of the primitive meet.

Freedom degrees. There are three conditions imposed on the state pair (S_1, S_2) at the boomerang switch: the first two come from the internal differential characteristics, i.e. $\delta(S_1) = \delta(S_2) = \overline{\Delta}$, while the third is from the standard characteristic, i.e. $S_1 \oplus S_2 = \overline{\nabla}$. Therefore, in total, we have 800 bits of freedom; that is, once we fix the first half of S_1, then the second half of S_1 is fully determined, as well as the whole S_2.

The limited degrees of freedom may lead to contradictions. For instance, if there is an active S-Box in the first halves of S_1 and S_2, then the symmetry imposes than such S-Box must also be active in the second halves. If, in addition, these two halves differ in the bits that belong to the S-Boxes (which can occur when there is a non-zero internal difference at these bits), then it may not be possible to fix simultaneously the inputs to the S-Boxes in both of the halves.

Matching. To avoid such contradictions, we first have to make sure that *the internal characteristics and the standard characteristic can be matched*, i.e. there exist two states S_1 and S_2 at the boomerang switch (Figure 2), that can pass the χ_{int} and χ_{std} steps and that can produce differences as specified by the characteristic. Our extensive computer experiments have shown that if the differences at the boomerang switch are not sparse, then the chance of a match is extremely low[7].

To overcome this issue, we find (S_1, S_2) that produce the required differences $\overline{\Delta}$ at the input of χ_{int} and $\overline{\nabla}$ at the output of the χ_{std}, but not necessarily have the correct differences right at the boomerang switch[8]. By relaxing the difference constraint at the boomerang switch, and by trying different standard characteristics[9], we are able to match the characteristics.

Matching. This matching process is actually implemented by a message modification to partially fix values of the two states S_1 and S_2 to ensure that the boomerang can work by linking the two characteristics. As the output difference of χ_{int} is denser, we start the matching in the boomerang switch right at the output of χ_{int} (see Figure 2). First, from the fixed output difference $\overline{\nabla}$ of χ_{std}, we produce all possible input differences ∇', which defines the standard difference at the boomerang switch. We propagate each such difference to the output of χ_{int}, and then try to fix the values of all active S-Boxes of χ_{int}. If all the S-Boxes can be fixed, then the matching for χ_{int} is complete. During the matching, the values of some bits of the states S_1 and S_2 are being fixed, but there are still free (non-fixed) bits. We use the freedom of these bits to check if the active S-Boxes of χ_{std} can be passed. If so, then the matching is complete.

Neutral bits. The above process fixes some bits of S_1 and S_2 but there are more free bits and they can be used as neutral bits [7]. Namely, if S_1 and S_2 have fixed bits according to the matching, then for any value of the free remaining bits, the active S-Boxes of χ_{int} and χ_{std} still produce the required differences.

4.4 Internal Differential Boomerang Distinguishers for `Keccak`-$p_{i,n}$

We use the internal differential boomerang technique to distinguish the round-reduced `Keccak`-f permutation. The boomerangs are based on the internal differentials and characteristics from Section 4.1, and the standard differential characteristics from Section 4.2. To produce a boomerang pair, we start at the boomerang switch, and we first find the values of the fixed bits of S_1 and S_2 according to the message modification, which allows to pass the two rounds that surround the boomerang

[7]This only confirms the fact that for boomerangs (both classical and internal differential), finding the two characteristics for f and g does not guarantee that the boomerang will work – see [29] for more details.

[8]This is the reason why we have omitted specifying the differences at the output of the internal characteristics from Section 4.1, and at the input of the standard characteristics from Section 4.2.

[9]The internal characteristic cannot be changed as its difference propagation is completely defined by the round constants RC_i. On the other hand, there are many different standard characteristics (built upon parity kernels) that hold with the same probability.

switch. Then, we randomize the remaining neutral bits of the states and finally, from the two middle states, we produce the corresponding inputs and outputs. If the internal differences of each of the two inputs and the difference of the two outputs are as expected by the boomerang, then we have found the pair. Otherwise, we randomize again the neutral bits and repeat the procedure. An example of the overall description of the 8-round case is given in Figure 3.

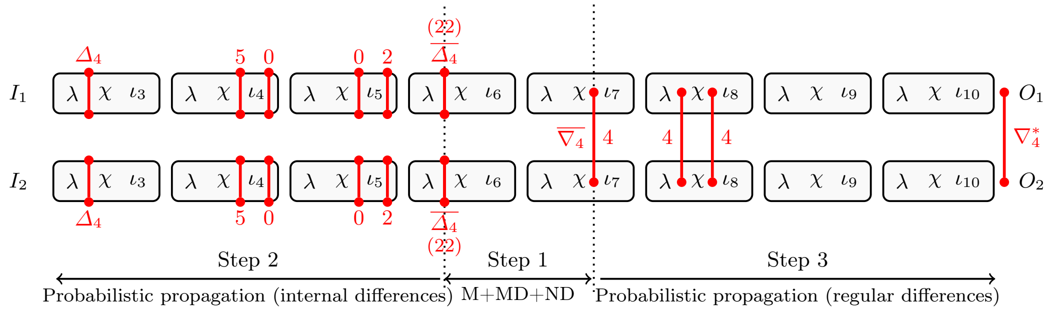

Figure 3: Example of the internal boomerang distinguisher in the case of Keccak-$p_{3,8}$. In step 1, we first perform the matching (M), then the message modification (MD) and we use neutral bits (ND). We finish the construction of the pair of inputs (I_1, I_2) with the probabilistic propagations in Step 2 and 3.

The query complexity of producing a pair is determined by the differential probability of the characteristics in all the rounds but the middle two[10]. We claim distinguishers for Keccak-$p_{i,n}$ for some (i, n) because the complexity of finding a boomerang pair for Keccak-$p_{i,n}$ is significantly lower compared to the complexity of producing a boomerang pair (with the same conditions on the input and output differences) for a random permutation defined by Lemma 1. In the four boomerangs below, the input internal difference is determined either in 800 bits (when $\mathcal{I}_3$ is used) or in 64 bits (when $\mathcal{ID}_4$ is used), while the output difference is determined either in 1278 bits (when $\mathcal{C}_3$ is used) or in 118 bits (when $\mathcal{C}_4$ is used). Therefore, by Lemma 1, the query complexity of producing a boomerang pair in the case of a random permutation requires at least $2^{57.5}$ queries.

Depending on the starting round i of Keccak-$p_{i,n}$, the boomerang pairs are produced for two cases. First, when the permutation starts at round 0, for the boomerang we use the first internal differential characteristic I_3 given in Section 4.1 and the standard characteristics $\mathcal{C}_3$, $\mathcal{C}_4$ given in Section 4.2. We can produce the boomerang pair for Keccak-$p_{0,6}$ by using the internal characteristic I_3 and the standard characteristic $\mathcal{C}_3$. As the probability of I_3 without χ_{int} is 2^{-2} and the probability of $\mathcal{C}_3$ without χ_{std} is 1 (recall both of these two χ steps are passed with the message modification), we can produce the boomerang pair with $2 \cdot 2^2 \cdot 2^2 \cdot 1 = 2^5$ queries to the 6-round permutation. Similarly, we can produce boomerang pair for Keccak-$p_{0,7}$ (we combine I_3 with $\mathcal{C}_4$) in $2 \cdot 2^2 \cdot 2^2 \cdot 2^8$ (the additional factor 2^8 is required to pass the 4 active S-boxes in the second round of $\mathcal{C}_4$), or approximately 2^{13} queries to the 7-round permutation.

Then, when the permutation starts at round 3, the boomerang uses the internal differential $\mathcal{ID}_4$ given in Section 4.1, and the standard characteristics $\mathcal{C}_3$, $\mathcal{C}_4$ from

[10]The cost of the message modification can be ignored because it is executed once, but it can be used for producing many boomerang pairs, thus on average it is negligible. The actual cost is around 2^8.

Section 4.2. The boomerang on Keccak-$p_{3,7}$, based on $\mathcal{ID}_4$ and $\mathcal{C}_3$, produces a pair with $2 \cdot 2^{4.6} \cdot 2^{4.6} \cdot 1 = 2^{10.2}$ queries. For Keccak-$p_{3,8}$ (see Figure 3), the boomerang is based on $\mathcal{ID}_4$ and $\mathcal{C}_4$, and for producing a boomerang pair, we need $2 \cdot 2^{4.6} \cdot 2^{4.6} \cdot 2^8 = 2^{18.2}$ queries.

Examples of boomerang pairs produced by the technique presented in this paper are given in Appendix C. We have also checked and confirmed the complexities of the four boomerangs given above. A summary of the distinguishers is given in Table 3.

Table 3: The internal differential boomerangs for Keccak-$p_{i,n}$ for $(i, n) \in \{(0, 6), (0, 7), (3, 7), (3, 8)\}$.

Rounds	**Internal**	**Standard**	**Prob. of internal**	**Prob. of standard**	**Prob. of the boomerang**	**Complexity of finding a pair**
6	$\mathcal{I}_3$	$\mathcal{C}_3$	2^{-68}	2^{-8}	2^{-140}	2^5
7	$\mathcal{I}_3$	$\mathcal{C}_4$	2^{-68}	2^{-16}	2^{-148}	2^{13}
7	$\mathcal{ID}_4$	$\mathcal{C}_3$	$2^{-48.6}$	2^{-8}	$2^{-105.2}$	$2^{10.2}$
8	$\mathcal{ID}_4$	$\mathcal{C}_4$	$2^{-48.6}$	2^{-16}	$2^{-113.2}$	$2^{18.2}$

5 Conclusions

We have presented the internal differential boomerang distinguishers, which are a combination of internal differentials and the boomerang technique. The new boomerangs can be used for cryptanalysis of functions and ciphers that have high-probability internal differentials. We have used the boomerangs to show non-randomness of reduced variants of the permutation Keccak-f. Based on truncated characteristics that hold with exceptionally high probability, and combined with a strong message modification, we have shown how to produce internal differential boomerang pairs for Keccak-f reduced to 6 rounds with only 2^5 queries to the permutation, 7 rounds with 2^{13} queries, and up to 8 rounds with 2^{18} queries.

Our results significantly outperform in terms of practical complexity all the previous cryptanalysis of Keccak-f. We emphasize that the results do not pose threat to the security of the future SHA-3 standard as there is no known way to date to extend the proposed reduced-round permutation distinguishers to the full sponge construction based on the full 24-round Keccak-f permutation. We were unable to extend our distinguishers to larger number of rounds while maintaining practical complexity. On the other hand, we leave as an open problem finding internal differential boomerang distinguishers that cover more rounds and that require theoretical complexity.

References

1. J.-P. Aumasson and W. Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. *rump session of Cryptographic Hardware and Embedded Systems-CHES*, 2009:67, 2009.
2. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

3. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Cryptographic Sponge Functions. online.
4. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. The keccak reference (version 3).
5. G. Bertoni, J. Daemen, M. Peeters, G. V. Assche, and R. V. Keer. Ketje v1. Submitted to the CAESAR competition, March 2014.
6. G. Bertoni, J. Daemen, M. Peeters, G. V. Assche, and R. V. Keer. Keyak v1. Submitted to the CAESAR competition, March 2014.
7. E. Biham and R. Chen. Near-collisions of SHA-0. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2004.
8. A. Biryukov, M. Lamberger, F. Mendel, and I. Nikolić. Second-order differential collisions for reduced SHA-256. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 270–287. Springer, 2011.
9. A. Biryukov, I. Nikolić, and A. Roy. Boomerang attacks on BLAKE-32. In A. Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 218–237. Springer, 2011.
10. C. Boura and A. Canteaut. Zero-sum distinguishers for iterated permutations and application to Keccak-f and Hamsi-256. In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2010.
11. C. Boura, A. Canteaut, and C. De Cannière. Higher-order differential properties of Keccak and Luffa. In A. Joux, editor, *Fast Software Encryption – FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, Feb. 2011.
12. A. Canteaut, editor. *Fast Software Encryption – FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*. Springer, Mar. 2012.
13. S. Das and W. Meier. Differential biases in reduced-round Keccak. In Pointcheval and Vergnaud [35], pages 69–87.
14. I. Dinur, O. Dunkelman, and A. Shamir. New attacks on Keccak-224 and Keccak-256. In Canteaut [12], pages 442–461.
15. I. Dinur, O. Dunkelman, and A. Shamir. Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In Moriai [28], pages 219–240.
16. I. Dinur, O. Dunkelman, and A. Shamir. Improved practical attacks on round-reduced Keccak. *Journal of Cryptology*, 27(2):183–209, Apr. 2014.
17. I. Dinur, P. Morawiecki, J. Pieprzyk, M. Srebrny, and M. Straus. Practical complexity cube attacks on round-reduced Keccak sponge function. *IACR Cryptology ePrint Archive*, 2014:259, 2014.
18. A. Duc, J. Guo, T. Peyrin, and L. Wei. Unaligned rebound attack: Application to Keccak. In Canteaut [12], pages 402–421.
19. J. Guo, I. Nikolić, T. Peyrin, and L. Wang. Cryptanalysis of zorro. *IACR Cryptology ePrint Archive*, 2013:713, 2013.
20. J. Kelsey, T. Kohno, and B. Schneier. Amplified boomerang attacks against reduced-round MARS and serpent. In B. Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer, 2000.
21. L. R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.
22. S. Kölbl, F. Mendel, T. Nad, and M. Schläffer. Differential cryptanalysis of Keccak variants. In M. Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 141–157. Springer, 2013.
23. S. Kuila, D. Saha, M. Pal, and D. R. Chowdhury. Practical distinguishers against 6-round Keccak-f exploiting self-symmetry. In Pointcheval and Vergnaud [35], pages 88–108.
24. X. Lai. Higher order derivatives and differential cryptanalysis. In *Communications and Cryptography*, pages 227–233. Springer, 1994.

25. M. Lamberger, F. Mendel, M. Schläffer, C. Rechberger, and V. Rijmen. The rebound attack and subspace distinguishers: Application to whirlpool. *Journal of Cryptology*, pages 1–40, 2013.
26. F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In O. Dunkelman, editor, *Fast Software Encryption – FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 260–276. Springer, Feb. 2009.
27. P. Morawiecki, J. Pieprzyk, and M. Srebrny. Rotational cryptanalysis of round-reduced Keccak. In Moriai [28], pages 241–262.
28. S. Moriai, editor. *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*. Springer, Mar. 2013.
29. S. Murphy. The return of the cryptographic boomerang. *IEEE Transactions on Information Theory*, 57(4):2517–2521, 2011.
30. National Institute of Standards and Technology. Cryptographic hash algorithm competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`.
31. National Institute of Standards and Technology. *Draft FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.*
32. M. Naya-Plasencia, A. Röck, and W. Meier. Practical analysis of reduced-round Keccak. In D. J. Bernstein and S. Chatterjee, editors, *Progress in Cryptology - INDOCRYPT 2011: 12th International Conference in Cryptology in India*, volume 7107 of *Lecture Notes in Computer Science*, pages 236–254. Springer, Dec. 2011.
33. I. Nikolić, J. Pieprzyk, P. Sokolowski, and R. Steinfeld. Known and chosen key differential distinguishers for block ciphers. In K. H. Rhee and D. Nyang, editors, *Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers*, volume 6829 of *Lecture Notes in Computer Science*, pages 29–48. Springer, 2010.
34. T. Peyrin. Improved differential attacks for ECHO and Grøstl. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 370–392. Springer, Aug. 2010.
35. D. Pointcheval and D. Vergnaud, editors. *AFRICACRYPT 14: 7th International Conference on Cryptology in Africa*, volume 8469 of *Lecture Notes in Computer Science*. Springer, May 2014.
36. D. Wagner. The boomerang attack. In L. R. Knudsen, editor, *Fast Software Encryption – FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, Mar. 1999.

A Round Constants

Table 4: Rounds constants RC used in the `Keccak-`f permutation. The table also shows the hamming weight w_i of the internal difference $\delta(RC[i])$ of the round constant $RC[i]$ used in the ith round.

i	$RC[i]$	$\delta(RC[i])$	w_i	i	$RC[i]$	$\delta(RC[i])$	w_i
0	0x0000000000000001	0x00000001	1	12	0x000000008000808b	0x8000808b	6
1	0x0000000000008082	0x00008082	3	13	0x800000000000008b	0x8000008b	5
2	0x800000000000808a	0x8000808a	5	14	0x8000000000008089	0x80008089	5
3	0x8000000080008000	0x00008000	1	15	0x8000000000008003	0x80008003	4
4	0x000000000000808b	0x0000808b	5	16	0x8000000000008002	0x80008002	3
5	0x0000000080000001	0x80000001	2	17	0x8000000000000080	0x80000080	2
6	0x8000000080008081	0x00008081	3	18	0x000000000000800a	0x0000800a	3
7	0x8000000000008009	0x80008009	4	19	0x800000008000000a	0x0000000a	2
8	0x000000000000008a	0x0000008a	3	20	0x8000000080008081	0x00008081	3
9	0x0000000000000088	0x00000088	2	21	0x8000000000008080	0x80008080	3
10	0x0000000080008009	0x80008009	4	22	0x0000000080000001	0x80000001	2
11	0x000000008000000a	0x8000000a	3	23	0x8000000080008008	0x00008008	2

B Internal and Standard Differentials for the Boomerangs

B.1 Internal differential and characteristic

The internal differential characteristic $\mathcal{I}_3$ has an input difference $\varDelta_3$ fully specified in all 800 bits. The output difference of this characteristic (as shown before) is irrelevant, however, the difference at the input of the last χ must be fixed. We denote this difference as $\overline{\varDelta_3^*}$ and note that it is defined in all 800 bits as well. See Table 5 for the values of $\varDelta_3$ and $\overline{\varDelta_3^*}$, where a dash (-) represents a zero bit difference.

Table 5: Internal differential characteristic $\mathcal{I}_3$.

	a6bc4d78	2f135e26	44d789af	b5e26bc4	f89af135
	a6bc4d79	2f135e26	44d789af	b5e26bc4	f89af135
$\varDelta_3$	a6bc4d79	2f135e26	44d789af	b5e26bc4	f89af135
	a6bc4d79	2f135e26	44d789af	b5e26bc4	f89af135
	a6bc4d79	2f135e26	44d789af	b5e26bc4	f89af135
	----8-82	-8-82---	--------	--------	4-41----
	--------	1-4---1-	--------	1-1-4---	--------
$\overline{\varDelta_3^*}$	---1-1-4	--------	--------	-1-1-4--	--------
	2----8-8	--------	-2-2-8--	--------	--------
	--------	--------	--8-82--	--------	---2-2-8

The internal differential $\mathcal{ID}_4$ has an input truncated difference $\varDelta_4$ specified in 64 bits, more precisely, in 55 bits the difference is 0, and in 9 bits the difference is 1. We use the mask $\varDelta_4^0$ to show the bits that have values of 0, and the mask $\varDelta_4^1$ for the bits with difference 1. As before, with $\overline{\varDelta_4^*}$ we denote the difference defined in 800 bits at the input of χ step of the last round of $\mathcal{ID}_4$. See Table 6 for the values of $\varDelta_4^0$, $\varDelta_4^1$ and $\overline{\varDelta_4^*}$.

Table 6: Internal differential $\mathcal{ID}_4$.

	------38	------38	------38	------38	------38
	------3-	------3-	------3-	------3-	------3-
$\varDelta_4^0$	------3-	------3-	------3-	------3-	------3-
	------3-	------3-	------3-	------3-	------3-
	------3-	------3-	------3-	------3-	------3-
	--------	-------4	--------	--------	--------
	-------8	-------4	--------	--------	--------
$\varDelta_4^1$	-------8	-------4	--------	--------	--------
	-------8	-------4	--------	--------	--------
	-------8	-------4	--------	--------	--------
	8------1	----18--	--------	--------	----c---
	--------	--3-----	--------	----3---	--------
$\overline{\varDelta_4^*}$	-------3	--------	--------	-----3--	--------
	18------	--------	-----6--	--------	--------
	--------	--------	-----18-	--------	-------6

B.2 Standard differential characteristics

In the standard differential characteristic $\mathcal{C}_3$, the input difference is irrelevant, but the difference at the input of χ^{-1} in the first round is fixed in all 1600 bits: we denote it $\overline{\nabla_3}$. The output difference of $\mathcal{C}_3$ is truncated, and with ∇_3^{*0} (resp. ∇_3^{*1}), we denote the masks of the bits that have 0 (resp. 1) difference. See Table 7 for the values of $\overline{\nabla_3}$, ∇_3^{*0} and ∇_3^{*1}.

Table 7: Standard differential characteristic $\mathcal{C}_3$.

$\overline{\nabla_3}$	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----8-------8---`	`----------------`	`----------------`
	`----------------`	`----------------`	`----8-------8---`	`----------------`	`----------------`
∇_3^{*0}	`7cffe1ff7cffe1ff`	`fcf7f1f9fcf7f1f9`	`faf7e9f9faf7e9f9`	`7bf7eff97bf7eff9`	`79ffe7ff79ffe7ff`
	`fbfb7cfafbfb7cfa`	`fbfb67fafbfb67fa`	`e3ff63fbe3ff63fb`	`e3ffe-ffe3ffe-ff`	`e3fbf8fee3fbf8fe`
	`ff73bf8eff73bf8e`	`ff63ff8fff63ff8f`	`bf6dff9fbf6dff9f`	`bfedbfbebfedbfbe`	`bff1bfcebff1bfce`
	`fd7efd77fd7efd77`	`ef7e9df7ef7e9df7`	`ef3f9dcfef3f9dcf`	`edbf9f4fedbf9f4f`	`fdbeff47fdbeff47`
	`cfd7f7c7cfd7f7c7`	`ffd7fec7ffd7fec7`	`ffd77eddffd77edd`	`cfdf76fdcfdf76fd`	`cfdf77e5cfdf77e5`
∇_3^{*1}	`----1-------1---`	`----------------`	`-----4-------4--`	`-------4-------4`	`----------------`
	`-----2-------2--`	`---4-------4----`	`----8-------8---`	`----------------`	`1-------1-------`
	`----------------`	`----------------`	`----------------`	`------4-------4-`	`4-------4-------`
	`-2-------2------`	`---1-------1----`	`----------------`	`----4-------4---`	`------2-------2-`
	`2-------2-------`	`------1-------1-`	`------2-------2-`	`--2-------2-----`	`----------------`

Finally, for $\mathcal{C}_4$ we use $\overline{\nabla_4}$ to denote the difference at the input of χ^{-1} in the first round, and $\nabla_4^{*0}, \nabla_4^{*1}$ to denote the masks of bits that have 0 and 1. See Table 8 for the values of $\overline{\nabla_4}$, ∇_4^{*0} and ∇_4^{*1}.

Table 8: Standard differential characteristic $\mathcal{C}_4$.

$\overline{\nabla_4}$	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----8-------8---`	`----------------`	`----------------`
	`----------------`	`----------------`	`----8-------8---`	`----------------`	`----------------`
∇_4^{*0}	`----------------`	`------8-------8-`	`----------------`	`----2-------2---`	`----6-------6---`
	`----------------`	`-----8-------8--`	`4-----2-4-----2-`	`--2-4-2---2-4-2-`	`----------------`
	`2--48---2--48---`	`----------------`	`1------11------1`	`9-------9-------`	`8-------8-------`
	`88-4----88-4----`	`-8----4--8----4-`	`-9---1---9---1--`	`-8---4---8---4--`	`88-2---888-2---8`
	`8---8-8-8---8-8-`	`4----8--4----8--`	`4-------4-------`	`----------------`	`-4-------4------`
∇_4^{*1}	`----------------`	`----------------`	`2-------2-------`	`----------------`	`-----1-------1--`
	`-----8-------8--`	`4------44------4`	`----------------`	`----------------`	`---8-------8----`
	`----------------`	`1-------1-------`	`----------------`	`----------------`	`----------------`
	`----------------`	`----------------`	`----------------`	`8--88--88--88--8`	`----------------`
	`4------44------4`	`--4-------4-----`	`----------------`	`-4-------4------`	`----------------`

C Examples of Boomerang Pairs for the 7- and 8-Round Permutations

Table 9: Example of one boomerang pair for the distinguisher on the 7-round Keccak-$p_{0,7}$ permutation.

lane	I1	$\delta(\texttt{I1}) \oplus \Delta_3$	I2	$\delta(\texttt{I2}) \oplus \Delta_3$	O1	O2	$(\texttt{O1} \oplus \texttt{O2}) \wedge \nabla_4^{*0} \oplus (\texttt{O1} \oplus \texttt{O2}) \wedge \nabla_4^{*1} \oplus \nabla_4^{*1}$
1	04297594a29538ec	0	d836ad227e8ae05a	0	2ff460b4a66b587d	cfb935c695325af6	0
2	f8cf1290d7dc4cb6	0	50c0b8e97fd3e6cf	0	acfd38ddf059e4d3	55eacaeeaeafe29e	0
3	bff3a92efb242081	0	58be84ab1c690d04	0	3d6a3f48eb6674b1	c65f85435e0ba89b	0
4	4c1f9239f9fdf9fd	0	1b12a26caef0c9a8	0	fc5ca78f24d20b70	ede6e40341bb0f26	0
5	8c9263d1740892e4	0	490f48a4b195b991	0	ab10fca9dc6a69a2	c4767f43bfc67e23	0
6	179b7707b1273a7e	0	8dbb6f8e2b0722f7	0	d492165be3e7056d	1ea52fd7e58d6aad	0
7	bc13608093003ea6	0	6823cb4a4730956c	0	69706a1778841e38	8247091b95dd8bfd	0
8	a94ab26aed9d3bc5	0	4ed6232d0a01aa82	0	e0ab378b4b5ccb4d	c2fd0201e8a02881	0
9	70a5e121c5478ae5	0	3338030b86da68cf	0	8dfb4e008ecbc72a	00bd4a5bcd8df3e9	0
10	ff0596ce079f67fb	0	d381763f2b1b870a	0	2c0c2ab1af37e5a3	2064eea68bec4f36	0
11	8035f9f72689b48e	0	8595312823297c51	0	96feb94a8b718549	8b74e9a958abfac1	0
12	8981a1f9a692ffdf	0	9a9cca2bb58f940d	0	91a6546141022c00	cceaf97652a76b4c	0
13	d6d260719205e9de	0	de4cc42f9a9b4d80	0	5d3f2e5df6baefe1	394aa92b533f8fa9	0
14	12f4d449a716bf8d	0	28b9b2679d5bd9a3	0	42dafdda5f8f04de	63a2734c17eae8c2	0
15	3317de9bcb8d2fae	0	9e438d6666d97c53	0	66bfb3869edc1cb6	2ab8f549b2f37486	0
16	4d67a1e2ebdbec9b	0	c69b25e460027689d	0	3bee2c685ee09a5d	0dbfc16079a9d995	0
17	d553cd7cfa40935a	0	0e55266a2146784c	0	f8de90d173040d76	fd9b1c64f5b5f842	0
18	98e0740cdc37fda3	0	f25d45a9b68acc06	0	cd9bcc2cbcee0a64	596aa8080e88da5d	0
19	1ffbb84eaa19d38a	0	4ca55c3bf94737ff	0	8f5ee77f40cff87b	3ff2445181b00945	0
20	c8ff2b5e3065da6b	0	b19d676f4907965a	0	5f4b839eb90bdf71	4f9b0ab9fa8abab3	0
21	cb067ace6dba37b7	0	fc9ae1505a26ac29	0	79e316823fe28a70	319e5bef6acd8a14	0
22	84d5320babc66c2d	0	10988fec3f8bd1ca	0	0cbadb5dff76ef53	37c18a89580c1f00	0
23	c40499a980d31006	0	992380abddf40904	0	9c76542a311c0abc	24703f722d567cef	0
24	6fa4b841da46d385	0	9e78edbe2b9a867a	0	655a86ad90eb701b	0aeacccd5c9bdedb	0
25	25684fa3ddf2be96	0	30598a28c8c37b1d	0	ad2ac125e9aa1143	2cff71db21eb7843	0

Table 10: Example of one boomerang pair for the distinguisher on the 8-round Keccak-$p_{3,8}$ permutation.

lane	I1	$\delta(\mathtt{I1}) \wedge \Delta_4^0 \oplus$ $\delta(\mathtt{I1}) \wedge \Delta_4^1 \oplus \Delta_4^1$	I2	$\delta(\mathtt{I2}) \wedge \Delta_4^0 \oplus$ $\delta(\mathtt{I2}) \wedge \Delta_4^1 \oplus \Delta_4^1$	O1	O2	$(\mathtt{O1} \oplus \mathtt{O2}) \wedge \nabla_4^{*0} \oplus$ $(\mathtt{O1} \oplus \mathtt{O2}) \wedge \nabla_4^{*1} \oplus \nabla_4^{*1}$
1	df2601755e189c35	0	a4950f83b9ccdf07	0	1458c29aef269226	d605870cf5cdc856	0
2	9771e76a97ecac6e	0	24378c432c7bdf87	0	30af390793fd7b5b	6d8c2a32dd5ab16d	0
3	31112c52b3d1ce12	0	7a3983cee541bd4e	0	b051f665429c9e00	8d967b64b7b3de2b	0
4	c5819876c7bf0af5	0	b064e65cadc34019	0	5881bb8e471643e5	0a47ecc4d024d3c5	0
5	07cf2d38043a8b3a	0	2c7d7960ad2de5e2	0	c8eff9acedb4c659	14056091ee525361	0
6	c30beece42b0e140	0	40ac766a44559765	0	a781c2cd1de14c43	3d098d1e9bb1c670	0
7	c179ae84c16b3789	0	f946f7057a134c8b	0	7f26547b1b0d4c64	0c113664454e3ec3	0
8	077cfa550698881d	0	ebaab094f8015e57	0	25d275603bdd5633	80a054f72bd266ba	0
9	c3842bd840b9c351	0	6e757dc7e78b934a	0	b84179f4be9ca0b6	8fc4c1fb25c31ce4	0
10	aea801a4add3776f	0	c6e73824c0e6ede9	0	479302ba587b718c	39ea80efa2b351b0	0
11	dc274b67df809dad	0	1e3d27661a66c2a9	0	fa3d88c2a57cdc64	bf6dfd6170fffc4d	0
12	4470bd7344c876f4	0	7fd57a9efde2cc1b	0	ee147bde12fbc16e	d2f43e8f26db7e06	0
13	e09ca53962444539	0	ed461cb6e7081e3e	0	c8e455591b641fef	6d7d2cddb20d1ef7	0
14	98b2153c98ad017f	0	9eaa7983171da24d	0	ebd6ebc066ecbd56	c38ed5340fc9f545	0
15	71a6f95af3fe4f58	0	db2bcc7ada2a84b2	0	62d37513fc20daf8	2dce6fa9a4ea319e	0
16	2a1acacfa9377bc4	0	3c14dbda2a647f17	0	5947750b4dfe6f84	2c3c55cb6f1f762f	0
17	13e9c92f11c73b20	0	090ccbd68a48bcd1	0	70710d1624be95e2	b4998a3ae50abee1	0
18	582367c05b824200	0	c53dd714de50fb16	0	373043f713784fd9	c716c5fcf720ddfa	0
19	ee5816686c4739e2	0	c2fd13865f5e7142	0	c311f883e46d6e75	633d191d41959e4f	0
20	708f73d6f3ee1294	0	8a792bce8e38b748	0	4b2f7c577e260da8	7b5b7ee378231fee	0
21	112e05779087c4bc	0	16beb10180edf4cb	0	1f9ff09274f70794	738aadc707ca44e9	0
22	8bb5aa570a12f0d3	0	859e4c9a96b0795c	0	732edb20f90d58b8	e3726fd8de55497a	0
23	ca1a320b4a8e960a	0	1d0dfc1b8a033f1c	0	6c2757da0f8fe149	78c17a62ba358298	0
24	4bc798a04a591fe9	0	e695b6aaf93658ab	0	ac3680743b3f1e01	612685f4d7ed182d	0
25	1cfa2a551c33d056	0	445651d35403081d	0	4aaf9eaf2682491a	a30b3c85be5960e6	0