IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, 40 pages.

Differential Collision Resistance of Feed-Forward Sponge Variants

Ruixi Lv[1,2], Guozhen Liu[a,1,2], Shun Li[1,2] and Siwei Sun[1,2]

[1] School of Cryptology, University of Chinese Academy of Sciences, Beijing, China
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, China

Abstract

Feed-forward sponge variants have recently been proposed in independent lines of work to overcome the capacity bound limitation of the sponge construction and to strengthen provable (second-)preimage security. Representative examples include `Sponge-DM`, `Sponge-DM`^{c}, `Sponge-EDM`^{c}, and `Sponge-F`$^{\mathrm{P,pd}}$. While these designs improve generic security guarantees, their behavior under dedicated cryptanalysis remains largely unexplored. Focusing on differential collision attacks, we review the inner and outer collision frameworks for sponge hashes and analyze how the feed-forward operation, the rate-capacity ratio, and the squeezing phase affect collision cryptanalysis. Our analysis shows that feed-forward sponge variants do not necessarily offer stronger collision resistance than the standard sponge, and that the modified collision conditions and the enlarged rate can even facilitate attacks. We substantiate this analysis on feed-forward sponge instantiations with Keccak-p[1600,24] that are configured to match the generic collision security levels of `SHA3-224`, `SHA3-256`, `SHA3-384`, and `SHA3-512`. For these instantiations, we develop classical and quantum collision attacks. Compared with the best known attacks on the corresponding `SHA-3` instances, our attacks reach more rounds or lower complexity in several cases. These results demonstrate that, for the reduced-round instantiations considered in this work, the structural changes introduced by feed-forward sponges do not carry over to collision resistance against dedicated classical and quantum attacks. They also indicate that strong collision resistance still relies on a careful choice of the feed-forward sponge configuration and on the differential propagation of the underlying permutation.

Keywords: Feed-Forward Sponge, Differential Collision Resistance, `SHA-3`, `Keccak`

1 Introduction

In recent years, the potential threat of quantum computing has significantly increased the demand for cryptographic primitives with longer security lengths. Many organizations worldwide are accelerating the development of new standards and the migration to more secure algorithms. While symmetric ciphers are generally considered more resistant to quantum attacks than public-key schemes, the practice of simply doubling the hash length is receiving renewed scrutiny. For instance, the recent Next-generation Commercial Cryptographic Algorithms Program (NGCC) in China [Ins25] requires candidate hash functions to support output lengths of at least 512 and 1024 bits. For an n-bit output,

Emails: lvruixi24@mails.ucas.ac.cn (Ruixi Lv), liuguozhen@ucas.ac.cn (Guozhen Liu), lishun@ucas.ac.cn (Shun Li), sunsiwei@ucas.ac.cn (Siwei Sun)
[a]Corresponding Author

This work is licensed under a "CC BY 4.0" license.

these candidates must achieve classical security strengths of at least $n/2$ bits against collision attacks and n bits against preimage attacks.

Two main paradigms dominate modern hash function design, namely the Merkle-Damgård (MD) construction and the sponge construction [BDPV11]. Compared to the MD paradigm, the sponge approach provides provable security bounds in the ideal permutation model and naturally avoids structural weaknesses such as the length-extension attack. Owing to these advantages, permutation-based hashes built upon the sponge framework have become a prevalent choice in modern cryptography, as exemplified by the NIST `SHA-3` standard [Dwo15] and the lightweight standard `Ascon-Hash256` [TMC+25]. In the standard sponge, however, these provable security margins are bounded jointly by the digest length n and the capacity c. Specifically, the construction tightly achieves $\min\{n/2, c/2\}$-bit collision resistance, $\min\{n, c/2\}$-bit second-preimage resistance, and $\min\{n, \max\{n-r, c/2\}\}$-bit preimage resistance [LM22]. Consequently, reaching a specific target security level requires the capacity to be at least twice the intended security margin. This strict requirement inevitably forces a reduction in the rate and restricts the overall performance of the hash function. Therefore, developing improved architectural designs and subjecting their concrete instantiations to rigorous cryptanalysis remain tasks of great practical importance.

To overcome the capacity bound, several *feed-forward sponge variants* have been proposed recently. Sun *et al.* [SLZ+26] introduced the `Sponge-DM`, `Sponge-DM`^{c} and `Sponge-EDM`^{c} constructions, which add full or partial permutation input to the output. Similarly, Guo *et al.* [GHJ+26] proposed `Sponge-F`$^{\mathrm{P,pd}}$, which feeds the capacity (inner) part forward. By feeding part or all of the permutation input forward to the output, these designs block the generic attacks that rely on inverting the underlying permutation. This significantly improves the provable preimage and second-preimage resistance, and in turn allows the capacity requirement to be relaxed without sacrificing the provable security margin.

Although these feed-forward constructions offer enhanced provable preimage and second-preimage security, their impact on dedicated cryptanalysis remains largely unexplored for collision, preimage, and second-preimage attacks. Dedicated cryptanalysis depends on the attack framework, which is shaped by the cryptanalytic goal, the sponge configuration, and the properties of the underlying permutation. These research lines involve different constraints and techniques, and results obtained in one setting do not in general directly extend to another; each therefore requires a separate analysis. For example, preimage attacks may rely on meet-in-the-middle [QHD+23, DZQ+24, QZH+25] or linearization techniques [GLS16, LS19, LLS+26], whereas differential collision attacks use differential trails and connector construction [GLL+20, GLST22], and internal differential attacks follow a different framework [DDS13, ZHL24]. Besides the direct structural modification introduced by feed-forward itself, many feed-forward instantiations permit a smaller capacity when configured to match the generic security targets of the corresponding NIST standard hashes, especially those with large state size like `Keccak`. For a fixed permutation width, this yields a larger rate and hence more message Degrees of Freedom (DoF). Attack frameworks whose feasibility is sensitive to the available DoF while remaining applicable after the introduction of feed-forward are therefore more directly affected by this parameter change. *Differential collision cryptanalysis* fits this setting: more message DoF can directly facilitate connector construction, while the overall differential collision framework remains applicable after accounting for the corresponding collision conditions. If a differential trail has zero output difference on the digest bits (a colliding differential trail) and admits message pairs that follow it, then the outputs collide.

Moreover, theoretical security bounds generally assume an ideal permutation and often fail to capture the structural vulnerabilities of specific instantiations, whereas real-world attacks rely heavily on the concrete propagation properties of the underlying permutation.

It is therefore essential to assess whether feed-forward introduces unexpected weaknesses in the collision resistance of instantiated hash functions.

Differential collision attacks on sponge hash functions. Differential collision attacks underpin much of the collision cryptanalysis of sponge instances, including `SHA-3` (`Keccak`) and `Ascon-Hash256`. Yet the constraint-input/constraint-output (CICO) feature of sponge construction makes such differential collision attacks more difficult than for a bare permutation. The adversary needs not only to find a colliding differential trail, but also to satisfy the conditions that appear during message absorption. Two bottlenecks therefore dominate this line of work, *i.e.*, finding colliding differential trails over the underlying permutation, which generally has a large internal state, and connecting the trail to the absorbing-phase input constraints. In particular, the difficulty of the connecting step depends strongly on the rate-capacity ratio. For example, connecting techniques that work for `SHA-3` instances with a large rate-capacity ratio (such as `SHA3-256`) often do not carry over to instances with a smaller ratio (such as `SHA3-512`).

In FSE 2012, Dinur *et al.* [DDS12] proposed the Target Difference Algorithm (TDA) to connect the input constraint to the colliding trail, and this led to a substantial advance for `Keccak` and `SHA-3` cryptanalysis. In particular, they gave the first practical 4-round collision attacks on several `Keccak` instances. Guo *et al.* extended this line in two ways, with related results published from 2017 to 2022. First, they developed trail search techniques [LQT19, GLST22]. Second, they analyzed the nonlinear layer to improve the connecting step and obtained the first practical 5-round and theoretical 6-round collision attacks on selected `SHA-3` instances [QSLG17, SLG17, GLL+20, GLST22]. In ToSC 2022, Huang *et al.* also continued this line of differential collision attacks on `SHA-3`, but used different connecting techniques for targets with a relatively small rate-capacity ratio such as `SHA3-384` in which the original TDA approach does not apply [HBDM22]. Within this line of work, differential collision attacks on `SHA-3` have reached up to six rounds, the strongest round coverage reported for `SHA-3`. Another research line focusing on internal differential collisions for `SHA-3` was proposed by Dinur *et al.* [DDS13] and extended by Zhang *et al.* [ZHL23, ZHL24]. Since that approach targets internal collisions rather than the standard differential collision investigated in this work, we do not elaborate on its technical details here.

By contrast, for the other major sponge hash standard `Ascon-Hash256`, the literature on dedicated differential collision attacks remains much more limited than for `SHA-3`. In 2021, Gérault *et al.* [GPT21] used a CP model to find a 2-round differential trail of probability 2^{-156}. In 2023, Yu *et al.* [YLW+23] reduced the complexity of 2-round collisions to $2^{62.6}$ with an optimized attack framework, which was later lowered to $2^{61.79}$ with tuned parameters by Liu *et al.* [LLL+25].

Our contribution. In this paper, we present a structural analysis of how feed-forward changes differential collision attacks. We then apply this analysis to the `Keccak`-based feed-forward instantiations proposed to match the generic security levels of `SHA-3`. Our contributions are summarized as follows.

- **A structural analysis of differential collision attacks on feed-forward sponge variants.** We revisit previous differential collision attacks on sponge constructions and classify them into two categories, namely outer collision attacks and inner collision attacks. This viewpoint provides a unified framework for understanding how differential collision attacks interact with sponge hash constructions. Using this framework, we analyze the effect of feed-forward operations on collision cryptanalysis and identify the situations in which existing attacks can be transferred directly, as well as the situations in which the feed-forward mechanism changes the collision conditions. Our analysis shows that feed-forward sponge variants do not necessarily

provide stronger resistance against dedicated differential collision attacks and may even reduce the attack complexity for certain parameter settings.

- **Concrete collision attacks on feed-forward sponge hashes instantiated with `Keccak`.** We apply the above analysis to several feed-forward sponge instantiations with Keccak-p[1600,24], configured to match the generic collision security of `SHA3-224`, `SHA3-256`, `SHA3-384`, and `SHA3-512`, and evaluate their concrete collision resistance. The results are summarized in Table 1. For the counterparts of `SHA3-224` and `SHA3-256`, we obtain the first classical 6-round collision attacks. For the counterparts of `SHA3-384` and `SHA3-512`, we obtain practical attacks that are not known for the corresponding `SHA-3` instances. These results confirm that the structural observations derived from our framework are reflected in concrete cryptanalytic settings.
- **Insights for feed-forward sponge hash design.** Our observations mainly concern settings where connector construction is sensitive to the rate-capacity configuration and the underlying permutation has a low-degree nonlinear layer. Within the differential attack framework considered here, the concrete attack behavior depends on the interaction among the rate-capacity configuration, the digest length, the squeeze rate, the feed-forward structure, and the differential propagation of the underlying permutation. In the outer collision setting, full-state feed-forward replaces the zero outer output difference condition by the requirement that the outer input and output differences coincide, which does not make colliding trail search harder and in several of our feed-forward `Keccak` instances even makes it easier. At the same time, the enlarged rate adopted in several proposed `Keccak`-based feed-forward instances supplies more message DoF and facilitates connector construction, while full-state feed-forward can make connector construction harder for large digest lengths. Despite these structural effects, none of our attacks extends beyond 6 rounds, which indicates that strong differential propagation of the underlying permutation remains the dominant limiting factor.

Organization. This paper is organized as follows. Section 2 introduces the feed-forward sponge design and briefly reviews the differential collision methods on `SHA-3`. Section 3 presents the generic differential collision frameworks for sponge hashes and analyzes the structural impact of the feed-forward mechanism. Section 4 provides the detailed collision attacks on the specific feed-forward instances and compares the results with those for the corresponding `SHA-3` instances, and briefly addresses the transfer of known `Ascon-Hash256` attacks to the `Ascon`-based variants. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 The Sponge and Its Feed-Forward Variants

The standard sponge construction. The sponge construction [BDPV11] is built from a b-bit permutation $f : \{0,1\}^b \to \{0,1\}^b$ operating on a state of size $b = r + c$, where the state is divided into an outer part of size r and an inner part of size c, referred to as the *rate* and *capacity*, respectively. As illustrated in Figure 1, the padded message is divided into r-bit blocks m_i and sequentially absorbed into the outer part of the state. The state update is defined as

$$S_{i+1} = f\big(S_i \oplus (m_i \,\|\, 0^c)\big).$$

After the absorbing phase, output blocks are iteratively extracted from the outer part of the state, with applications of f between successive extractions until the required digest length is reached.

Table 1: Comparison of collision attack results

Target	Type	Round	Time	Generic	Reference
SHA3-224	Classical	5	Practical	2^{112}	[GLL+20]
SHA3-224	Quantum	6	$2^{97.5}/\sqrt{S}$	$2^{112}/S^{\dagger}$	[GLST22]
Keccak-EDM^{c}[288, 224]	**Quantum**	**6**	$\mathbf{2^{80.0}/\sqrt{S}}$	$2^{112}/S$	Sect. 4.2
Keccak-DM[288, 224]	**Classical**	**6**	$\mathbf{2^{68}}$	2^{112}	Sect. 4.2
SHA3-256	Classical	5	Practical	2^{128}	[GLL+20]
SHA3-256	Quantum	6	$2^{104.25}/\sqrt{S}$	$2^{128}/S$	[GLST22]
Keccak-DM[320, 256]	**Classical**	**5**	**Practical**	2^{128}	Sect. 4.3
Keccak-EDM^{c}[320, 256]	**Quantum**	**6**	$\mathbf{2^{88.3}/\sqrt{S}}$	$2^{128}/S$	Sect. 4.3
Keccak-DM[320, 256]	**Classical**	**6**	$\mathbf{2^{75}}$	2^{128}	Sect. 4.3
SHA3-384	Classical	4	$2^{56.11}$	2^{192}	[CHCW26]
SHA3-384	Classical	5	$2^{170.73}$	2^{192}	[ZHL24]
Keccak-DM[448, 384]	**Classical**	**4**	**Practical**	2^{192}	Sect. 4.4
Keccak-DM[448, 384]	**Classical**	**5**	**Practical**	2^{192}	Sect. 4.4
Keccak-EDM^{c}[448, 384]	**Classical**	**5**	**Practical**	2^{192}	Sect. 4.4
SHA3-512	Classical	3	Practical	2^{256}	[DDS13]
SHA3-512	Classical	4	$2^{225.29}$	2^{256}	[ZHL24]
Keccak-DM[576, 512]	**Classical**	**4**	$\mathbf{2^{238}}$	2^{256}	Sect. 4.5
Keccak-EDM^{c}[576, 512]	**Classical**	**4**	**Practical**	2^{256}	Sect. 4.5
Keccak-Sp-F$_{512}$	**Classical**	**4**	**Practical**	2^{256}	Sect. 4.5

† S denotes the size of the quantum computer.

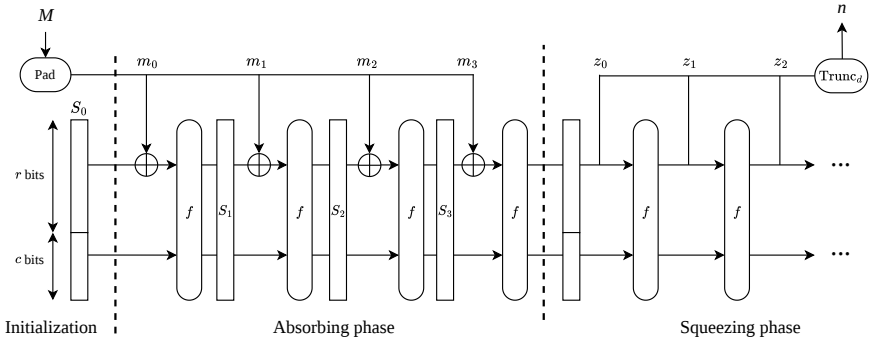


Figure 1: Overview of the standard sponge construction

Feed-forward sponge variants. Recently, several feed-forward sponge variants have been proposed to improve the provable preimage and second-preimage security beyond the classical capacity bound of the standard sponge [SLZ+26, GHJ+26]. These constructions modify the absorbing and/or squeezing procedures by introducing feed-forward operations around the permutation calls. The `Sponge-DM` construction adopts a full-state feed-forward strategy during absorption. For each absorbed block, the entire input state of the permutation is XORed into the permutation output, yielding

$$S_{i+1} = f\big(S_i \oplus (m_i \,\|\, 0^c)\big) \oplus \big(S_i \oplus (m_i \,\|\, 0^c)\big).$$

The overall structure is illustrated in Figure 2.

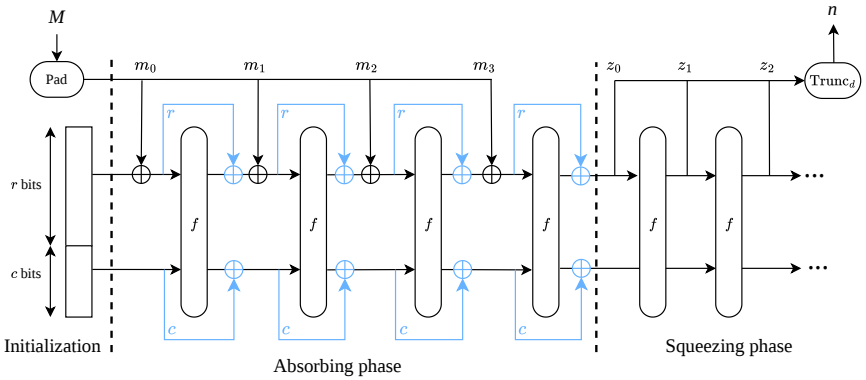


Figure 2: Overview of the `Sponge-DM` construction

Building upon this idea, $\texttt{Sponge-EDM}^c$ mitigates the security degradation for long messages by splitting the permutation into two sub-permutations, denoted as $f^{1/2}$ (defaulting to a balanced $3+3$ round split for 6-round attacks, or explicitly noted for asymmetric splits such as $2+3$ rounds), and feeding forward the c-bit inner state between them.

Concurrently, Guo *et al.* [GHJ+26] proposed $\texttt{Sponge-F}^{\text{P,pd}}$, which applies inner part feed-forward during both the absorbing and squeezing phases. At the final absorption step, a fixed non-zero constant θ is simultaneously XORed into the inner part with the final padded message block. During the squeezing phase, $r' \leq c$ bits are extracted from the inner part of the state at each permutation call, rather than from the outer part. The overall structure is illustrated in Figure 3.

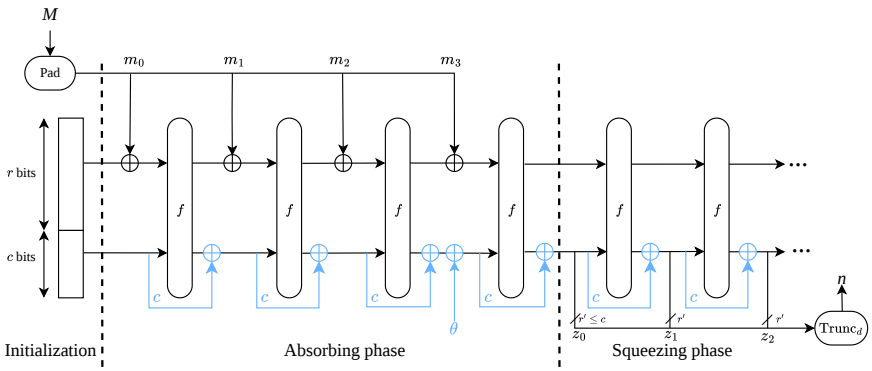
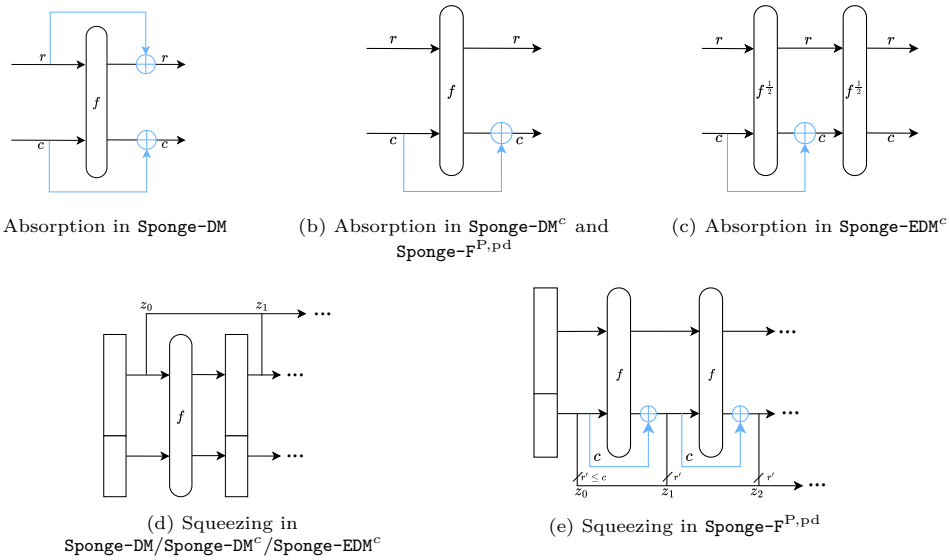


Figure 3: Overview of the $\texttt{Sponge-F}^{\text{P,pd}}$ construction

The absorption and squeezing procedures of these feed-forward variants are summarized in Figure 4. In particular, Sponge-DM and Sponge-EDM^c follow the standard sponge-style outer squeezing strategy, where the full digest can be extracted directly from the outer part of the state. In contrast, Sponge-DM^c typically operates with a narrow squeeze rate to preserve n-bit preimage security, necessitating that the digest be generated iteratively across multiple squeeze calls.



(a) Absorption in Sponge-DM

(b) Absorption in Sponge-DM^c and Sponge-F$^{\mathrm{P,pd}}$

(c) Absorption in Sponge-EDM^c

(d) Squeezing in Sponge-DM/Sponge-DM^c/Sponge-EDM^c

(e) Squeezing in Sponge-F$^{\mathrm{P,pd}}$

Figure 4: The absorption and squeezing of Sponge-DM, Sponge-DM^c, Sponge-EDM^c, and Sponge-F$^{\mathrm{P,pd}}$

Generic security bounds. Table 2 compares the generic collision, preimage, and second-preimage security bounds of the standard sponge and the feed-forward sponge constructions considered in this work.

Table 2: Generic security bounds of sponge and feed-forward sponge constructions

Mode	Output size	Input rate	Security		
			Collision	Preimage	2nd preimage
Sponge	n	r	$\min\left\{\frac{n}{2}, \frac{c}{2}\right\}$	$\min\left\{n, \max\left\{n-r, \frac{c}{2}\right\}\right\}$	$\min\left\{n, \frac{c}{2}\right\}$
Sponge-DM	n	r	$\min\left\{\frac{n}{2}, \frac{c}{2}\right\}$	n	$\min\{n, c-\log_2(l)\}$
Sponge-EDM^c	n	r	$\min\left\{\frac{n}{2}, \frac{c}{2}\right\}$	$\min\{n, \max\{n-r, c\}\}$	$\min\{n, c-\log_2(l)\}$
Sponge-F$^{\mathrm{P,pd}}$	n	r	$\frac{n}{2}$	n	$\min\{n, c-\log_2(l)\}$

Note. l is the number of blocks in the second preimage challenge message. For Sponge-F$^{\mathrm{P,pd}}$, a fixed n-bit output with $n \le c$ is assumed. Sponge-DM^c is omitted since [SLZ+26] does not provide a corresponding generic security bound.

As shown in Table 2, the main provable advantage of the feed-forward constructions lies in their generic preimage and second-preimage resistance. In contrast, all the listed constructions retain the generic collision bound $\min\{n/2, c/2\}$. These bounds are established at the mode level in the ideal-permutation model and do not characterize dedicated attacks exploiting the differential properties of a concrete underlying permutation.

2.2 Instantiation of Feed-Forward Sponge Hash Functions

This paper analyzes the differential collision resistance of feed-forward sponge variants, including `Sponge-DM`, `Sponge-DM`^{c}, `Sponge-EDM`^{c}, and `Sponge-F`$^{\mathrm{P,pd}}$, through a combination of theoretical analysis and concrete attacks on standardized instantiations. The considered instances are chosen to achieve the same generic security levels as the corresponding NIST standard hash functions. Table 3 summarizes all instantiations studied in this work.

NIST standard hash functions. All constructions are instantiated using either the `Keccak` permutation Keccak-p[1600,24] or `Ascon` permutation Ascon-p[320,12]. We compare the differential collision resistance of these feed-forward sponge instances against the NIST standard hash functions `SHA-3` and `Ascon-Hash256`.

The `SHA-3` standard defines a family of hash functions including `SHA3-224`, `SHA3-256`, `SHA3-384`, and `SHA3-512`, as well as extendable-output functions `SHAKE128` and `SHAKE256`. All `SHA-3` instances use Keccak-p[1600,24], which operates on a 1600-bit state represented as $A[x][y][z]$ with $0 \leq x, y \leq 4$ and $0 \leq z < 64$. Each round consists of five mappings $\iota \circ \chi \circ \pi \circ \rho \circ \theta$ detailed below.

- **Linear layer** $\lambda = \pi \circ \rho \circ \theta$: the column-parity mixing step θ, lane rotation ρ, and lane permutation π are all linear over GF(2).
- **Non-linear layer** χ: the only nonlinear operation, defined row-wise as

$$A[x][y] \leftarrow A[x][y] \oplus \big(\neg A[x+1][y]\big) \wedge A[x+2][y], \quad 0 \leq x \leq 4.$$

- **Round constant layer** ι: XORs a round-dependent constant into lane $A[0][0]$, which does not affect differential propagation.

The Ascon-p[320,12] operates on a 320-bit state composed of five 64-bit words $w_0, \ldots, w_4$. Each round applies a transformation $f = p_L \circ p_S \circ p_C$ following an SPN structure consisting of a constant addition layer p_C, a substitution layer p_S, and a linear diffusion layer p_L. The substitution layer consists of 64 parallel applications of a 5-bit Sbox defined by a χ-type function of algebraic degree two. The linear diffusion layer p_L updates each word w_i via a rotation-XOR function Σ_i.

Instantiation of feed-forward sponge hash functions. We next specify the concrete feed-forward sponge instances considered in this work. We follow the feed-forward instantiations designed in [SLZ+26, GHJ+26], which target the intended generic security targets of the corresponding NIST standardized hash functions `SHA-3` and `Ascon-Hash256`. The resulting parameter sets are summarized in Table 3. We therefore focus on these concrete instantiations and investigate their dedicated differential collision resistance.

For instantiations with the `Keccak` permutation Keccak-p[1600,24], we consider four security levels corresponding to `SHA3-224`, `SHA3-256`, `SHA3-384`, and `SHA3-512`. As a NIST standardized hash family, `SHA-3` provides a natural reference baseline for instantiating and comparing feed-forward counterparts at the same target generic security level. For the designed feed-forward instances, we follow the capacity choice $c = n + 64$ adopted in [SLZ+26, GHJ+26]. This choice takes into account the second-preimage security bound in Table 2. Specifically, for the maximum supported message length of $l = 2^{64}$ blocks, maintaining an n-bit second-preimage security level requires $c - \log_2(l) \geq n$, which leads to $c \geq n + 64$.

The resulting parameter sets preserve the intended generic security levels while yielding a larger rate than the corresponding SHA-3 instances. Other rate-capacity choices are possible, but they would define different feed-forward instances. To maintain a meaningful comparison with the corresponding `SHA-3` instances, we therefore

Table 3: Target instantiations and their NIST-matched hashes

Hash Instances	State size	Hash size	r	c	Generic security (bits)		
					Coll.	2nd-pre.	Pre.
`SHA3-224`	1600	224	1152	448	112	224	224
`Keccak-DM`$[288, 224]$	3200	224	1312	288	112	224	224
`Keccak-EDM`$^c[288, 224]$	1888	224	1312	288	112	224	224
`SHA3-256`	1600	256	1088	512	128	256	256
`Keccak-DM`$[320, 256]$	3200	256	1280	320	128	256	256
`Keccak-EDM`$^c[320, 256]$	1920	256	1280	320	128	256	256
`SHA3-384`	1600	384	832	768	192	384	384
`Keccak-DM`$[448, 384]$	3200	384	1152	448	192	384	384
`Keccak-EDM`$^c[448, 384]$	2048	384	1152	448	192	384	384
`SHA3-512`	1600	512	576	1024	256	512	512
`Keccak-DM`$[576, 512]$	3200	512	1024	576	256	512	512
`Keccak-EDM`$^c[576, 512]$	2176	512	1024	576	256	512	512
`Keccak-Sp-F`$_{512}$	2176	512	1024	576	256	512	512
`Ascon-Hash256`	320	256	64	256	128	128	192
`Ascon-DM`	640	256	64	256	128	192	256
`Ascon-Sp-F`	576	256	64	256	128	192	256

Note. Parameters are given in [SLZ+26, GHJ+26] and all values in bits. State size is the permutation width plus the feed-forward part. The `Sponge-DM`^{c} instances are omitted.

focus on the proposed parameter sets that match the same target generic security levels and use the same underlying permutation under the same differential collision framework. This setting allows us to examine how the feed-forward structure and the associated rate-capacity choice affect dedicated differential collision resistance. This yields `Keccak-DM`$[288, 224]$, `Keccak-DM`$[320, 256]$, `Keccak-DM`$[448, 384]$, and `Keccak-DM`$[576, 512]$, the corresponding `Keccak-EDM`$^c[288, 224]$, `Keccak-EDM`$^c[320, 256]$, `Keccak-EDM`$^c[448, 384]$, and `Keccak-EDM`$^c[576, 512]$, and `Keccak-Sp-F`$_{512}$ at the `SHA3-512` security level.

For instantiations with Ascon-p[320,12], we use $(r, c, n) = (64, 256, 256)$, matching the 128-bit collision security level of `Ascon-Hash256`. Instantiating `Sponge-DM` with Ascon-p[320,12] under these parameters gives the instance `Ascon-DM` = `Sponge-DM`$[256, 256]$, while the corresponding `Sponge-F`$^{\mathrm{P,pd}}$ instance is `Ascon-Sp-F`. The two variants share the same parameters but differ primarily in their squeezing procedures. `Ascon-DM` extracts the digest at a 64-bit squeeze rate. Conversely, `Ascon-Sp-F` extracts the entire 256-bit digest from the inner part in a single call.

2.3 Differential Collision Attacks on Sponge Hashes

We adopt the differential notation for Keccak-p[1600,24]. A dedicated differential collision attack on a permutation-based sponge hash generally consists of two phases, namely, the *colliding* phase and the *connecting* phase. In the colliding phase, the adversary constructs a high-probability differential trail over the reduced-round permutation such that the resulting output difference satisfies the collision condition. For sponge hashes, this typically requires the output difference on the squeezing-relevant part of the state to be zero. Following the conventional notation, let β_i denote the input difference to the χ layer of round i, and let α_i denote the corresponding output difference. For the Keccak-p[1600,24]

used in SHA-3, one round propagates as

$$\alpha_i \xrightarrow{\lambda} \beta_i \xrightarrow{\chi} \alpha_{i+1},$$

where $\lambda = \pi \circ \rho \circ \theta$ denotes the linear layer. An r-round differential trail is therefore written as

$$\alpha_0 \xrightarrow{\lambda} \beta_0 \xrightarrow{\chi} \alpha_1 \xrightarrow{\lambda} \beta_1 \xrightarrow{\chi} \cdots \xrightarrow{\lambda} \alpha_{r-1} \xrightarrow{\chi} \beta_{r-1}.$$

The differential probability is commonly represented by its propagation weight $w = -\log_2 p$, where p denotes the propagation probability.

For sponge hashes, however, a colliding trail alone is not enough. Besides the output collision condition, the attack must also meet difference constraints on the capacity part during absorption, leading to the CICO problem. To overcome this obstacle, the adversary extends the colliding trail backward to construct a connecting trail satisfying the required capacity conditions, *i.e.*, the connecting phase. This phase exploits the cryptanalytic properties to construct message structures, commonly referred to as connectors. A connector uses the available Degrees of Freedom (DoF) of the message blocks to generate a subspace of message pairs such that every pair in the subspace follows the connecting trail deterministically. The remaining unconstrained message space is then exhaustively searched to identify message pairs that also satisfy the colliding trail, thereby yielding a collision.

In practice, both stages become increasingly difficult as the permutation width grows. High-probability colliding trails are sparse in the large permutation state space, while connector construction must satisfy the capacity constraints without exhausting too many DoF. The primary method for constructing such connectors is the TDA, which exploits the low algebraic degree of χ to reduce the search for valid message pairs to solving linear systems. Subsequent work extended this approach to multi-round connectors, and SAT-based methods further improved DoF allocation.

2.4 Quantum Collision Attack Model

Quantum collision attacks can be evaluated under different resource models. Following the classification commonly used in quantum collision cryptanalysis, we briefly distinguish three representative settings. In the first setting, exponentially large quantum random-access memory (qRAM) is available, under which the BHT algorithm [BHT98] finds a collision for an n-bit ideal hash function in $O(2^{n/3})$ time using $O(2^{n/3})$ qRAM. In the second setting, we evaluate the efficiency of quantum algorithms in terms of their time–space tradeoff, following the framework discussed in [HS21]. Let S denote the hardware size, *i.e.*, the maximum size of the quantum computer (equivalently, the width of the quantum circuit) and the classical computer. Under this constraint, no existing quantum collision attack on a random function achieves a time complexity better than

$$T = \frac{2^{n/2}}{S}. \tag{1}$$

In the third setting, only polynomial-size quantum memory is assumed, while exponentially large classical memory is allowed. These settings rely on different assumptions on quantum resource and therefore represent different resource models rather than directly comparable complexity bounds.

In this work, we adopt the second setting, following the dedicated quantum collision analyses of [HS21] and [GLST22]. Thus, $2^{n/2}/S$ serves as the generic baseline throughout this paper and is not intended to represent a generic bound applicable to other quantum-computing models.

For the dedicated attacks, we follow the SHA-3 framework of [GLST22]. The connector is constructed classically to generate message pairs satisfying the connecting constraints,

while Grover's algorithm is applied only to the remaining exhaustive-search phase. Let p_{coll} denote the classical success probability of a candidate after the connector has been fixed. Let T_A and S_A denote, respectively, the depth and width of the quantum circuit used to test one candidate. With a total hardware resource S, parallel Grover search gives

$$T_q = T_A \frac{\pi}{4} \sqrt{\frac{S_A}{p_{\text{coll}} S}}. \tag{2}$$

In the attacks considered below, connector construction is treated as a classical preprocessing step, and the quantum cost is dominated by the Grover oracle implementing the corresponding reduced-round one-block hash evaluation.

3 Differential Collision Resistance of Feed-Forward Sponge Variants

In this section, we evaluate the differential collision resistance of the instantiated feed-forward sponge variants. We first review the outer and inner differential collision frameworks for standard sponge hashes, based on which we then examine how the feed-forward mechanism affects dedicated collision attacks. Specifically, we discuss the feed-forward impact on the differential trail conditions and the available DoF in connector construction.

3.1 Differential Collision Framework for Sponge Hashes

In dedicated cryptanalysis of sponge hash functions, including preimage or collision attacks, a specialized *attack framework*, which links the cryptanalytic goal to the input constraints imposed by the sponge construction, is required to overcome the CICO obstacle. For differential collision attacks, the concrete form of the attack framework depends on where the collision condition is imposed, which in turn is influenced by the *squeeze rate* and the squeezing procedure.

Outer collision framework. An *outer collision* refers to the case where the zero output difference for a hash collision is imposed on the outer part of the state used for digest extraction. This setting naturally arises when the digest is extracted in a single squeeze whose rate equals the digest length, while the digest length is still smaller than the remaining inner part. The `SHA-3` family is a typical example of this case.

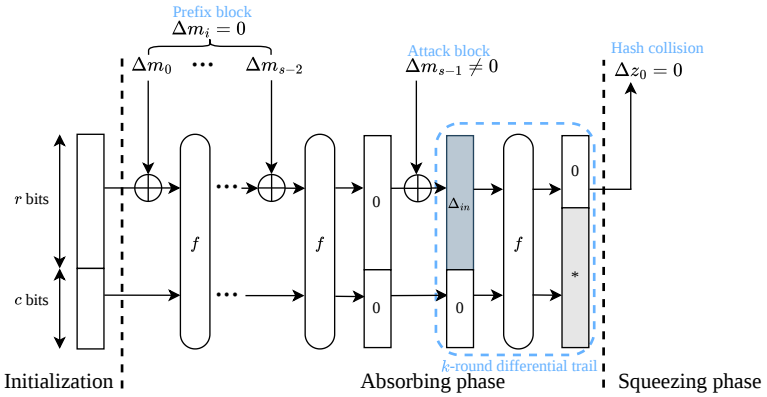


Figure 5: The outer collision framework for standard sponge hashes

Figure 5 illustrates the corresponding *outer collision framework*. The attack is performed on the final absorbed message block, called the *attack block*, where the input difference

has zero inner part and non-zero outer part injected through the message block. After absorption, a single squeeze extracts the digest. The framework therefore requires zero difference on the squeezed output, while the differences on the remaining state bits are irrelevant. At the attack block, the connecting and colliding trails, together with the construction of connectors, are as described in Section 2.3. A collision is obtained whenever a message pair follows the differential trail in the attack block with a computational cost below the birthday bound.

In practice, the DoF available at the attack block may be insufficient to fulfill all connecting conditions or to sustain the exhaustive search stage. In such cases, *prefix blocks* are prepended to provide additional DoF. Without loss of generality, we assume these prefix blocks introduce no difference in the inner part. [3] Since the adversary cannot effectively control the differential propagation across multiple preceding blocks, any non-zero inner difference they might introduce cannot be constructively exploited, rendering it analytically equivalent to a zero inner difference at the input of the attack block.

Inner collision framework. Depending on the rate–capacity ratio and the squeezing procedure, the outer collision may not always be the most suitable choice. Two representative situations arise. First, if the squeeze rate exceeds the capacity, enforcing a collision on the outer part becomes structurally more constrained than targeting a zero difference on the inner part. Second, if the squeeze rate is small and the digest is extracted through multiple squeeze calls, *e.g.*, in `Ascon-Hash256` each squeeze outputs 64 bits and four calls are required to obtain a 256-bit digest, the outer collision framework is no longer directly applicable, as satisfying only partial digest constraints per squeeze does not naturally compose into a full collision.

In such settings, an inner collision provides a natural alternative and is a standard approach in the analysis of sponge constructions [BDPV11]. The adversary first aims to obtain a zero difference on the inner part of the state and subsequently converts it into a full-state collision, leading to the *inner collision framework* illustrated in Figure 6.

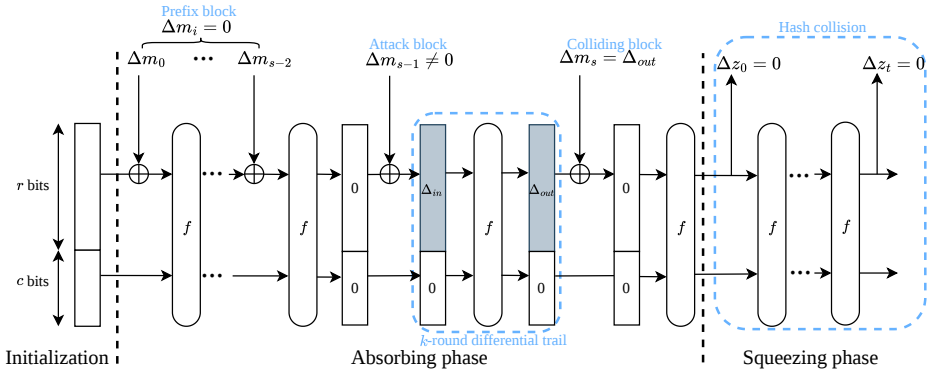


Figure 6: The inner collision framework for standard sponge hashes

Here, the *attack block* is placed at the second-to-last absorbed message block. Since an inner collision at this block does not immediately guarantee a hash collision at the

[3] Alternatively, one may probabilistically search over prefix blocks to obtain a favorable non-zero inner difference before the attack block, as considered in [HBDM22]. We considered this possibility, but such attacks are still case-specific and not yet well explored. Existing examples mainly sieve *prefix blocks* by exhaustive search to obtain a convenient inner difference. This can be viewed as a separate variant of the attack framework, while the resulting inner difference is used at the attack block essentially in the same way as in the zero-difference case. Thus, in our analysis, $\Delta_c^{in} = 0^c$ is a representative modeling choice.

squeeze interface, the subsequent block (the *colliding block*) is utilized. At the colliding block, the inner input difference is already zero. The adversary can then freely choose the message difference to perfectly cancel the non-zero outer difference propagating from the attack block. This cancellation yields an all-zero difference across the full state (a *state collision*). Once a state collision is achieved, the difference remains zero throughout the entire squeezing phase.

3.2 Feed-Forward Effects on Outer Collision Attacks

We evaluate the differential collision resistance of feed-forward sponge hash functions under the outer collision framework. The feed-forward sponge variants are introduced in Section 2.1, and their concrete instantiations are listed in Section 2.2. Among the instantiated targets, `Sponge-DM`, `Sponge-EDM`^{c}, and `Sponge-F`$^{\mathrm{P,pd}}$ can be analyzed directly within the outer collision framework shown in Figure 7. The `Sponge-DM`^{c} construction will be discussed in Section 3.3 as its digest is obtained through multiple squeeze calls. A special case is `Keccak-Sp-F`$_{512}$, which is based on `Sponge-F`$^{\mathrm{P,pd}}$. Although the digest is extracted from the trailing part and may therefore appear to be an inner part extraction, the entire digest is still obtained in a single squeeze. Consequently, `Keccak-Sp-F`$_{512}$ belongs to the outer collision framework.

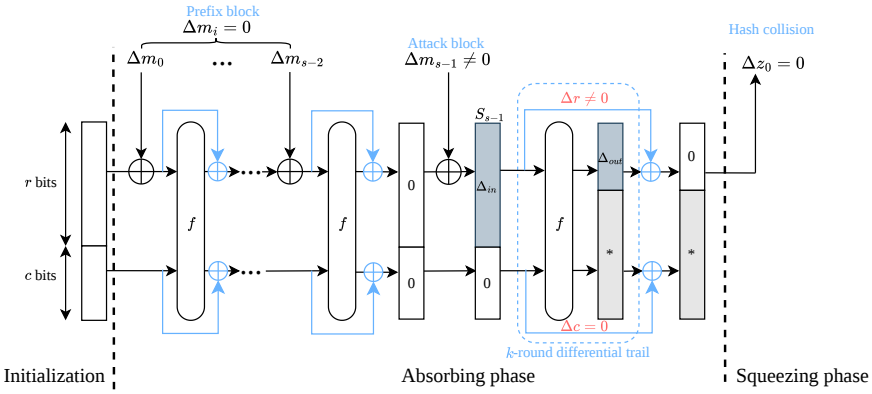


Figure 7: The outer collision framework for feed-forward sponge variants

In what follows, we focus on how the feed-forward operation changes the cryptanalysis at the *attack block*. Two components of the outer collision attack may be affected, *i.e.*, the *differential collision trail* (including the connecting trail and the colliding trail), and the *connector construction* used to realize the connecting trail.

The feed-forward effect on the differential collision trail. In feed-forward sponge design, the feedback operation may affect either the full state or only part of the state. For example, full-state feed-forward during absorption is used in `Sponge-DM`, whereas `Sponge-DM`^{c}, `Sponge-EDM`^{c}, and `Sponge-F`$^{\mathrm{P,pd}}$ apply feed-forward only to the capacity (inner) part. Since these two cases have fundamentally different effects on collision attacks, we distinguish between feed-forward that affects the outer part and that is confined to the inner part.

When *the feed-forward affects the outer part*, the collision condition at the attack block changes directly. As depicted in Figure 7, a hash collision requires the outer input difference and the outer output difference at the attack block to be identical. In a standard sponge function under the same framework, the colliding trail is conventionally formalized by imposing a zero outer output difference. With feed-forward, however, the outer part

of the input difference also becomes part of the collision condition. At first sight, this modified condition may appear more restrictive than the standard one. However, such an appearance alone does not imply that collision differential trails become harder to find. Replacing the zero-output condition by an input-output matching condition does not necessarily reduce the set of candidate trails. In fact, depending on the instance, it may preserve the same set of solutions or even introduce additional alternatives. Consequently, the effective search space for differential trail finding need not become smaller.

Specifically, from the viewpoint of difference propagation, feed-forward on the outer part simply replaces the requirement of a zero outer output difference by requiring that the outer part of input and output differences coincide along the differential collision trail. At this level of generality, it is therefore impossible to determine whether the resulting dedicated collision attack becomes easier or harder than on the corresponding standard sponge construction. What can be concluded is only that feed-forward does not automatically strengthen resistance against dedicated differential collision attacks, despite its original motivation of improving provable preimage security.

Conversely, when *the feed-forward operation is confined to the capacity (inner) part*, the differential trail at the attack block remains invariant. In the outer collision framework, the attack block is chosen such that its input difference has a non-zero outer part injected through the message block and a zero inner part. Consequently, the inner input difference remains zero throughout the attack block. If feed-forward affects only the inner part, it merely XORs this zero inner difference back into the state and therefore does not alter any of the differential conditions imposed at the attack block. Accordingly, capacity only feed-forward leaves the differential collision trail unchanged under the outer collision framework.

In summary, full-state feed-forward, as used in `Sponge-DM`, modifies the trail conditions at the attack block, although the resulting cryptanalytic impact remains instance dependent and cannot be determined in general. In contrast, for constructions employing capacity only feed-forward, including `Sponge-EDM`^{c} and `Sponge-F`$^{\mathrm{P,pd}}$, the differential collision trail is unaffected in any nontrivial way. The same conclusion will also apply to `Sponge-DM`^{c} under the inner collision framework discussed in Section 3.3, as the relevant inner input difference at the attack block is likewise zero in that setting.

The feed-forward effect on the connector construction. At first sight, feed-forward mainly modifies the differential conditions at the attack block, while leaving the message blocks and their structure unchanged. However, this view does not fully capture its impact on the overall attack, since feed-forward also changes certain design parameters of the resulting hash function, as reflected in Table 3 in Section 2.2. In particular, several feed-forward `Keccak-DM` instances that achieve a comparable generic security level to a `SHA-3` reference exhibit different rate and capacity parameters. A common consequence is that the rate-capacity ratio of the feed-forward construction is larger than that of the corresponding `SHA-3` instance. Since a larger rate directly increases the amount of message DoF available, this modification has a direct impact on the connector construction phase.

In practice, connector construction is highly sensitive to the available message DoF. Even when the differential trail conditions are only mildly affected, an increase in DoF can significantly change the feasibility of constructing valid connectors. In this sense, the main influence of feed-forward on outer collision attacks does not necessarily arise from the modification of the differential trail itself, but rather from the induced change in the parameter configuration of the construction.

On the other hand, although the increased number of message DoF typically facilitates the construction of connectors and may reduce the practical complexity of the attack, this effect is instance dependent, as the final complexity still depends on the underlying permutation, the specific differential trail, and the precise distribution of DoF. Therefore,

we do not give a universal quantitative statement on the feed-forward effect. In practical design, this connector sensitivity should be examined carefully when aiming for strong differential collision resistance.

Discussion. Taken together, the outer collision analysis indicates that feed-forward sponge constructions do not provide a clear strengthening of resistance against this class of dedicated differential collision attacks. In some cases, the dedicated collision behavior may even appear weaker than that of the corresponding standard sponge construction. Nevertheless, dedicated collision attacks remain highly instance dependent. The final attack complexity is determined by multiple parameters, including the *state size*, the *rate-capacity ratio*, and the *digest length*. For sponge and sponge-like hash functions, these parameters and design choices should therefore be selected carefully in order to achieve an appropriate trade-off between security and performance.

3.3 Inner Collision Attacks on Feed-Forward Sponge Variants

We now examine the feed-forward sponge variants under the inner collision framework. Among the targets listed in Section 2.2, this framework specifically applies to `Ascon-DM` and the `Sponge-DM`^{c} instances whose digests require multiple squeeze calls. Following the outer collision case, we decompose the analysis into two components, *i.e.*, the differential collision trail and the connector construction. Figure 8 illustrates the inner collision framework adapted for feed-forward variants. In this setting, the feed-forward operation generally preserves the differential trail and has a highly constrained, instance dependent impact on the connector phase.

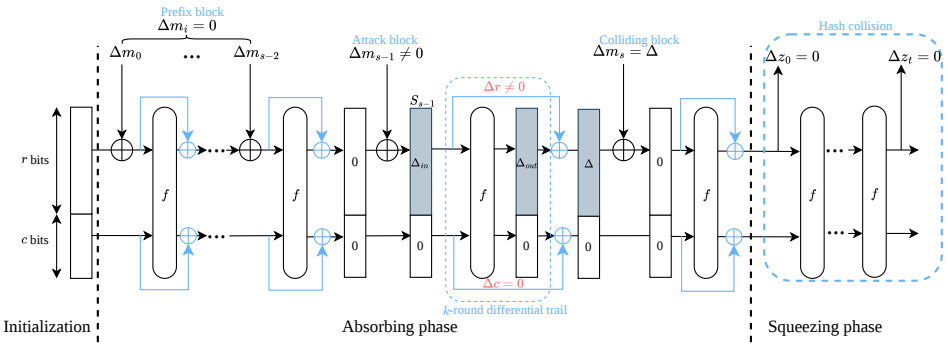


Figure 8: The inner collision framework for feed-forward sponge variants

The feed-forward effect on the differential collision trail. Under the inner collision framework, the adversary targets a zero difference in the capacity part (*i.e.*, $\Delta c = 0$) at the output of the attack block. We consider the case where the input to the attack block has a zero inner difference, while a non-zero outer difference is injected via the message. In a full-state feed-forward construction like `Sponge-DM`, the feed-forward mechanism XORs the input difference and the output difference of the permutation. Since the inner input difference is strictly zero, this XOR operation naturally preserves the inner collision condition on the capacity part. The remaining non-zero difference in the rate part is subsequently canceled at the colliding block, identical to the standard sponge setting. Thus, inner collision trails designed for the standard sponge can be directly reused without modification. For constructions utilizing capacity only feed-forward (*e.g.*, `Sponge-DM`^{c}), the relevant inner input difference at the attack block is similarly zero, meaning the feedback does not affect the trail.

The feed-forward effect on the connector construction. For `Ascon-DM`, the rate and capacity parameters are identical to those of `Ascon-Hash256`. As a result, the amount of available message DoF at the attack block remains unchanged, and the connector construction is identical to that of `Ascon-Hash256`, except for the modified state update in the absorbing phase. In this case, the feed-forward operation does not introduce any additional connector effect beyond what already follows from the unchanged rate and capacity parameters. For `Sponge-DM`^{c} instances, the rate and capacity parameters differ from those of the corresponding `SHA-3` reference, and the available message DoF may increase. The resulting impact on connector construction is analogous to the outer collision analysis in Section 3.2, and we do not repeat the details here.

Overall, feed-forward sponge constructions behave similarly to their standard sponge counterparts in the inner collision setting. The differential trail at the attack block is preserved, and for `Ascon-DM` even the connector construction remains unchanged at the parameter level. For `Sponge-DM`^{c} instances, the only notable effect is the change in parameters affecting the connector phase. In summary, compared with outer collision attacks, the feed-forward mechanism has a much smaller impact in the inner collision framework, and its effect is more straightforward to characterize.

4 Collision Attacks on Feed-Forward Sponge Instances

In the previous section, we analyzed how feed-forward affects differential collision attacks against these feed-forward sponge hash functions. The main impact appears in the outer collision setting, where feed-forward may alter the collision conditions and, in some cases, lead to weaker collision resistance than in the corresponding standard sponge constructions. In the inner collision setting, feed-forward typically preserves the differential trail and has only a limited effect on connector construction.

In this section, we confirm these observations through dedicated collision attacks on the instances listed in Table 3. The primary focus is on outer collision attacks against the feed-forward constructions `Sponge-DM`, `Sponge-EDM`^{c}, and `Sponge-F`$^{\mathrm{P,pd}}$ instantiated with Keccak-p[1600,24]. We compare these results with those of the corresponding `SHA-3` instances at the same generic security level. We describe the collision construction at the attack block in Section 4.1 and then present the attack results grouped by digest length. Finally, we briefly discuss inner collision attacks on `Ascon-DM` and `Ascon-Sp-F`, which share (r, c, n) with `Ascon-Hash256`.

4.1 Constructing Outer Collisions at the Attack Block

Following the outer collision framework, we adopt the two-phase strategy of [GLL+20] to construct collisions at the attack block, as illustrated in Figure 9. The overall procedure is divided into the colliding phase and the connecting phase, which were introduced briefly in Section 2.3. We now explain the specific methods used for trail search and connector construction in our experiments.

Colliding phase. The colliding phase searches for an n_{r_2}-round differential trail from an intermediate difference ΔS_I to an output difference ΔS_O satisfying the feed-forward collision condition. As discussed in Section 3, the standard zero-output condition is replaced by the equality constraint

$$\Delta S_O^n = \Delta \overline{M}^n,$$

where $\Delta\overline{M}$ is the input difference of the connecting phase. In the dedicated attacks in this section, we employ two strategies to satisfy this constraint.

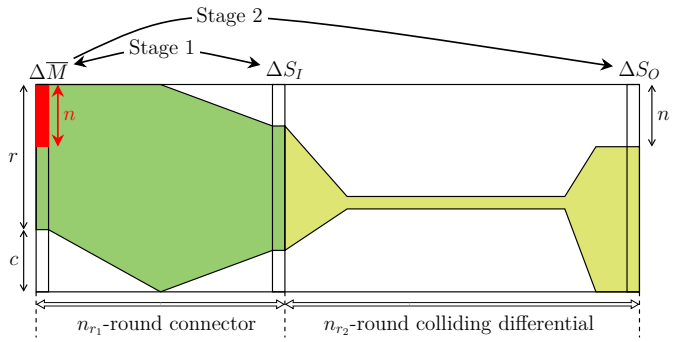


Figure 9: Two-phase collision construction at the attack block taken from [GLL+20]

The first strategy imposes zero difference on the feed-forward affected output part, *i.e.*, $\Delta S_O^n = \Delta\overline{M}^n = 0^n$ for `Sponge-DM`. For the instances considered under this strategy, $n < r$. This reduces the collision condition back to that of the corresponding `SHA-3` instance and allows the direct reuse of existing `SHA-3` collision trails. We apply this approach to the attacks on `Sponge-EDM`^{c} and to instances with fewer attacked rounds, such as the 5-round `Keccak-DM`$[320, 256]$ where we reuse the known colliding trails of `SHA3-256`.

The second strategy utilizes SAT-based automated tools to search for trails satisfying either $\Delta S_O^n = 0^n$ or the generalized condition $\Delta S_O^n = \Delta\overline{M}^n$. We apply this method when suitable `SHA-3` trails are not reported yet or the all-zero condition is too restrictive. In the practical search for a colliding trail with the $\Delta S_O^n = \Delta\overline{M}^n$ condition, we place no restrictions on the outer difference ΔS_O^n and only consider the propagation probability. This generally leads to higher-probability trails because a strict zero outer output difference is no longer required. Simultaneously, the $\Delta S_O^n = \Delta\overline{M}^n$ condition must be incorporated into the connecting trail. For the feed-forward instances considered here, the number of connecting conditions increases only slightly, from roughly $2n$ in `SHA-3` to about $2n + 64$. The overall difficulty of constructing the connecting trail therefore remains comparable.

During our search experiments, we observed that finding high-probability trails with a strict all-zero outer output difference is practically difficult for the `SHA-3` and `Sponge-EDM`^{c} instances. Targeting the generalized equality constraint instead enables the automatic search models to identify valid trails with relatively high probabilities much more effectively. Nevertheless, this operational flexibility does not bypass the inherent limitations of the `Keccak` permutation. Extending the colliding trails to 5 rounds while maintaining a probability high enough for a valid attack remains infeasible in our experiments, which aligns with the known differential propagation limitations in standard `SHA-3`.

Connecting phase. The connecting phase aims to construct a connector that generates a subspace of message pairs (M, M') propagating $\Delta\overline{M}$ to ΔS_I. Depending on the instance size and the available difference level DoF, we apply either the TDA method [DDS12] or a SAT-based connector construction [GLST22].

We apply the TDA to `Keccak-DM`$[288, 224]$ and `Keccak-EDM`$^c[288, 224]$. For these smaller targets, the difference level DoF is sufficient to list a solvable system of difference equations. For larger instances, however, the difference level DoF becomes insufficient. For example, in TDA, each active χ-Sbox contributes 2 difference level DoF. For `Keccak-DM`$[320, 256]$, the number of conditions on $\Delta\overline{M}$ is approximately 580 (comprising $320 + 256 + 4$, including 4 conditions from the `01` domain separator and `11` padding bits). Even assuming all first-round Sboxes are active to yield at most $320 \times 2 = 640$ DoF, TDA fails to produce a valid 2-round connector in practice. For `Keccak-DM`$[448, 384]$, the condition count increases to roughly 836, which strictly exceeds the theoretical difference level DoF limit. Consequently,

we switch to the SAT-based connector construction for these larger instances, which efficiently handles the search for $\Delta\overline{M}$ under the additional feed-forward constraints.

Structurally, the larger rate in Sponge-DM, Sponge-EDM^c, and Keccak-Sp-F$_{512}$ provide more message DoF than their SHA-3 counterparts. In our actual attack implementations, this parameter shift noticeably facilitates the construction of 1-round and 2-round connectors and yields larger valid message subspaces than those in standard SHA-3. However, this structural convenience is practically limited. The complex interplay among the specific collision trail, the modified collision condition, and the rate-capacity ratio prevents us from effectively utilizing the message DoF to extend the connectors to 3 rounds.

4.2 Collision Attacks on 224-bit Feed-Forward Instances

The best known classical collisions on SHA3-224 and SHA3-256 are the practical 5-round attacks presented in [GLL+20]. We use these attacks as the baseline for the feed-forward instances. For the classical comparison, we focus on Sponge-DM, since the Sponge-EDM^c setting is more favorable to the adversary than standard SHA-3 under the attack framework considered here. Since Keccak-DM$[288, 224]$ represents the easier Sponge-DM target, the discussion of the practical 5-round search is deferred to Keccak-DM$[320, 256]$ in Section 4.3. Instead, we report a 6-round classical collision on Keccak-DM$[288, 224]$, for which no corresponding classical attack on SHA3-224 has been reported, and a 6-round quantum collision on Keccak-EDM$^c[288, 224]$.

6-round collision attack on Keccak-DM$[288, 224]$. We mount a 6-round classical collision attack on Keccak-DM$[288, 224]$ (the Sponge-DM counterpart of SHA3-224) using the differential trail listed in Table 6 (Supplementary Material Section A.1). The attack follows the two-phase outer collision framework formalized in Section 4.1, utilizing a 2-round connector. Crucially, the relaxed outer collision condition and the additional message DoF provided by Keccak-DM$[288, 224]$ make this 6-round classical attack feasible, contrasting sharply with the current security status of SHA3-224.

On average, the connector yields a subspace of approximately 2^{27} message pairs that deterministically follow the connecting trail. As detailed in Table 6, the subsequent 4-round colliding trail has a total probability of 2^{-68}. Consequently, about $2^{68-27} = 2^{41}$ distinct connectors are required to identify a message pair satisfying both phases. Since the computational cost of the connector construction is relatively cheap, it is negligible in the overall estimate. The total classical time complexity is therefore $T_{\text{cl}}^{(224)} = 2^{68}$ evaluations of the 6-round Keccak-p[1600,24] permutation. A verified message pair extracted from the connector subspace is provided in Table 7 (Supplementary Material Section A.1).

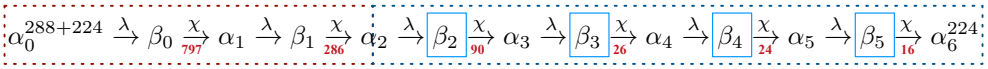
6-round quantum collision attack on Keccak-EDM$^c[288, 224]$. Let $X = S \oplus (M \parallel 0^c)$. At the attack block, Keccak-EDM$^c[288, 224]$ updates the state as

$$S \leftarrow f^{1/2}\big(\, f^{1/2}(X) \oplus (0^r \parallel \lfloor X \rfloor_c)\,\big),$$

where the two sequential $f^{1/2}$ calls are 3-round truncations of Keccak-p[1600,24] (see Figure 4c). Therefore, our 6-round attack targets the full 3+3 design split, rather than a single 6-round permutation call as in SHA-3. The differential attack adheres to the outer collision framework and adopts the quantum collision model of [GLST22] on a single absorbing block.

The complete trail is depicted in Figure 10 and detailed in Table 4 (Supplementary Material Section A.1). In the figure, the red-dotted frame indicates the 2-round *connecting* trail (realized by a connector), the blue-framed states β_i constitute the 4-round *colliding* trail, and the red numbers denote the log-weights of the differential propagation. Notably,

the capacity feed-forward does not alter the connecting constraints or the connector construction, nor does it increase the quantum circuit width under the model of [GLST22].



Note. Here and in subsequent figures, connecting-trail probabilities are shown only to indicate connector difficulty.

Figure 10: The collision trail for the 6-round quantum attack on Keccak-EDM^c[288, 224]

In the 6-round quantum collision on SHA3-224, the 2-round connector leaves 28 difference conditions to be resolved by brute force. In contrast, the 2-round connector for Keccak-EDM^c[288, 224] perfectly satisfies all connecting constraints, yielding a message pair subspace of approximately 2^{45}. While advantageous, this subspace is still insufficient to extend the connector to three rounds. Thus, the remaining four rounds are handled via exhaustive search. Along this trail, the colliding differential has a propagation probability of $2^{-(90+61.4)}$ (a concrete message pair fulfilling the connecting trail is given in Table 5 in Supplementary Material Section A.1). Since this probability is far below the classical birthday bound of 2^{-112}, finding a collision classically is out of reach.

Quantum complexity. Generating $2^{151.4-45} = 2^{106.4}$ independent connector instances covers the required classical search space, yielding an overall classical work of $2^{151.4}$ permutation evaluations on this quantum-oriented trail. Following the framework of [GLST22], the connector construction is treated as a classical precomputation with a negligible quantum depth. Subsequently, Grover's algorithm is deployed to search for a collision among the remaining DoF by evaluating the *one-block* Keccak-EDM^c[288, 224] operation (which encapsulates the two $f^{1/2}$ evaluations and the intermediate capacity feed-forward). Given a hardware resource S, the quantum time complexity is estimated as

$$T_{\mathrm{q}}^{(224)} = \frac{\pi}{4} \cdot \frac{\sqrt{C_{224} \cdot 2^{151.4}}}{\sqrt{S}} \approx \frac{2^{80.0}}{\sqrt{S}},$$

where $C_{224} = \frac{286+1600\cdot 1601+4000}{4000}$ represents the circuit cost factor for the one-block evaluation. This metric uses the same normalization as [GLST22], adapted to the Sponge-EDM^c absorption. While the linear capacity feed-forward introduces negligible Toffoli depth, it requires an additional 288×2 qubits to temporarily store and uncompute the inner state. Consequently, the baseline quantum circuit width increases from 3424 ($1600 \times 2 + 224$) to 4000, which is rigorously accounted for in the resource evaluation. For comparison, the corresponding 6-round quantum collision on SHA3-224 has a cost of approximately $2^{97.75}/\sqrt{S}$. The improvement observed on Keccak-EDM^c[288, 224] is driven fundamentally by the smaller classical exponent (151.4), which is a direct consequence of the enhanced trail and connector construction.

4.3 Collision Attacks on 256-bit Feed-Forward Instances

This subsection presents the practical 5-round baseline attack on Keccak-DM[320, 256], followed by a 6-round classical attack on Keccak-DM[320, 256] and a 6-round quantum attack on Keccak-EDM^c[320, 256].

5-round collision attack on Keccak-DM[320, 256]. The attack reuses the colliding trail from the practical 5-round collision attacks on SHA3-224 and SHA3-256 reported in [GLL+20], which is detailed in Table 8 in Supplementary Material Section A.2. Figure 11 shows the complete attack trail. The blue framed states β_i correspond to the colliding phase, while the red dotted frame highlights the connecting phase. The red values below each χ operation indicate the differential propagation weights.

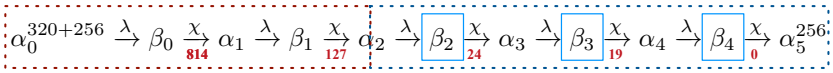


Figure 11: The collision trail for the 5-round attack on Keccak-DM[320, 256]

As in the attack on SHA3-256, the overall complexity is fully practical and a concrete collision message pair is provided in Table 9 in Supplementary Material Section A.2. The main difference lies in the connector construction. Owing to the larger rate of Keccak-DM[320, 256], the message DoF available during the connecting phase increases substantially, resulting in a solution subspace of approximately 2^{199} message pairs. By comparison, the corresponding connector for SHA3-256 produces a subspace of only about 2^{60} message pairs.

6-round collision attack on Keccak-DM[320, 256]. We further extend the attack to 6 rounds on Keccak-DM[320, 256], the full-state feed-forward counterpart of SHA3-256. To the best of our knowledge, no classical 6-round collision attack has previously been reported for this target. The attack follows the outer collision framework introduced in Section 4.1 and employs the differential trail listed in Table 12 in Supplementary Material Section A.2. The 2-round connector generates a message-pair subspace of approximately 2^{40} solutions. As shown in Table 12, the subsequent 4-round colliding trail has a total propagation probability of 2^{-75}.

Combining the two phases requires approximately $2^{75-40} = 2^{35}$ independent connector instances before a complete collision is expected. Since the connector construction contributes only a negligible fraction of the overall cost, the attack complexity is dominated by the exhaustive search phase and amounts to $T_{\text{cl}}^{(256)} = 2^{75}$ evaluations of the 6-round Keccak-p[1600,24]. A verified message pair satisfying the connecting trail is provided in Table 13 in Supplementary Material Section A.2.

6-round quantum collision attack on Keccak-EDM^c[320, 256]. We next apply the quantum outer collision framework to Keccak-EDM^c[320, 256]. As in the 224-bit instance, the attack targets the complete 3+3 absorption structure rather than a single 6-round permutation. The colliding trail is taken directly from the 6-round quantum collision attack on SHA3-256 presented in [GLST22]. Figure 12 depicts the full trail and Table 10 in Supplementary Material Section A.2 provides the detailed differential transitions.

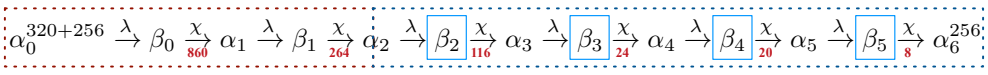


Figure 12: The collision trail for the 6-round quantum attack on Keccak-EDM^c[320, 256]

The 2-round connector satisfies all connecting constraints and produces a message pair subspace of approximately 2^{60} solutions. Although considerably larger than in the SHA3-256 case, this subspace remains insufficient to extend the connector to three rounds. A concrete message pair satisfying the connecting trail is given in Table 11 in Supplementary Material Section A.2. The remaining four rounds therefore are handled through exhaustive search. The 4-round colliding differential has a propagation probability of $2^{-(116+52)}$. This value is significantly below the classical birthday bound of 2^{-128}, rendering a classical collision search infeasible. Covering the corresponding search space requires approximately $2^{168-60} = 2^{108}$ independent connectors, leading to an overall classical workload of 2^{168} permutation evaluations on this trail.

The quantum search proceeds exactly as in the 224-bit attack. Applying Grover's

algorithm to the remaining search space gives the quantum complexity

$$T_{\mathrm{q}}^{(256)} = \frac{\pi}{4} \cdot \frac{\sqrt{\frac{264+1600\cdot 1601+4096}{4096} \cdot 2^{168}}}{\sqrt{S}} \approx \frac{2^{88.3}}{\sqrt{S}} .$$

The factor inside the square root corresponds to the circuit cost of a one-block evaluation. For comparison, the corresponding attack on SHA3-256 has complexity approximately $2^{104.25}/\sqrt{S}$.

4.4 Collision Attacks on 384-bit Feed-Forward Instances

For the 384-bit instances, the enlarged rate substantially strengthens the outer collision framework and enables practical attacks beyond what is currently known for SHA3-384. We obtain practical collisions on both 4 and 5 rounds, whereas the best known 4-round collision attack on SHA3-384 has complexity $2^{59.65}$ and the existing 5-round attack remains far from practical.

4-round collision attack on Keccak-DM[448, 384]. The 3-round colliding trail utilized for this attack is listed in Table 14 (Supplementary Material Section A.3) alongside a 1-round connector. Figure 13 illustrates the full connecting and colliding trail. For comparison, the best known 4-round collision attack on SHA3-384 requires a complexity of $2^{56.11}$ [CHCW26] and no concrete collision message pair has been reported. Conversely, this 4-round attack on Keccak-DM[448, 384] is fully practical. A collision message pair is provided in Table 15 (Supplementary Material Section A.3).

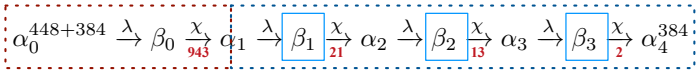


Figure 13: The collision trail for the 4-round practical attack on Keccak-DM[448, 384]

5-round collision attack on Keccak-DM[448, 384]. The 5-round collision attack on Keccak-DM[448, 384] combines a 2-round connector with a 3-round colliding trail, detailed in Table 16 in Supplementary Material Section A.3. The colliding phase has a propagation probability of approximately 2^{-48}, while the connector produces a solution subspace of size about 2^{55}. Consequently, the expected collision search falls well within practical complexity. A collision message pair is provided in Table 17 in Supplementary Material Section A.3.

Similar to the 6-round attacks on Keccak-DM[288, 224] and Keccak-DM[320, 256], this attack also benefits from two favorable properties of the feed-forward construction. First, the relaxed collision condition enables differential trails that lead to valid attacks. For comparison, when we apply the same backward extension to the colliding trails currently used against SHA3-384, we are unable to obtain a connecting trail that can be converted into a valid connector. Among all colliding trails examined in our experiments, none resulted in a feasible connector construction. Second, the enlarged rate provides substantially more message DoF during the connecting phase than in SHA3-384. For comparison, the best known 5-round outer collision attack on SHA3-384 has complexity $2^{170.73}$ [ZHL24]. The large gap between the two results suggests a substantial cryptanalytic impact of the feed-forward modification.

5-round collision attack on Keccak-EDM^c[448, 384]. The same outer collision framework applies to Keccak-EDM^c[448, 384], targeting the full absorption structure with a 2+3

design split. We construct a 2-round connector that satisfies the required connecting conditions and then complete the attack using the 3-round colliding trail listed in Table 18 (Supplementary Material Section A.3). As in the corresponding Keccak-DM[448, 384] attack, the enlarged rate provides sufficient freedom for the connector construction, allowing the collision to be completed in practice. A concrete collision message pair is given in Table 19 (Supplementary Material Section A.3).

4.5 Collision Attacks on 512-bit Feed-Forward Instances

At the 512-bit security level, the outer collision framework becomes noticeably less effective than for the lower digest instances. Nevertheless, we obtain a 4-round collision attack on Keccak-DM[576, 512] with complexity 2^{238} together with practical 4-round collision attacks on Keccak-EDM^c[576, 512] and Keccak-Sp-F$_{512}$. For comparison, the best known 4-round collision attack on SHA3-512 has complexity $2^{225.29}$ [ZHL24], while no 5-round collision attack has been reported.

4-round collision attack on Keccak-DM[576, 512]. We present a 4-round classical collision attack on Keccak-DM[576, 512], the full-state feed-forward counterpart of SHA3-512. The attack combines a 1-round partial connector with the 3-round colliding trail listed in Table 20 in Supplementary Material Section A.4. The colliding trail has probability approximately 2^{-213}, while the partial connector leaves 25 remaining conditions to be resolved during the final exhaustive-search phase. A message pair satisfying the partial connector is provided in Table 21 in Supplementary Material Section A.4. The resulting complexity is $T_{\text{cl}}^{(512)} = 2^{238}$ evaluations of the 4-round Keccak-p[1600,24].

Although this complexity does not improve upon the best known attack on SHA3-512, it shows that the differential collision framework remains applicable to the Sponge-DM setting even at the 512-bit security level. In contrast, no comparable connector is currently known for SHA3-512. The larger rate of the Sponge-DM construction provides enough message DoF to make the connecting phase feasible.

4-round practical collision attack on Keccak-EDM^c[576, 512]. The outer collision framework applies to Keccak-EDM^c[576, 512], the capacity feed-forward counterpart of SHA3-512. We combine a 1-round connector with the 3-round colliding trail listed in Table 22 in Supplementary Material Section A.4. The enlarged rate provides sufficient message DoF for the connector construction, allowing the collision to be completed in practice. A concrete collision message pair is given in Table 23 in Supplementary Material Section A.4.

4-round practical collision attack on Keccak-Sp-F$_{512}$. For Keccak-Sp-F$_{512}$, we apply the same outer collision framework using a 1-round connector together with the 3-round colliding trail listed in Table 24 in Supplementary Material Section A.4. A concrete collision message pair is reported in Table 25 in Supplementary Material Section A.4. Although Sponge-F$^{\text{P,pd}}$ performs squeezing from the tail of the state rather than from the leading bits, this modification does not significantly affect the colliding trail. The attack remains fully practical due to the large amount of message DoF available during the connector search.

Compared with Keccak-DM[576, 512], Keccak-EDM^c[576, 512] and Keccak-Sp-F$_{512}$ exhibit substantially weaker resistance to this class of collision attacks at the same round count. The difference stems from the location of the feed-forward operation. In both Sponge-EDM^c and Sponge-F$^{\text{P,pd}}$ constructions, the feed-forward affects only the inner capacity part of the state, leaving substantially more freedom for the connector construction than in Keccak-DM[576, 512]. By contrast, the extra outer part collision constraints in

Keccak-DM[576, 512] significantly restrict the connecting phase and make connector construction considerably more difficult. Notably, unlike other attacks, the final trail for Keccak-DM[576, 512] was not obtained by extending a colliding trail backward to derive a connector. Instead, the complete 4-round trail was searched as a whole.

4.6 Inner Collision Attacks on Ascon-DM and Ascon-Sp-F

Both Ascon-DM and Ascon-Sp-F inherit the parameters of Ascon-Hash256. For Ascon-DM, the small squeeze rate prevents the application of the outer collision framework, leaving the inner collision framework as the only effective approach. The situation is slightly different for Ascon-Sp-F. Since the digest is extracted from the inner part of the state in a single squeeze call, an outer collision framework can in principle be formulated. However, the resulting attack structure coincides with that of the inner collision framework. In contrast to Ascon-Hash256 and Ascon-DM, no additional colliding block is required after the attack block. From a cryptanalytic perspective, the outer and inner formulations therefore become equivalent for Ascon-Sp-F.

We thus focus on the inner collision framework. As discussed in Section 3.3, the feed-forward operation does not affect the differential trail at the *attack block*, so the best known differential collision attack pipeline for Ascon-Hash256 can be transferred to both Ascon-DM and Ascon-Sp-F without separate experiments. Specifically, we directly reuse the best known 2-round differential trail from Ascon-Hash256. The 2-round collision attacks of [YLW+23, LLL+25] achieve a complexity of approximately $2^{61.79}$ by exploiting the algebraic structure of the Ascon Sbox to optimize the connecting phase.

For Ascon-DM, the full-state feed-forward introduces only linear terms into the constraint system and does not affect the algebraic connector construction. The same observation applies to Ascon-Sp-F, where the feed-forward is restricted to the inner part of the state. Since the algebraic degree of the resulting system remains unchanged in both variants, the same solving strategy applies directly. Consequently, the time complexity of the 2-round collision attack remains approximately $2^{61.79}$ for both Ascon-DM and Ascon-Sp-F.

4.7 Cryptanalytic Insights for Feed-Forward Sponge Designs

This work evaluates the differential collision resistance of several feed-forward sponge variants. From a cryptanalytic perspective, we summarize the structural reasons why the dedicated collision resistance of these constructions can fall below that of standard SHA-3 in several parameter settings, as well as the settings in which feed-forward does not reduce this resistance. We emphasize that our concrete attacks mainly target the feed-forward sponge instances based on Keccak-p[1600,24], whose parameters are chosen to match the generic security targets of the corresponding SHA-3 instances. Therefore, the following discussion should not be interpreted as a general characterization of all feed-forward sponge constructions. Rather, the observations apply to the differential collision framework, the concrete rate-capacity configurations, and the underlying permutations considered in this work.

First, the full-state feed-forward operation in the Sponge-DM construction fundamentally changes the target collision condition. Instead of requiring the outer part of the output difference to be strictly zero, the feed-forward operation only requires the outer part of the input difference to match the outer part of the output difference. This modification relaxes the strict CICO requirement and brings it closer to a constrained-input only (CI) problem. In this case, the search for the colliding trail only needs to account for the propagation probability, without any restriction on the exact value of the outer output difference. Our experiments indicate that searching for differential trails that satisfy this relaxed identical-difference condition is considerably easier than finding trails that strictly collide to zero in the outer part. This relaxation is the main factor that enables the

6-round attacks on Keccak-DM[288, 224] and Keccak-DM[320, 256] and the 5-round attack on Keccak-DM[448, 384].

Second, the larger rate adopted in the considered Keccak-based feed-forward instances further facilitates collision attacks under the connector constructions used in this work. In instances where the feed-forward is applied only to the inner part, the differential collision framework remains structurally identical to standard SHA-3. However, the enlarged rate provides substantial extra message DoF that can be exploited directly to improve the attacks. This advantage is clearly demonstrated by the 6-round quantum attacks on Keccak-EDM^c[288, 224] and Keccak-EDM^c[320, 256], together with the practical attacks on 5-round Keccak-EDM^c[448, 384], 4-round Keccak-EDM^c[576, 512], and 4-round Keccak-Sp-F$_{512}$. Although extra DoF assist cryptanalysis in general, their impact is especially obvious in these configurations.

Third, a more nuanced comparison between full-state and capacity only feed-forward shows that full-state feed-forward does not always favor the adversary in dedicated collision attacks. When the digest length is large, the constraints imposed on the input of the connecting trail grow considerably. In full-state feed-forward designs, the additional requirement that the input and output outer differences coincide further restricts the connecting phase and makes valid connectors harder to construct. As a result, for large digest lengths, full-state feed-forward can offer stronger resistance to differential collision attacks than capacity only feed-forward designs. At the 384-bit level, this effect already appears in the connector construction. The zero-difference outer-part trails that work for Keccak-EDM^c[448, 384] cannot be extended into valid connectors for Keccak-DM[448, 384], even though both instances are eventually broken in practice at 5 rounds. The effect becomes more decisive at the 512-bit level, where Keccak-DM[576, 512] is markedly harder to attack than Keccak-EDM^c[576, 512] and Keccak-Sp-F$_{512}$ because the colliding trail must be searched as a whole.

Finally, we emphasize the fundamental role of the underlying permutation state size and the propagation properties of the round function. Despite the structural relaxations introduced by feed-forward designs, none of our attacks extends beyond 6 rounds. This limitation stems directly from the strong differential propagation of the underlying Keccak permutation. The robust round function and the large state make it very difficult to find 5-round colliding trails with sufficiently high probability or to build connecting trails that enable a 3-round connector. The same limitation explains why we could not obtain valid 5-round attacks on Keccak-DM[576, 512], Keccak-EDM^c[576, 512], and Keccak-Sp-F$_{512}$, since no 5-round colliding trail that can be converted into a valid attack could be generated by our current search model. This observation is a strong reminder that good permutation propagation remains the ultimate baseline for cryptographic security.

In summary, our results indicate that the dedicated differential collision resistance of a concrete feed-forward sponge instance depends on several distinct factors, including the feed-forward location, the rate-capacity configuration, the digest length, the squeeze rate, and the differential propagation of the underlying permutation. For the SHA-3 counterpart instances studied here, the feed-forward structure can modify the collision condition, while the larger rate adopted to match the generic security target can provide additional message DoF for connector construction. In addition, configurations with a small squeeze rate restrict the adversary to the inner collision framework and remove the extra leverage available in the outer setting, although a small squeeze rate also lowers hashing throughput. At the same time, the differential propagation of the Keccak permutation continues to limit the number of rounds reached by the attacks. Our results therefore provide cryptanalytic feedback for these concrete instances and for the differential collision framework considered in this work, rather than general design conclusions for feed-forward sponge constructions. Broader design comparisons would additionally require considering other parameter choices, underlying permutations, and different classes of dedicated attacks.

5 Conclusion

In this work, our main concrete comparison focuses on `Keccak`-based feed-forward sponge instances that use the same underlying permutation as `SHA-3` and are configured to match the corresponding generic security levels. We analyze these instances under the differential collision framework based on differential trails and connector construction. This framework is particularly sensitive to the rate-capacity configuration, since the feasibility of connector construction depends strongly on the available message DoF.

Within this setting, the considered feed-forward instances do not consistently exhibit stronger dedicated differential collision resistance than their corresponding `SHA-3` instances. For several proposed `Keccak`-based instantiations, the relaxed capacity requirement leads to a larger rate and hence more message DoF, which facilitates connector construction under the attack framework considered here. In addition, full-state feed-forward modifies the outer collision condition from a zero outer output difference to an input-output matching condition, which can further facilitate differential trail search in several instances. However, this effect is not uniform. For large digest lengths, the additional connecting constraints introduced by full-state feed-forward can instead make valid connectors harder to construct.

From a design perspective, these observations are most relevant to feed-forward sponge instances whose connector construction is sensitive to the rate-capacity configuration and whose underlying permutation has a low-degree nonlinear layer. In such settings, using the improved generic preimage and second-preimage security of feed-forward to reduce the capacity and increase the rate should be accompanied by a careful evaluation of dedicated differential collision resistance. At the same time, the large-digest results show that the effect of full-state feed-forward remains instance dependent and cannot be inferred from the rate-capacity configuration alone. Strong differential propagation of the underlying permutation therefore remains an important limiting factor for the attacks considered in this work.

More broadly, these results do not contradict the provable security guarantees, which are derived under the ideal permutation assumption, but rather show that reduced-round variants of feed-forward sponge constructions can still be susceptible to dedicated classical and quantum collision attacks. We hope that the techniques and observations developed in this work will contribute to the future analysis and design of feed-forward sponge hashes and other sponge-like hash constructions.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments. This work was supported by the National Cyber Security-National Science and Technology Major Project (2026ZD1500800), the Fundamental Research Funds for the Central Universities, and the National Natural Science Foundation of China (62532013, 62502493, 62502494). Guozhen Liu is supported by the Fellowship Funds from China Postdoctoral Science Foundation (2024M753187), Siwei Sun is supported by the National Cryptologic Science Fund of China (2025NCSF01012) and the Scientific Research Innovation Capability Support Project for Young Faculty.

References

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions. *2011-STMicroelectronics NXP Semiconductors, Version 0.1 January 14*, 2011. URL: https://keccak.team/files/CSF-0.1.pdf.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Claudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN '98: Theoretical Informatics, Third Latin American Symposium, Campinas, Brazil, April, 20-24, 1998, Proceedings*, Lecture Notes in Computer Science, pages 163–169. Springer, 1998. URL: https://doi.org/10.1007/BFb0054319, doi:10.1007/BFB0054319.

[CHCW26] Longzheng Cui, Kai Hu, Jiamin Cui, and Meiqin Wang. Improved collision attacks on 4-round SHA3-384. In Feng-Hao Liu, editor, *Topics in Cryptology – CT-RSAC 2026: Cryptographers' Track at the RSAC 2026 Conference, San Francisco, CA, USA, March 23–26, 2026, Proceedings*, pages 291–309, Berlin, Heidelberg, 2026. Springer-Verlag. doi:10.1007/978-3-032-22931-1_11.

[DDS12] Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on keccak-224 and keccak-256. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, Lecture Notes in Computer Science, pages 442–461. Springer, 2012. doi:10.1007/978-3-642-34047-5_25.

[DDS13] Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In Shiho Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, Lecture Notes in Computer Science, pages 219–240. Springer, 2013. doi:10.1007/978-3-662-43933-3_12.

[Dwo15] Morris J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions. Technical Report FIPS 202, National Institute of Standards and Technology (NIST), 2015. doi:10.6028/NIST.FIPS.202.

[DZQ+24] Xiaoyang Dong, Boxin Zhao, Lingyue Qin, Qingliang Hou, Shun Zhang, and Xiaoyun Wang. Generic mitm attack frameworks on sponge constructions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 3–37. Springer, 2024. doi:10.1007/978-3-031-68385-5_1.

[GHJ+26] Chun Guo, Kai Hu, Shuntian Jiang, Yanhong Fan, Yong Fu, Bart Preneel, and Meiqin Wang. Permutation-based hash from non-idealized assumptions: Adding feed-forward to sponge. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part VI*, volume 16805 of *Lecture Notes in Computer Science*, pages 65–97. Springer, 2026. doi:10.1007/978-3-032-35412-9_3.

[GLL+20] Jian Guo, Guohong Liao, Guozhen Liu, Meicheng Liu, Kexin Qiao, and Ling Song. Practical collision attacks against round-reduced SHA-3. *J. Cryptol.*, 33(1):228–270, 2020. URL: https://doi.org/10.1007/s00145-019-09313-3, doi:10.1007/S00145-019-09313-3.

[GLS16] Jian Guo, Meicheng Liu, and Ling Song. Linear structures: Applications to cryptanalysis of round-reduced keccak. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *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, volume 10031 of *Lecture Notes in Computer Science*, pages 249–274, 2016. doi:10.1007/978-3-662-53887-6_9.

[GLST22] Jian Guo, Guozhen Liu, Ling Song, and Yi Tu. Exploring SAT for cryptanalysis: (quantum) collision attacks against 6-round SHA-3. In Shweta Agrawal and Dongdai Lin, editors, *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, pages 645–674. Springer, 2022. doi:10.1007/978-3-031-22969-5_22.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ASCON. *IACR Trans. Symmetric Cryptol.*, 2021(3):102–136, 2021. URL: https://doi.org/10.46586/tosc.v2021.i3.102-136, doi:10.46586/TOSC.V2021.I3.102-136.

[HBDM22] Senyang Huang, Orna Agmon Ben-Yehuda, Orr Dunkelman, and Alexander Maximov. Finding collisions against 4-round SHA-3-384 in practical time. *IACR Trans. Symmetric Cryptol.*, 2022(3):239–270, 2022. URL: https://doi.org/10.46586/tosc.v2022.i3.239-270, doi:10.46586/TOSC.V2022.I3.239-270.

[HS21] Akinori Hosoyamada and Yu Sasaki. Quantum collision attacks on reduced SHA-256 and SHA-512. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 616–646. Springer, 2021. doi:10.1007/978-3-030-84242-0_22.

[Ins25] Institute of Commercial Cryptography Standards. Call for comments on submission requirements for cryptographic hash algorithms (draft) and evaluation criteria for cryptographic hash algorithms (draft). https://www.niccs.org.cn/niccs/Notice/pc/content/content_1937427069632253952.html, 2025. [Accessed 16-May-2026].

[LLL+25] Guozhen Liu, Shun Li, Huina Li, Weidong Qiu, and Siwei Sun. Sat-based space partitioning and applications to ascon-hash256 cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2025(3):577–632, 2025. doi:10.46586/tosc.v2025.i3.577-632.

[LLS+26] Xiaoen Lin, Zhengrong Lu, Yantian Shen, Chongxu Ren, Hongbo Yu, Enming Dong, and Wenhao Wu. Practical preimage attack on 5-round keccak[r=640, c=160]. *IACR Trans. Symmetric Cryptol.*, 2026(1):189–211, 2026. URL: https://doi.org/10.46586/tosc.v2026.i1.189-211, doi:10.46586/TOSC.V2026.I1.189-211.

[LM22] Charlotte Lefevre and Bart Mennink. Tight preimage resistance of the sponge construction. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 185–204. Springer, 2022. doi:10.1007/978-3-031-15985-5_7.

[LQT19] Guozhen Liu, Weidong Qiu, and Yi Tu. New techniques for searching differential trails in keccak. *IACR Trans. Symmetric Cryptol.*, 2019(4):407–437, 2019. URL: https://doi.org/10.13154/tosc.v2019.i4.407-437, doi:10.13154/TOSC.V2019.I4.407-437.

[LS19] Ting Li and Yao Sun. Preimage attacks on round-reduced keccak-224/256 via an allocating approach. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 556–584. Springer, 2019. doi:10.1007/978-3-030-17659-4_19.

[QHD+23] Lingyue Qin, Jialiang Hua, Xiaoyang Dong, Hailun Yan, and Xiaoyun Wang. Meet-in-the-middle preimage attacks on sponge-based hashing. In Carmit Hazay and Martijn Stam, editors, *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*, volume 14007 of *Lecture Notes in Computer Science*, pages 158–188. Springer, 2023. doi:10.1007/978-3-031-30634-1_6.

[QSLG17] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New collision attacks on round-reduced keccak. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *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, pages 216–243, 2017. doi:10.1007/978-3-319-56617-7_8.

[QZH+25] Lingyue Qin, Boxin Zhao, Qingliang Hou, Keting Jia, Gaoli Wang, and Xiaoyang Dong. Linear cancellations in the mitm attacks on sponge functions. *IACR Trans. Symmetric Cryptol.*, 2025(3):633–692, 2025. URL: https://doi.org/10.46586/tosc.v2025.i3.633-692, doi:10.46586/TOSC.V2025.I3.633-692.

[SLG17] Ling Song, Guohong Liao, and Jian Guo. Non-full sbox linearization: Applications to collision attacks on round-reduced keccak. In Jonathan Katz and Hovav Shacham, editors, *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, pages 428–451. Springer, 2017. doi:10.1007/978-3-319-63715-0_15.

[SLZ+26] Siwei Sun, Shun Li, Zhiyu Zhang, Charlotte Lefevre, Bart Mennink, Zhen Qin, and Dengguo Feng. Permutation-based hashing with stronger (second) preimage resistance. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part VI*, volume 16805 of *Lecture Notes in Computer Science*, pages 131–162. Springer, 2026. doi:10.1007/978-3-032-35412-9_5.

[TMC+25] Meltem Sönmez Turan, Kerry A. McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-based lightweight cryptography standards for constrained devices. Technical Report NIST SP 800-232, National Institute of Standards and Technology (NIST), 2025. doi:10.6028/NIST.SP.800-232.

[YLW+23] Xiaorui Yu, Fukang Liu, Gaoli Wang, Siwei Sun, and Willi Meier. A closer look at the s-box: deeper analysis of round-reduced ascon-hash. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *Selected Areas in Cryptography – SAC 2023 (Revised Selected Papers)*, pages 22–42. Springer, 2023. doi:10.1007/978-3-031-53368-6_2.

[ZHL23] Zhongyi Zhang, Chengan Hou, and Meicheng Liu. Collision attacks on round-reduced SHA-3 using conditional internal differentials. In Carmit Hazay and Martijn Stam, editors, *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, pages 220–251. Springer, 2023. doi:10.1007/978-3-031-30634-1_8.

[ZHL24] Zhongyi Zhang, Chengan Hou, and Meicheng Liu. Probabilistic linearization: Internal differential collisions in up to 6 rounds of SHA-3. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, Lecture Notes in Computer Science, pages 241–272. Springer, 2024. doi:10.1007/978-3-031-68385-5_8.

Supplementary Material

A Trail Cores and Message Pairs

This section collects the differential trails and concrete message pairs referenced in Section 4. Some trails are taken from the SHA-3 collision attacks in [GLL+20, GLST22].

Notation. Throughout both sections the 1600-bit Keccak state is displayed as a 5×5 array of 64-bit lanes ordered from left to right, with '|' separating lanes. Each lane is written in hexadecimal using little-endian byte order; in differential-trail tables '0' is replaced with '−' for readability. For message tables, each row is one 64-bit lane. The *digest* row lists the final hash output, computed after all construction-specific feed-forward operations, in the same lane-wise hexadecimal format. Each trail row β_i denotes the state difference at the corresponding round. The rightmost column gives the differential probability. The differential probabilities shown for the connecting trails are intended only to indicate the difficulty of connector construction.

A.1 Trails and Messages for 224-bit Feed-Forward Sponge Collision

Table 4: Trail for the 6-round Keccak-EDM^c[288, 224] quantum attack

β_0	A-C2831138D8D-45|-952834-9-5CD829|A23266-A2-D-1-6F|6672C145E28-E-6-|-1-38582888--874 -225A-511CEC1-AB|5-9-624E1-1--8-C|E3B644A-2478--8F|4482428-E2487882|4281C282246--85- 8219A-8B1EF-6-56|852-86C2EE4CA872|2583E68E14E-F-E8|C43A929CCED-9-74|-7BA----922-8-73 861-C4129D-C58-D|958-42CC24D4D--D|262-C68E7D144--7|C-B1C218E7482823|C681421E8844185A -65-6EF19AA48-8D|27-1EC68326888D7|44E1429A22D-1887|C6F22AA64-A4F-7D|25D-461-22-4--C9	2^{-734}
β_1	--8--1----8----4|49---B8--2-----8|--C42----1----A-|--8-1-8---8---4-|2--284--------1- ------8-7------4|498--A----9---1-|-44-2-8--1----A1|-----B----8-124-|2-8418--------1- --8---8---8----2|41---A---2----1-|--442-8--1---2A-|-----B----8---5-|2--28----------4 --------7-----14|49---28--21-----|--8-28---1---2B-|2----B8-------4-|82828-----8----- --8-48--------14|498--88--28-----|--4-22---18---28|----13------135-|2--68-8---8-----	2^{-286}
β_2	-------------9-|-------------29|---------------|-------------42|-------------61 -------------E-|-------------69|-------------3-|-------------22|-------------11 -------------B-|-------------49|-------------2-|-------------A2|--------------1 -------------68|--------------7|-------------23|--------------62|-------------13 -------------9-|--------------9|---------------|-------------E3|-------------6-	2^{-90}
β_3	---------------|---------------|----1----------|---------------|--------------- ---------------|---------------|---------------|---1E----------|--------------- ---------------|---------------|---------------|-----------8---|--------------- ---------------|---------------|-----------8---|-----------8---|--------------- ---------------|---------------|----3----------|---1E----------|---------------	2^{-26}
β_4	---------------|---------------|---------------|------1--------|--------------- ---------------|---------------|---------------|---------------|-----2---------- ---------------|---------------|-----1---------|---------------|--------------- ---------------|---------------|---------------|---------------|-----1E--------- -----4----------|------F--------|---------------|---------------|---------------	2^{-24}
β_5	---------------|---------------|---------------|BC-------------|-214---------- -------------59|A7-------------|---------------|---------------|------2--------- ------2---------|----C----------|--------------B|---99----------|1--------------- 8------------43|---------------|---4-----------|---------------|-------58------- ------4---------|-------2C-------|----------438--|---------------|-----3C---------	2^{-16}

Note. The nominal round weights for β_3–β_5 multiply to 2^{-66}, but from β_3 to a 224-bit collision we found 16 completing trails (8 of total weight 66 and 8 of total weight 65), so the effective probability of the last three rounds is about $2^{-61.4}$, *i.e.*, $8 \cdot 2^{-66} + 8 \cdot 2^{-65}$.

A.2 Trails and Messages for 256-bit Feed-Forward Sponge Collision

Table 5: Message pair following the connector of 6-round Keccak-EDM^c[288, 224]

M_1	3B821D572BA8F6C1|2279FC54E85098AD|128CFC7BA66BB345|AD1F1AE97BA05C11|1F06F037ACD00228 7862F205B3862738|A3E21F8A19B5C023|076BF42854CB2AEC|9F2A3AD2D3229EC6|636EAD5765B364A9 7F2FD81D1CB24AF3|ADA662390C6C9589|CE1FEAED8C37F2C5|2380F8B127761B70|410D51411580BC34 34EE8E6301618F87|45D82F62BAF6B5D2|46568E837B04F3F6|807CA8F00952BBBC|B988764A37B636C8 00000000ECBA7D38|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	BB5C5FA891C0DFC1|2A013D956FA9A8B4|1FC353ACEFF9F3F3|E29C0ED11DE203C6|9F5DF58978C2A27B 9A337EEBE86EDA51|DE9F0BC393A45539|D97162227C725ACF|1DB0F43E462B6A4E|423E6C50759DA7A8 A345D1679A8DB3A7|E51BFB8CA772D0AB|1B55E4FEA2C3F60B|D2BA922BA29D06BC|5498F57E0D035CB1 47D21DF5BD320FD7|18AB2CA5766591F2|021F1BF1DC357FE4|063DB54BDB40CE6E|EC9C48D6BD72C458 00000000EE0AA0F5|0000000000000000|0000000000000000|0000000000000000|0000000000000000

Table 6: Trail for the 6-round Keccak-DM[288, 224] classical attack

α_0	---8--------1---|--8----------1-|----8---2-------|B28C94A7--------|F9ACA24ED2BFF316 7737A4F-FB516125|454B2992F3825374|B63514E3D8369A28|1C816BEEB649EC3-|6CAD37-DF725--C7 1CC4C86132576E9C|C8D34BD7D6E3A6E2|A3A26135B4E41BBB|16E1ABFC-2C9A53E|B-8-572896547C8E 4874F64-31531266|13-C324-7FC--9C8|A5D83C7EB-9113D1|E77664AF5582A9A-|5D8--2E-178C5-86 ---------114-9AA|----------------|----------------|----------------|----------	-
β_0	44216-8111-13744|47384-65B2-12F24|-1354C1412882962|821C289-8A14E745|416E44E-3A11-621 8B4C62DF-F2C-DC1|2B4E4A5E88374283|C76D47-11AB24EC2|C527-A-38778-FC-|4424-6-6131487A2 872-13---38C41C-|C54934D-24A9CEA5|876C1F119CAD5BC6|-5-759-4-CB8C245|82-44-54BBB91-A5 BE-9F8DA97-94D3F|A5-4661331EC471E|-D-95F5C96184A2F|-627141A9761C27C|F-4-4C48BFB4C62D 88482C-C2E29-F44|E-2E6691A88147E-|EA4A6554-2A948C2|A46A4418BB2D824-|-C4-26---71883C1	2^{-782}
β_1	--14-44--4-15---|---2128--28-A2--|---C-22------8--|---2498--1---4--|--24262------2-- --12-64--4-154--|----1-2--2--86--|---8-48-----28--|---24DA--1---2--|--2422------2--- --12-66--4--52--|---41-8--28-A---|---C--2------A--|---24B8--1---4--|--2-24-----1---- --1--68----152--|----1-6--28-A---|---A-----------|---4-BC--1---6--|--26246------8-- --14-4E--4-15---|---2128--28-A---|---C-2-------A--|---249A--1---4--|---426-------2--	2^{-299}
β_2	----------------|--------4------8|---------------8|---------------8|--------4------8 ----------------|--------4-------|-----------8----|-----------8----|---------------- -----2----------|-----------8----|-----2----------|-----------8----|-----2---------- ----------------|----------------|----------------|----------------|---------------- -----2----------|----------------|----------------|---------------8|-----2----------	2^{-23}
β_3	----------------|------------4--|----------------|----------------|---------------- --------8-------|----------------|----1-----------|----------------|---------------- --------8-------|----------------|----1-----------|----------------|-8-------------- ----------------|----------------|----------------|----------------|-8-------------- ----------------|------------4--|----------------|----------------|----------------	2^{-16}
β_4	----------------|----------------|----------8-----|----------------|---------------- ----------------|----------------|-------4--------|----------------|---------------- ------------8--|---4------------|----------------|---------------8|---------------- ----------------|---------------8|----------------|----------------|---------------- ----------------|----------------|-------4--------|----------------|------------1---	2^{-16}
β_5	---8--------1---|--8----------1-|----8---2-------|----1--4--------|----------2---- ---8-2----------|-------1--8-----|--4---------C---|-1-1----------2-|----8---8------2 ---------2--1---|-1---1-------4--|---1--4-1-------|----------1--8--|--------4-----2- ------8-4-------|---1------8-----|1------4--2-2---|----------8---2|------------8-2- ---1------2----4|------------4-1-|---8-4----------|--2-----1-------|---------4--2---	2^{-13}
α_6	---8--------1---|--8----------1-|----8---2-------|----1--4--------|----------2---- ---8-2----------|-------1--8-----|--4---------C---|-1-1----------2-|----8---8------2 ---------2--1---|-1---1-------4--|---1--4-1-------|----------1--8--|--------4-----2- ------8-4-------|---1------8-----|1------4--2-2---|----------8---2|------------8-2- ---1------2----4|------------4-1-|---8-4----------|--2-----1-------|---------4--2---	-

Table 7: Message pair following the connector of 6-round Keccak-DM[288, 224]

M_1	2F8D163F257877EF|B04EE0C7BC570E6B|CBDDD17905236F1C|B9FC69DDA15BA31D|0484838F5B64C1F9 C24AF4CCA621D395|19ACB95CEC9DC1C6|04CDEA3E954E3D42|DEC917AF9389A6B0|B5233F586531295D 0CCB0A32F79A0F37|88D867A8121F15A0|1A882C505CD0F97E|06E1DAFADB8D13BF|7CCBEED67EC84CEB 8AF93C8F97FBED46|179CB3160CC2559B|FE82DB16325CF89F|5BBE31AD828F5C16|0D926C60FFD20E0D 00000000E9E7D27C|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	2F85163F257867EF|B0CEE0C7BC570E7B|CBDD517925236F1C|0B70FD7AA15BA31D|FD2821C189DB32EF B57D503C5D70B2B0|5CE790CE1F1F92B2|B2F8FEDD4D78A76A|C2487C4125C04A80|D98E08559214299A 100FC253C5CD61AB|400B2C7FC4FCB342|B92A4D65E834E2C5|10007106D944B681|CC4BB9FEE89C3065 C28DCACFA6A8FF20|0490815673025C53|5B5AE76882CDEB4E|BCC85502D70DF5B6|50126E80E85E5E8B 00000000E8F3DBD6|0000000000000000|0000000000000000|0000000000000000|0000000000000000

Table 8: Trail core for the 5-round Keccak-DM[320, 256] attack (trail core No. 1 from [GLL+20])

β_2	---------------1|----------------|----------------4|----------------|---------------- ----------------4|----------------4|----------------4|-----------2----|---------------- 2---------------|----------------|----------2-----|----------------|---------------- 2---------------|----------------|----------------|-----------2----|---------------- ----------2----1|----------------|----------2-----|----------------|---------------1	2^{-24}
β_3	---------------1|----------------|----------------1|------4---------|---------------- ----------------|----------------|----------------1|----------------|-----------4---- ----------------|-------------1--|----------------|----------------|-----------4---- ----------------|----------------|----------------|----------------|---------------- ---------------1|-------------1--|----------------|------4---------|----------------	2^{-19}
β_4	----------------|----------------|----------------|----------------|---------------- --------1------4|------4---------|----------------|----------------|---------------- ------8---------|--------------4-|----------------|----------------|-----------4---- ----------------|------1---------|-----------4----|----------------|-1------4------- 4---------------|----------------|-2--------------|----------------|-----1-------4--	1

Table 9: Collision message pair for 5-round Keccak-DM[320, 256]

M_1	FC35FEE1ECBA59EA|E3ED083446801D07|4ECBA2DE37593AE3|DD3B6C08248D94CE|3EB61FE7E7212743 E738409C745860D6|F1A24878503FEB64|E158D4151952FBF8|428545CF96649A36|84E540AC4C5B2B1F A49A909F94EB03A6|A4DC21180AB706B9|05897C8FEE0646BC|13BC7AC87DA0ABC6|40D76D1CE8D0A46E 66D38E8CF42E18A3|3FC5227E77B8483A|54D25A775D178497|64CC6E2ED79C6523|E178328F647580CC 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	FC35FEE1ECBA59EA|E3ED083446801D07|4ECBA2DE37593AE3|DD3B6C08248D94CE|2B069618086B140D 48533FD0477D5EBF|9ADDF942B88AD47F|61A85AF80426383A|8244162BB1149BF9|6AEBC2780A9E01E4 BEAEAEC74111DD84|E53545451E64F665|F62FE2432E72AACE|EE6C692B9A90C635|02D6A85DA91AF77B 3F072FB88159B062|2266A539970805B4|71D196143D81A762|80C467E979A0083A|E2AD1AE9C1F3CEFC 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	1676EDB3B104C5FB|F0D12B03511340D1|D91E78DFDE43C8C0|26F67A59A94E1FC8

Table 10: Trail for the 6-round Keccak-EDM^c[320, 256] attack (β_1 to β_5 from [GLST22])

β_0	DB8-2-C-EF1792A1|CAD28B8C9764E8C5|15-66-41-7394-A1|94123--158F9B-88|214B4-C33F32F161 64C229822-124-A-|D8-2C9-F-A24B--5|51446-8-425-C1A-|5-8-32-1958362C8|D71944C--1-2118- 174-3--D26B2813-|CDC699BCC-AC1-88|1-----51-A41C12-|15C--2A38BC-B2C8|2-9944724D1-15-8 -4E-42--A-528-2-|C82-98-C512411A9|1164--A-425141--|-1A21-C1869-3268|2399E5-3-84284-8 14482---A8D-712-|498-CA8--3A48-81|5--C4125625381--|1-4212-1248-1-6-|A799474A--4841--	2^{-723}
β_1	-48-8--4-----22-|----4------82D88|1--2---46---1---|---14------4--1-|-82-4-228------- -48-C-12------2-|-------6---8---8|1--24---6----48-|-------4--7--21-|-8--4-248------- -48-C-------12-8|-------44--8--2-|1--2---9-----4--|--5-4--2---4--1-|-82-----8------- -48-8--2------2-|-------2-------8|----4---6----4--|--5-4------4-21-|-82---248-----43 ----8---------2-|----4--4---82988|1--2---46---14--|-8--4-----74-21-|--2---228------3	2^{-264}
β_2	----------17--|----------24--|----------7B--|----------A--|----------C2-- ----------14--|----------42--|----------B1--|----------CC-1|----------82-- ----------73--|----------A4--|----------E1--|----------4B--|----------12-- ----------6--|----------1--|----------E--1|----------8-1|----------7--- ----------5--|----------3A--|----------11--|----------C--|----------F6--	2^{-116}
β_3	----------|----------|----------|---------2-----|---------- ----------|----------|----------|--E----------|---------- ----------|---------2-----|----------|---------A-----|---------- ----------|----------|---------8-----|---------8-----|---------- ----------|--8----------|--2----------|--E----------|----------	2^{-24}
β_4	----------|----------|----------|----1----------|---------- ---2----------|----------|----------|----------|---------- ----------|----------|---14----------|----------|---------- ----------|----------|----------|----------|----E---------- ----------|----7----------|----------|----------|----------	2^{-20}
β_5	----------|----------|----------|----------|2C----------1 ----------9--|----------49|----------|----------|----28---------- ----2----------|--5----------|----------1--|-47----------|---------- ----------248-|---------2-----|----------|8----------|------8--------- ----5----------|------4---------|---------248----|----------|---1C----------	2^{-8}

Table 11: Message pair following the connector of 6-round Keccak-EDM^c[320, 256]

M_1	4B8669EE7D6A4B37|FB782A5C8413A68D|DEC89868BE4CE68F|C994FBED5B874D1A|54E22576676571F3 4000EAF4E6856B7D|68B299A582F3E287|B0C44169C0E36C03|11CE71C85B59A5CF|92E2BA6E23DC7D61 87806B5B71C396A8|F6C8535D14EF2390|A8EC8660EE4F3E82|C8FB76202EB3199F|97282105B464B4BD 296E05B94664BA11|D9AE1C8574D458AE|293DEC89E376B079|DE2106E5CBAC8F85|EE367D27BD7A022D 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	95444108C3614AD2|D93E63889758F655|37223E6A151DAE09|5155F2E35521DD1B|9B36A45BF861CD2C 8050A3C82B1BF1B9|F99DBE398EBD5FEF|2B397D0F3BF15047|89BE72C750DCAF7F|1C6662C30C407108 88EAEF6D28951DD8|9F2A5B8FBCEDA780|18064748F27D1248|71FEE5206C373939|1868032AB7C850F4 2CBE4D972770085C|6044E14CE284CCEF|F527C9CDCAE5315F|8A40414D18A88D61|E3A738081DFD3E54 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000

Table 12: Trail for the 6-round `Keccak-DM`$[320, 256]$ classical attack

α_0	--8-------------|-4--------------|--2----2--4-----|-------4--------|514FECE287-A58F4 C7A34-DD12-6722F|4B48A98B79D18ED9|4A41DC1-34857DAA|EFBE88ABDCB211C-|256C559864BB8BDA DC83E-EC-6BC1393|F2431D51BE3BE1FA|F6A656FB89BDECDD|EA-49F3AB-AB5468|35F873-7478-954A 3984D81A93D2241-|3D8A42459C63-9FB|3DD2347E7-DD376D|45-AC6DBE28986-F|-858-D5B-2A19DC9 ----------------|----------------|----------------|----------------|----------------	-
β_0	48--2A181182161C|2874-6B47AC8F65F|A86A5FBA3-17CE78|C-2C41B2E9EAE31E|-8466-1-123FC13E E8-84114-12123-D|481E4FBC32521947|A-1E57A-B9F-2C7C|---C43B-A8E83--8|C-3C-A8-48D51C32 481E-A-838461332|E862-49C8B67-F12|C-6-5E8A51F-2D5F|A-2C4282E1D536-C|A86-46-8587122-- D83A5-8-5AADBE6F|38464338F236AC5-|3-61568863A18F49|3-3E1B3AEB7D9E19|4-12123-DA8-8411 8-7-15--11BA3865|4-7ED65C38AE1B--|E-6EDA988-29-F63|A-6418E2-9E4-5-4|8-3C141-7-8C2664	2^{-783}
β_1	8---2---22------|1----4---5--8---|6---88--1---4---|1---1------2----|A-------48814--9 8---14--22--4---|9----2---5------|4---B8--1-------|B---4------3C---|2---1---48--8--9 C----4--22--8---|1----2---5--C---|4---B8--18-----2|2---1------14---|3-------488----1 -----4--22--C--4|1---12---5--8---|C---A8--1---4---|8---1------3----|3---1---488----9 9---1---2-------|-----6---5--C---|4---38--1-------|3--------2-34---|A---1----88-8--9	2^{-262}
β_2	----1C----------|----------------|----1C----------|----1-----------|----18---------- ----18----------|----18----------|----1-----------|----1C----------|----1----------- -----C-----------|----------------|-----4----------|----------------|----18---------- -----8----------|----------------|------2---------|-----4----------|---------------- -----8----------|------2---------|-----42---------|-----8----------|----1-----------	2^{-43}
β_3	----------------|----------------|----------------|----------------|---------------- ----------------|----------------|----------------|----------------|-------4-------- ----------------|----------------|----------------|--1-------------|---------------- ----------------|----------------|--1-------------|--1-------------|-------4-------- ----------------|----------------|----------------|----------------|----------------	2^{-9}
β_4	----------------|----------------|----------------|-------------2--|---------------- ----------------|--4-------------|----------------|----------------|---------------- ----------------|----------------|-----------2----|-----4----------|---------------- ----------------|----------------|----------------|----------------|---------------- ----------------|----------------|----------------|----------------|----------------	2^{-8}
β_5	--8-------------|-4-----4--------|-12----2--4-----|-------4--------|-1--------8----- -----22---------|4-------2-------|-4--------------|-8--------------|---8-1--------8- ------------8---|1--2-------1----|------4--------8|---4-------2----|-------------2-- ------1-------2-|---------8------|---------1------|-4-------2----2-|-------------2- --1--2-------1--|--------------1-|---1-------2----|-------1--------|-----------1----	2^{-15}
α_6	--8-------------|-4--------------|--2----2--4-----|-------4--------|-1--------8----- -----22---------|4-------2-------|-4--------------|-8--------------|---8-1--------8- ------------8---|1--2-------1----|------4--------8|---4-------2----|-------------2-- ------1-------2-|---------8------|---------1------|-4-------2------|---------------- --1--2-------1--|--------------1-|---1-------2----|-------1--------|-----------1----	-

Note. The listed probability 2^{-15} at β_5 is obtained by only considering the lowest 4 bits of each output difference and not on the remaining output bits. This accounting is therefore lighter than the full DDT weight.

Table 13: Message pair following the connector of 6-round `Keccak-DM`$[320, 256]$

M_1	C7D591085D8A271E|C5621469F16421E3|065C6A017355B8A5|5553E1A08801818F|2C4863E215E74C99 E85AAA49C96E0593|019FB6C2FEA5171A|94B37DE98653E545|7DA47F58CB81452D|8BFD2638FAA67D31 F5573C99356C1E55|68CA970607C43760|E1A33D5E3C89F5D9|83AC65A36F2E665A|7337DFA455796EF7 A57FB1AADE1382FE|6481B644C4F0ED0E|6D6B2FCE292E253A|305A9C329F20E9AD|E5B74AD2FF082AF3 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	C75591085D8A271E|C1621469F16421E3|067C6A037315B8A5|5553E1A48801818F|7D078F0092ED146D 2FF9EA94DB6877BC|4AD71F49877499C3|DEF2A1F9B2D698EF|921AF7F3173354ED|AE9173A09E1DF6EB 29D4DC7533D00DC6|9A898A57B9FFD69A|17056BA5B5341904|69A8FA99DF853232|46CFACA312F9FBBD 9CFB69B04DC1A6EE|590BF4015893E4F5|50B91BB059F31257|75505AE97DA96FA2|EDEF4789FDA9B73A 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000

A.3 Trails and Messages for 384-bit Feed-Forward Sponge Collision

Table 14: Trail for the 4-round Keccak-DM[448, 384] attack

β_1	----------1----|---------------|---------------|----------1----|--------------- ---------------|-------------4-|-8-------------|-8-------------|--------------- ---------------|------2-------4-|------2---------|---------------|------2--------- -8-------------|---------------|-8-------------|-8-------------|--------------- ----------1----|------2---------|------2---------|---------------|------2---------	2^{-21}
β_2	----------1----|---4-----------|----------1----|----------1----|--------------- ---------------|---------------|---------------|---------------|-------4-------- ---------------|---------------|---------------|---------------|-------4-------- ---------------|---------------|----------1----|---------------|--------------- ---------------|---4-----------|---------------|---------------|---------------	2^{-13}
β_3	---------------|---------------|---------------|---------------|--------------- ---------------|--4------------|---------------|---------------|--------------- ---8-----------|---------------|---------------|---------------|--------------- ---------------|---------------|---------------|-------8-------|--------------- -----------4---|---------------|------------2--|---------------|--1------------	2^{-2}

Table 15: Collision message pair for 4-round Keccak-DM[448, 384]

M_1	56401511E540AA16|5485735AB3D118D2|46F4CFD8F2F99547|9A43E6226CCA750A|69F8F30587B24551 231CE41DFB133593|30152ED26C859AF3|6DD7AB6B17904B56|3CC75C4985489832|21751437A027FAB2 6B94EF7CDD04C7BE|942F24545C8AC730|F540AA41EA4D8323|5A104DE215A97B52|5B31EA0C0BEFCDB8 699E1836FE3C254A|4E2EDB14AAE0EE51|E7D88C5BD071E30E|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	56401511E540AA16|5485735AB3D118D2|46F4CFD8F2F99547|9A43E6226CCA750A|69F8F30587B24551 231CE41DFB133593|69120E5E3C95EB3F|7F86D0F0D9616AA7|3207592A5415D533|41C441F1E7A82C77 E1154D79FFF7C7DB|E38D02D22D83FBF2|28EA148CFF44C275|21C0085CACF9799A|8B12348DE70A2523 B62D6F4168CE0276|579C5CBFA24CC230|EE7655B469DDFF6E|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	8C83F1C5C6F02EE5|4D62EFC95A3F2AAF|02446003CCA64669|B66870534DC13750|2803A96DE6072C76 3AC02DCA32CA26CC

Table 16: Trail for the 5-round Keccak-DM[448, 384] attack

α_0	------48-8------|-----------8---|------1-------2-|-4-------------|---4----------4- ----------1----2|C6547E7F7C-85FBC|42-EAC9E5A68BD-A|FC-3C63-2E-D-991|972BD15F62B8EAA1 -E385-98821F-48D|5EE114AE9C8-456C|8D48E-4DC89CC398|F-21-1899D27D291|AAAC3ACFB15F98-D -18B5-6C9E67E--F|864DFF2F318D21D8|-461A8D2-33FCC44|----------------|---------------- ----------------|----------------|----------------|----------------|----------------	-
β_0	--72c-2579ec-4fc|7ded15fa8b6e-4bf|6674841fafd6-3fe|66-6c-aae5d6-4-6|31b-66fec48--5d1 -36-5572eb42-333|79ef943f8a18-6f1|725487af9f98-388|5416a3f626f6--ce|-1d7a351b6ea-1a1 33f9913e6ffe-2db|2ced84fb4e2c--93|735645-b7e8--265|44c6c19bfb12--17|-1b5c7b-13f--1cb -cdfd892--ba-636|1fc-4fe--72c-6d7|777-c6adff1--71c|592fda37e-a6-56e|2eb-2-333-36-557 -3af46a76dd4-34a|5fa611fa8--c-42f|5245-6def47e-715|17c9e6-3f32--3df|67f3227cdffe-5b6	2^{-923}
β_1	----------------|----------------|----------------|----------------|---------------- 8-4-------------|--4--8----------|8-4-------------|-----8----------|8-4--8---------- ----------------|----------------|----------------|----------------|---------------- ----------1-----|--4-------------|--4-------1-----|----------1-----|--4------------- 8----8----1-----|----------------|-----8----1-----|8----8----1-----|8----------------	2^{-26}
β_2	----------------|-------4--------|----------------|----------------|---------------- ----------------|----------------|----------------|-------8--------|-----------2---- ----------------|----------------|----------------|----------------|-----------2---- ----------------|-------8--------|----------------|-------8--------|-------8-------- ----------------|-------4--------|----------------|----------------|----------------	2^{-13}
β_3	----------------|----------------|----------------|-1--------------|---------------- ----------------|------2---------|----------------|----------------|---------------- -------8--------|----------------|----------------|----------------|---------------- ----------------|----------------|----------------|----------------|---------------- ----------------|---------4------|-1--------------|----------------|------1---------	2^{-12}
β_4	------58-8------|-----------8---|------1-------2-|-4-------------|---4----------4- ----------1----2|-----------1---|-----28-4-------|----------5----|--6----2--8----- ------5---------|8------1--------|4---------------|---------------1|-16-2----------- -----------8----|8------------58-|----a-----------|-----2-------1--|--------3------- --c------1-------|--------1-------|--------8----4--|------------b-1-|------a-1-------	2^{-23}
α_5	------48-8------|-----------8---|------1-------2-|-4-------------|---4----------4- ----------1----2|-----------1---|-----28-4-------|----------5----|--6----2--8----- ------5---------|8------1--------|4---------------|---------------1|-16-2----------- -----------8----|8-----------48-|----a-----------|-----2-------1--|--------3------- --c------1-------|--------1-------|--------8----4--|------------b-1-|------a---------	-

Remark. Although β_4 has a large number of active S-boxes (53 in total), the effective propagation weight is only 2^{-23}, because we target a *384-bit collision*. We evaluate propagation through $\chi(\beta_4)$ in two groups of 64 S-boxes each. For the first group, we use the standard DDT of the 5-bit χ S-box. For the second group, an input difference in $\{0, 1, 9, 17, 25\}$ propagates with probability 1; any other input difference would contribute a factor 2^{-1} (such cases do not occur on this trail). The state α_5 is the first DDT-consistent output of $\chi(\beta_4)$ on the 384-bit connector lanes (other bits are not concerned).

Table 17: Collision message pair for 5-round Keccak-DM[448, 384] attack

M_1	F6DEABD514F54831|05D3B2DE0FC2F2E6|45001BABD2934960|61251237E2CD8108|32AB055152E63151 C7F80F5CF1715F26|213829D63F53826E|8831DC06E42B31E2|0D5BBCD0B258E233|CE60D8F30ACB08CE B83658138B3BD356|441AF1E8BC331F33|F08C9D0B3752EC8A|88227F682ED5879C|F3CBB10F0EAE2C3E 737B460AF9BCE4BE|C0B9C816988B8CC5|E1DFF71C26A16D42|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	F6DEAB9D1CF54831|05D3B2DE0FC272E6|45001BBBD2934940|65251237E2CD8108|32AF055152E63111 C7F80F5CF1615F24|256EE3898767BB01|4037FA18727B60F0|E4D752005E537DB9|59693685FEACB28F 5E3500CB177F56CB|9DAB8086E4881A8D|02A31EB03F482B0B|1903EFC2713297D2|97D65AC89B63BEF0 A08CC64770AED9C3|E4FB11D9688D268D|E564AFC8CFBAE75F|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	554C5DDEB1920F30|B5B1893BC7B9B786|9201578154617668|39F806DEEC7E0952|22BD970FEAFB5205 EF0ACE3D52ECED03

Table 18: Trail core for the 5-round Keccak-EDM^c[448, 384] attack

β_2	----------|----------|----------|----------|---------- -------4--------|---------2------|-----------4---|-------4----4---|---------- -------4--------|----------|4----------|4----------|4---------- ----------|---------2------|----------|----------|4---------- ----------|----------|-----------4---|-----------4---|----------	2^{-21}
β_3	----------|----------2-|----------|----------|---------- ----------|----------|------2---------|----------4-|---------- ----------|----------|----------|----------4-|---------- ----------|----------4-|----------|----------|----------4- ----------|----------2-|------2---------|----------|----------	2^{-15}
β_4	----------|----------|----------|----------|---------- ----------|----------|----------|---8----------|-------4-------- ----------4-|-----8----------|--------8-------|----------|---------- ----------|----------|----------1----|----------|---------- ----------|2---------|----------|----------|----------8-	1

Note. The probability 1 at β_4 is obtained by only considering the lowest 1 bit of each output difference of the second 64 Sboxes and not on the remaining output bits.

Table 19: Collision message pair for 5-round Keccak-EDM^c[448, 384] attack

M_1	45A9C9A443F835A8|EBEAA22F4EDE0D33|11911F7D05BCFD98|3E112C856E5FFBAB|F3B5CD0B0A1CD2B9 AE19458C2BB605C6|DB77BB1E8D733B9D|A8A41950D6A999ED|165AAFCA8C346186|151AEBE0C5B067BC CB03E9D749A88BAE|B7D439E7A2A092B7|359D446C3D842CE5|14DEB54CA80F83CD|B785EA1DC5B7D469 BDBA387103DBFD14|571D199C2066AF8E|E3DCF3CE40E9BB92|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	8E81AF3B561E99E2|7644E9320FD40D57|2AFBBBC33769DD76|E8595268837DEABA|6B628C1272049553 C059E44A67B6C22B|2F00FB0C47F017DF|A2ED1C3FCC8DA224|F763F1F2C800B67A|45FFEF620EE15096 C105AA685A18D48C|9C10F6B999033467|1136542B6640C54C|183E558E3873534A|7DE0E3480CCAE671 073E48FEF83EFB30|63AB5BA9A1CD2705|E2F239D44B9484AD|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	18B4140A8AA52AAB|328D54D217872364|505CB42460B7CDF4|C689805F6BDD8FA6|0884A8B1EC5523AB 9BFE784BEC315D93

A.4 Trails and Messages for 512-bit Feed-Forward Sponge Collision

Table 20: Trail for the 4-round Keccak-DM[576, 512] attack

α_0	26416-881EC31842|238----238-91--2|2D6B4-12-2881-11|81-8--9A-4--843-|92A-1-1824-91848 8894-54--15-1E25|1895--457352-224|1--3-46-335632-8|88-4166-831EF4A3|CB945BD6F9EFB7BD -774D5324A938EEE|315453-EB-6B6E5-|27BC86CAFD632AA3|C118F1C3FBAC2E76|1729D5-2D51F9F2B -37F32713469E725|----------|----------|----------|----------	-
β_0	7CDE58D7E-5AD-7-|17CFC87E2-7BF8B4|BF58F5EEA-DF8FF9|E41BA9ED2-D-DDFF|788BEF982-6F276A AD94FED36-7E7FFB|9F9478F36-1563DB|EF5F6B6DA-5236E2|6FDB13EEE-FF5F-2|B34D13A76-4A9837 79EE-FF18-34DDB5|9AE656CC4-BCED62|C64DC68EE-8FEE1D|A9E22FBE6-81BC9D|E17FFA672-C96A7C BCD34CCFA-784F91|FC9D617D2-B3D1FF|8D53D121E-4222B8|4E9D812A6-DECD34|-686EFFF2-DD4F69 2DC-F74A4-37346B|D28775F4A-2F28CE|BF8F4BC54-1B96B5|E-5E2EB3C-145D95|7DDC1FEBE-4DFB62	2^{-1007}
β_1	------4----2----|----------2----|------4----2----|------4----2----|---------- ----------8---|------4---------|----------8---|----------8---|------4--------- ------4---------|8---------------|8-----4---------|----------|---------- 8-------1--28---|----------8---|8-------1--28---|----------8---|--------1---8--- --------1-------|----------|----------|----------|----------	2^{-34}
β_2	------8---------|----------4----|-----4----------|----------|---------- ----------|----------|-----4----------|----------|-------8-------- ------8---------|----1-----2-----|----------|----------|-2--4---------- ----------|----------8--|---1----------|--2----14---4---|---------- ----------8---|----------|----------|----------1--2-|----------	2^{-38}
β_3	-2--6-88--4A184-|--8-----38-1---2|3C43---2-68814-1|-328---2----8438|9-2-5-1-14-8---8 8----5----4-1C-1|-894-4-5-2---224|1--3-44--35-C2--|--------7--2---4|---8122-5--4F1-C ---7--2---481---|1-2542A--9E218--|3-----A----8-3-1|224-814-4-582---|8-2--12A61---8-1 -A-2-281---112--|-4A184--2--6--8-|-E4-4-----2-----|C8814-13E43---2-|2--4-1C-18---15- --1-254-A--9E218|---2--E--C----28|2-281--111----A-|942-8--4-2C-1---|---E--4---842-8-	2^{-141}
α_4	26416-881EC31842|238----238-91--2|2D6B4-12-2881-11|81-8--9A-4--843-|92A-1-1824-91848 8894-54--15-1E25|1895--457352-224|1--3-46-335632-8|8--8132-3-46B5-D|889C13255-44ED29 3-2742A--1A813-1|3-65C3C-49FA3B-1|B-4-81EA611--B-1|A247816A611838--|9--743AA61EA18-1 -C23C2812--7128-|4E61C412E416--8-|C6C54-93F42--17-|EA8743D3FC31-37-|2CA785C138-113D- 2-3835E1BD-9E23-|B42A8-E51E4-1-88|342E9-4513C-3-A-|942AA544824DB28-|--1E25E-A485E2B8	-

Note. The listed probability 2^{-141} at β_3 is obtained by evaluating $\chi(\beta_3)$ in two groups of 64 Sboxes each. For the first group we use the standard DDT of the 5-bit Sbox. For the second group the weight depends only on the lowest three bits of each output difference, namely bits indexed 0, 1, and 2 within the 5-bit χ output, and not on the remaining output bits. This accounting is therefore lighter than the full DDT weight.

Table 21: Message pairs following partial connector of 4-round Keccak-DM[576, 512] attack

M_1	347A12B3E3ED0EC9|DA30FCA3D039A29D|73C6B3B124DA3C9B|34020778148200BB|3020D52D3939FF08 BAB2D39BC3AE6202|E67B60F54BD843E4|1A514BD408E3F168|E389E5A4F0C4C835|06A5639F71C043C6 CCE594174BD1390A|B8FEDBA4C064E5CE|5B2C95AE72E38533|BC3E68CD9DFEF35D|D69CEE459459709B EFD1701D2A88675F|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	123B723BFD2E168B|F9B0FCA1E830B29F|5EADF3A326522C8A|B50A07E21082848B|A280C5351D30E740 3226D6DBC2FE7C27|FEEE60B0388A41C0|0A524FB43BB5C360|6B8DF3C473DA3C96|CD313849882FF47B CB9141250142B7E4|89AA88AA700F8B9E|7C9013648F80AF90|7D26990E6652DD2B|C1B53B474146EFB0 ECAE426C1EE1807A|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000

Table 22: Trail for the 4-round Keccak-EDM^c[576, 512] attack

β_1	----------|----------|----------|----------|---------- -4--------|-4-2--4---------|---2------|-4----4---------|-4-2------ -4--------|-4--------|----------4-----|-4--------|-4--------4----- ----------4-----|---2------|---2------4-----|----------|---2------4----- ----------|------4---------|----------|------4---------|----------	2^{-26}
β_2	----------|--------2-------|----------|----------|---------- ----------|----------|2---------|--------4-------|---------- ----------|----------|----------|--------4-------|---------- ----------|--------4-------|----------|----------|--------4------- ----------|--------2-------|2---------|----------|----------	2^{-15}
β_3	----------|----------|----------|----------|---------- ----------|----------|----------|----------8--|-4-------- --------4-------|----------8|--8---------|----------|---------- ----------|----------|-----1----------|----------|---------- ----------|----------2-----|----------|----------|--------8-------	2^{-4}

Table 23: Collision message pair for 4-round Keccak-EDM^c[576, 512] attack

M_1	964A643901E3F309|27B720CB8C0C3DAA|5C13E7117029E0CA|10B071D3D1020B02|D44184FAB7C302F0 674F25A84E75134C|85E3E086EC26776C|63A74CFFA10DF73B|2919DA44C0C00B56|C2C3D89AEF129F2A F873A378A7310A82|9E27389515C485A6|0F4ECB5D98F7D8F2|159B659993837781|8EF3CEFB9C187A5B E606638B6F04FC41|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	37E687C2D9BE4713|ED43D03F140C27A1|79436C00702FE0E7|17CD04EDEAD004F6|D418704831B720BE 4B4D4A9F65F81F4C|870D60822C9586D6|E676627DD15630F4|1C9993E4C60002E6|DAEDBFE95C959E3A 6350B1C6D96A2583|98F0DF86C409F2D0|A3BDC0EF99E60F8D|100717746A23DFA1|131D304A134D5C8B EB2C21A0C0163C60|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	B145147FC5FDA4FE|38B3A30257AA168C|CF671B2FE93BBB8A|650FF38FC4FB40F4|792A5DFD8F64ED1A 3E4AAA06EEF7CD5B|1F6F1C4783AE5F0D|3ABD9F6692394EE1

Table 24: Trail core for the 4-round Keccak-Sp-F_{512} attack

β_1	---------2------|-8-------2------|-8-------2------|-8----------|-8-------2------ ----------|----------|----------|----------|---------- ----------|----------|----------|----------|---------- -8--------|-8--------|-8--------|-8--------|-8----------1-- ---------2------|----------|---------2---1--|---------2------|----------1--	2^{-20}
β_2	----------|----------|----------|----------1----|----------4----- ----------8-----|----------|----------|----------|----------4----- ----------|----------|----------|----------1----|---------- ----------|----------|----------|----------|---------- ----------8-----|----------|----------|----------|----------	2^{-12}
β_3	----------|----------|----------|----------|---------- ----1----------|-----4----------|----------|----------|---------- ----------|----------|-----2----------|----------|-----2---------- ---2----------|-8--------|----------|----------|---------- ----------|----------|----------|----------|----------	2^{-4}

Table 25: Collision message pair for 4-round Keccak-Sp-F$_{512}$

M_1	3FDD3CAB68FF3270|6728B888D340E51B|30233A3564463803|28F1856AB90A53EE|61AB01E920D11061 41603154DC006975|4AEB1E41EAB8D76F|00A68AA285A20852|AD868100A5B0012C|00A1FA635800394F 6E561D2D300829B6|0F238919270EC438|CB2B68B3863FDFAE|0C12E283A4606DA5|CA9E85F8193CCE47 E494215240000975|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
M_2	0FD2FC7513DAFA5C|672801784EC05D13|3018ED002B4FAB03|ADE264AAB999E01E|E98D0B7873602E1A 28FD2F905AA4AEFE|D386EE58CFB8D75A|B09E8F338C896D63|AF89CA9726DC052C|1AE63A723A62A3D7 DE5B0A6029FCF870|1322E4545DFB346E|0B828885C435BCB1|49A45A83A5A04FE0|622A06312AA10E33 EC5EE126CCA4FD4C|0000000000000000|0000000000000000|0000000000000000|0000000000000000 0000000000000000|0000000000000000|0000000000000000|0000000000000000|0000000000000000
digest	CC3682CB3932E6F1|C784D10C2F3F5DB3|886B0B8AD40A189F|374CD7AE47DEEF92|CA315CF0394FB352 A0754BCC44FFDE6F|D257B23BE3E815BC|7307CD718CF8EF54