

Truncated Differential Preimage Attacks via Differential-Linear Correlations

Chunning Zhou², Kai Hu^{1,3,4}, Zhongfeng Niu², Thomas Peyrin², and Hongyi Zhang²

¹ School of Cyber Science and Technology, Shandong University, Qingdao, China
kai.hu@sdu.edu.cn

² Nanyang Technological University, Singapore
{chunning.zhou, zhongfeng.niu, thomas.peyrin}@ntu.edu.sg,
hongyi003@e.ntu.edu.sg

³ State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Qingdao, China

⁴ Suzhou Research Institute, Shandong University, Suzhou, 215123, China

Abstract. We derive joint truncated differential (TD) probabilities from multiple differential-linear (DL) approximations and use them in a framework for preimage filtering. For a fixed input difference, the inverse Walsh transform recovers the probabilities of affine output cosets from DL correlations over a complete mask subspace. In our applications, these correlations are estimated using the round-based DL method. A filter accepts a selected union of affine cosets, whose probability is computed from their joint distribution without assuming independence among the DL events. The framework handles multiple targets, sharing base evaluations and confirming each accepted candidate only once. An independent sampling analysis allows candidate sets from different iterations to overlap and relates success probability to confirmation cost. Applying this framework, we obtain the first preimage attacks on **KNOT-Hash** below the designers' claimed preimage bounds, for 9 to 12 rounds. For all four **JH** members, we obtain preimage attacks on the reduced-round hash functions under the padding rule of the specification, covering 3 rounds per compression-function call for **JH-224/256/384** and 4 rounds for **JH-512**, with speedups ranging from 2^{54} to 2^{158} over generic preimage search. For reduced-round **SKINNY-Hash**, we also present the first second-preimage attacks on target messages containing at least two padded blocks, covering 5 and 6 rounds of **SKINNY-tk2-Hash** and 5, 6, and 7 rounds of **SKINNY-tk3-Hash**. As a further application of the DL-to-TD construction, we give a 6-round TD distinguisher on **Xoodoo/Xoodyak** in the related-key setting, whose estimated sample complexity improves on the previous best by about 25 bits.

Keywords: Truncated differential · Preimage attacks · Hash function · Differential-linear

1 Introduction

Preimage attacks exploit the internal structure of a hash function to find messages with a prescribed digest more efficiently than generic search. Several techniques have proved useful for reduced-round designs. Linearization methods impose conditions under which selected state bits become linear in the remaining variables. Relations obtained from the target digest can then reduce the search. Li *et al.* [18] combine such structures with guessing in their preimage attacks on **Ascon**. Meet-in-the-middle (MitM) attacks divide the computation into parts controlled by separate sets of variables and combine the results by matching intermediate values [24,31]. The choice of variables and the conditions imposed on them determine how much freedom remains for the search.

Statistical relations between the outputs of related inputs provide another way to save work. Niu *et al.* [22] use differential-linear (DL) approximations to filter preimage candidates. A DL approximation associates an input difference with a biased linear condition on the output difference. Once a base input is evaluated, these conditions help filter its related inputs against the target digest. Only candidates that pass the filter require further evaluations. The attack thus uses one evaluation to test several candidates through their output relations.

However, combining several DL approximations for the same input difference requires the probability that their conditions hold jointly. In the strictest strategy of Niu *et al.*, this probability is estimated by multiplying the individual probabilities. This calculation assumes independence among the DL events, which does not follow from linear independence of the masks. Their success analysis also assumes that candidate sets from different iterations are disjoint. This condition limits the sampling budget and can exclude useful attack parameters. Estimating the resulting speed-up requires both joint probabilities and an analysis of the sampling process.

From DL correlations to truncated differentials. Truncated differential (TD) cryptanalysis, introduced by Knudsen [16], allows the output difference to range over a set. Its probability is the aggregate probability of the differences in that set. This gives a direct way to describe the joint outcomes of several DL conditions. The connection between TDs and multidimensional DL approximations was established by Blondeau, Leander, and Nyberg [5,6]. Guo *et al.* [12] used this connection to construct TDs for **CTC2**. For a fixed input difference, the inverse Walsh transform converts DL correlations over a complete mask subspace into the probabilities of the corresponding output cosets. Each coset describes one joint outcome of the mask conditions.

Recent progress in DL cryptanalysis makes these correlations more accessible. The differential-linear connectivity table [1] and subsequent algebraic and geometric methods [20,15,3] have improved the estimation of DL correlations. In our applications, we use the round-based (RB) approximation of Hu, Niu, and Wang [14] to obtain the correlations used in the TD construction. This avoids enumerating the differential trails that contribute to each output coset.

Our contributions. We develop a TD-based framework for preimage filtering that captures the joint distribution of the output conditions based on the TD/DL connection. This gives the probabilities needed to combine dependent DL events and to decide which outcomes are worth accepting. The framework supports both a single target and multiple targets and provides a success analysis that allows candidate sets from different iterations to overlap.

More specifically, for a fixed input difference and a complete subspace of output masks, the inverse Walsh transform converts the corresponding DL correlations into the probabilities of all affine output cosets. A filter may accept any selected union of these cosets, whose acceptance probability is obtained by summing their probabilities, replacing the product estimate of Niu *et al.* [22] by a joint probability that accounts for dependencies among the DL events. The choice of accepted cosets also becomes part of the attack design and is selected according to the overall gain, balancing the increase in preimage success probability against the additional confirmation evaluations. For multiple targets, preprocessing combines their acceptance sets so that they share a base evaluation and each accepted candidate is confirmed only once. For fixed functions, targets, and filters, we further analyze independently and uniformly sampled base inputs and derive the resulting success probability and expected confirmation cost even when candidate sets overlap. This removes the disjointness restriction on the iteration budget of [22].

We apply the framework to **KNOT-Hash**, **JH**, **SKINNY-Hash**, and **Xoodoo/Xoodyak**. The resulting preimage and second-preimage attacks, together with the related-key distinguisher, are summarized in Table 1 and detailed below.

- **Preimage attacks on KNOT-Hash.** We obtain the first preimage attacks on three of the four **KNOT-Hash** members with complexities below the designers’ claimed security bound, covering 9, 10, and 11 rounds and reaching 12 rounds for **KNOT-Hash** (512, 512, 64, 256). In particular, the attack on 11-round **KNOT-Hash** (256, 256, 32, 128) requires an iteration budget that is excluded by the disjointness condition of [22] but permitted by our analysis of independent sampling.
- **Preimage attacks on JH.** For all four **JH** members, we obtain preimage attacks on the reduced-round hash functions under the padding rule of the specification. We show that when several input differences yield output differences in a common linear subspace with probability 1, the same holds for their linear span, allowing one test to filter an entire input coset. The resulting attacks cover 3 rounds per compression-function call for **JH-224/256/384** and 4 rounds for **JH-512**, with speedups ranging from 2^{54} to 2^{158} .⁵
- **Second-preimage attacks on SKINNY-Hash.** We use the intermediate capacity states of a given message as the targets of the multi-target attack. A preimage of any one of these states gives a new prefix that can be joined

⁵ Our attacks cross the last two compression calls, a free message block followed by the fixed padding block, each reduced to R rounds, so the differential path covers $2R$ rounds. Compression-function attacks instead count rounds in a single call, a less constrained setting, so their round numbers are not directly comparable.

Table 1. Summary of our results. Complexity denotes time for (second) preimage attacks and data for distinguishers. For **SKINNY-Hash**, $K \leq \ell - 1$ counts the guaranteed reachable targets among the ℓ padded message blocks, and $g(K) = \log_2$ Gain; costs are leading estimates (Appendix D.2).

Target	Rounds	Complexity	Baseline	Ref.
Preimage attacks				
KNOT-Hash (256, 256, 32, 128)	9	$2^{122.715}$	2^{128}	Sect. 4
	10	$2^{125.166}$		Sect. 4
	11	$2^{126.893}$		Sect. 4
KNOT-Hash (384, 384, 48, 192)	9	$2^{185.891}$	2^{192}	Sect. 4
	10	$2^{188.496}$		Sect. 4
	11	$2^{190.404}$		Sect. 4
KNOT-Hash (512, 512, 64, 256)	9	$2^{247.626}$	2^{256}	Sect. 4
	10	$2^{248.436}$		Sect. 4
	11	$2^{250.950}$		Sect. 4
	12	$2^{253.415}$		Sect. 4
JH-384	3	$2^{226.001}$	2^{384}	Sect. 5
JH-256	3	$2^{138.001}$	2^{256}	Sect. 5
JH-224	3	$2^{122.001}$	2^{224}	Sect. 5
JH-512	4	$2^{458.001}$	2^{512}	Sect. 5
Second-preimage attacks				
SKINNY-tk2-Hash	5	$2^{224-\log_2(K)-g(K)}, 0 < g(K) \leq 9.909$	$2^{224-\log_2(K)}$	Sect. 6
	6	$2^{224-\log_2(K)-g(K)}, 0 < g(K) \leq 1.497$		Sect. 6
SKINNY-tk3-Hash	5	$2^{256-\log_2(K)-g(K)}, 0 < g(K) \leq 11.674$	$2^{256-\log_2(K)}$	Sect. 6
	6	$2^{256-\log_2(K)-g(K)}, 0 < g(K) \leq 9.414$		Sect. 6
	7	$2^{256-\log_2(K)-g(K)}, 0 < g(K) \leq 1.783$		Sect. 6
Related-key distinguishers				
Xoodyak	6	$\approx 2^{95}$	—	[7]
-initialization	6	2^{70}		Sect. 7.2

Data count four-tuples for [7] and pairs here, using inverse-capacity estimates. Our pairs use independently sampled base-key offsets (Section 7.2).

to the original message. For reduced-round **SKINNY-Hash**, our framework yields the first second-preimage attacks on target messages with at least two padded blocks, covering 5 and 6 rounds of **SKINNY-tk2-Hash** and 5, 6, and 7 rounds of **SKINNY-tk3-Hash**.

- **Truncated differential distinguisher on Xoodyak/Xoodyak.** We apply the DL-to-TD construction to a distinguishing problem. We obtain a 6-round related-key TD distinguisher on **Xoodyak/Xoodyak** with independent random offsets to the base key and sample complexity 2^{70} pairs, about 25 bits below the estimate for the previous higher-order DL (HDL) distinguisher of Che and Tian [7].

Outline. Section 2 fixes the notation and recalls truncated differentials, differential-linear correlations, the round-based DL method, and the DL-based preimage framework. Section 3 builds the TD framework and its complexity analysis. Sec-

tions 4, 5, and 6 instantiate the framework on KNOT-Hash, JH, and SKINNY-Hash, respectively, and Section 7 gives the Xoodoo/Xoodyak distinguisher.

2 Preliminaries

2.1 Notation

We work over $\mathbb{F}_2^n$, the n -bit vector space over the field $\mathbb{F}_2 = \{0, 1\}$. For $x, y \in \mathbb{F}_2^n$, their binary inner product is $x^\top y = \sum_{i=0}^{n-1} x_i y_i \in \mathbb{F}_2$. Vector addition is denoted by either $\oplus$ or $+$. For a linear subspace $V \subseteq \mathbb{F}_2^n$, its orthogonal complement is $V^\perp = \{\lambda : \lambda^\top v = 0 \text{ for all } v \in V\}$. For $u \in \mathbb{F}_2^n$, the set $u + V = \{u \oplus v : v \in V\}$ is the coset of V represented by u . For a finite set $\mathbb{D}$, $\overline{\mathbb{D}}$ denotes its $\mathbb{F}_2$ -linear span, and $|\mathbb{D}|$ its cardinality. Concatenation is denoted by $\|$. For a finite index set I , $\mathbb{R}^I$ denotes the space of real vectors indexed by I , with $v[i]$ denoting coordinate i . All logarithms are to base 2 unless stated otherwise, and $\mathbb{E}_x[\cdot]$ denotes expectation over the random variable x .

2.2 Truncated differentials and differential-linear correlations

Truncated differential. Let $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ be a vectorial Boolean function, $\delta \in \mathbb{F}_2^m$ a fixed nonzero input difference, and $V \subseteq \mathbb{F}_2^n$ a nonempty subset. A TD over F is determined by the pair (δ, V) : the input difference is δ , and the output difference ranges over V . Its probability is

$$\Pr[\delta \xrightarrow{F} V] = 2^{-m} |\{x \in \mathbb{F}_2^m : F(x \oplus \delta) \oplus F(x) \in V\}| = \sum_{\Delta \in V} \Pr[\delta \xrightarrow{F} \Delta],$$

where $\Pr[\delta \xrightarrow{F} \Delta]$ is the ordinary differential probability of output difference Δ . If $|V| = 1$, the TD reduces to an ordinary differential. In general, we compare its probability with the uniform baseline $|V|/2^n$, the acceptance probability when the output difference is uniform in $\mathbb{F}_2^n$.

Differential-linear approximation. One-dimensional DL cryptanalysis uses a nonzero input difference $\delta \in \mathbb{F}_2^m$ and an output mask $\lambda \in \mathbb{F}_2^n$. The pair (δ, λ) describes the parity $\lambda^\top (F(x) \oplus F(x \oplus \delta))$, and its correlation is

$$\mathcal{E}_{\delta, \lambda} = 2^{-m} \sum_{x \in \mathbb{F}_2^m} (-1)^{\lambda^\top (F(x) \oplus F(x \oplus \delta))}.$$

For a nonzero mask, a positive correlation favors parity 0, whereas a negative correlation favors parity 1. The zero mask gives the constant coordinate $\mathcal{E}_{\delta, 0} = 1$.

Multidimensional differential-linear approximation. The one-dimensional DL approximation generalizes to the multidimensional setting [6] by replacing the single mask with a linear subspace of masks. Let $W \subseteq \mathbb{F}_2^n$ be a k -dimensional mask subspace with basis $\lambda_0, \dots, \lambda_{k-1}$, and let $U = W^\perp$, so that $\dim U = n - k$ and $|U| = 2^{n-k}$. The k parities associated with a basis of W have 2^k joint

outcomes, one for each coset of U . In particular, all parities are zero exactly when the output difference lies in U ; its probability $\Pr[\delta \xrightarrow{F} U]$ is compared with the uniform baseline $|U|/2^n = 1/|W|$. For $W = \{0, \lambda\}$ with $\lambda \neq 0$, this bias is $\frac{1}{2}\mathcal{E}_{\delta, \lambda}$. The other joint outcomes correspond to the TDs $\delta \xrightarrow{F} u + U$. Section 3.1 reconstructs the probabilities of all these cosets from the DL correlations.

Remark. More generally, a TD can take a nonempty input difference set $D \subseteq \mathbb{F}_2^m \setminus \{0\}$, with probability $\Pr[D \xrightarrow{F} V] = |D|^{-1} \sum_{\delta \in D} \Pr[\delta \xrightarrow{F} V]$. Such a generalization is used in [6] to link multidimensional DL attacks with truncated differentials. We define TD probabilities for individual differences and combine several such differences in the attacks below.

2.3 Round-based approximation of DL correlations

The TD constructions of this paper take DL correlations as input, so we recall the round-based (RB) approximation of Hu, Niu, and Wang [14], which builds on Beyne’s geometric approach [3].

DL correlation as a coordinate. Let $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$. The 2-wise form of F is the map

$$F^{\times 2} : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \times \mathbb{F}_2^n, \quad (x, \theta) \mapsto (F(x), F(x) \oplus F(x \oplus \theta)).$$

For a nonempty multiset $\mathbb{P}$ of vectors in $\mathbb{F}_2^{2n}$, its *correlation vector* is $\text{CV}(\mathbb{P}) \in \mathbb{R}^{\mathbb{F}_2^{2n}}$ with

$$\text{CV}(\mathbb{P})[u] = \frac{1}{|\mathbb{P}|} \sum_{x \in \mathbb{P}} (-1)^{u^\top x}, \quad u \in \mathbb{F}_2^{2n},$$

where elements are counted with multiplicity. For a function $G : \mathbb{F}_2^a \rightarrow \mathbb{F}_2^b$, its correlation matrix has rows indexed by output masks $v \in \mathbb{F}_2^b$ and columns by input masks $u \in \mathbb{F}_2^a$:

$$\mathbb{C}^G[v, u] = 2^{-a} \sum_{z \in \mathbb{F}_2^a} (-1)^{u^\top z + v^\top G(z)}.$$

For a fixed nonzero $\delta \in \mathbb{F}_2^n$, taking the input multiset $\mathbb{P} = \mathbb{F}_2^n \times \{\delta\}$ and the output multiset $\mathbb{C} = F^{\times 2}(\mathbb{P})$ gives $\mathcal{E}_{\delta, \lambda} = \text{CV}(\mathbb{C})[(0, \lambda)]$. The geometric approach gives the exact propagation rule

$$\text{CV}(\mathbb{C}) = \mathbb{C}^{F^{\times 2}} \text{CV}(\mathbb{P}). \quad (1)$$

Round-by-round factorization. For an r -round function $F = F_{r-1} \circ \dots \circ F_0$, the correlation matrix factorizes as $\mathbb{C}^{F^{\times 2}} = \mathbb{C}^{F_{r-1}^{\times 2}} \dots \mathbb{C}^{F_0^{\times 2}}$. The exact intermediate vectors satisfy $\nu_0 = \text{CV}(\mathbb{P})$ and $\nu_{i+1} = \mathbb{C}^{F_i^{\times 2}} \nu_i$ for $0 \leq i < r$, so $\nu_r = \text{CV}(\mathbb{C})$. The RB method approximates these vectors by a representation whose size is linear in the number of S-boxes and exponential only in their size.

One substitution-permutation round. Let a round $F_i = L \circ S$ act on N_s parallel w -bit S-boxes, with $n = N_s \cdot w$, $S = s \parallel \dots \parallel s$, and a linear map L . Write a mask pair as (v_0, v_1) and let $v_{a,j} \in \mathbb{F}_2^w$ be cell j of v_a , for $a \in \{0, 1\}$ and $0 \leq j < N_s$. At a factorization step, RB stores N_s component vectors $\sigma_i^{(j)} \in \mathbb{R}^{\mathbb{F}_2^{2w}}$ and represents the current estimate by

$$\sigma_i[(v_0, v_1)] = \prod_{j=0}^{N_s-1} \sigma_i^{(j)}[(v_{0,j}, v_{1,j})].$$

This rank-one representation is exact for $\sigma_0 = \mathbf{CV}(\mathbb{P})$, since the correlation vector factorizes across cells. In cell-pair coordinates, the S-box layer has correlation matrix $\mathbf{C}^{5 \times 2} = \bigotimes_{j=0}^{N_s-1} \mathbf{C}^{s \times 2}$, where $\bigotimes$ denotes the Kronecker product. Because the S-box layer acts independently on the N_s cells, updating the current factored representation through this layer is exact and introduces no additional approximation. The optimized S-box update of [14] costs $\mathcal{O}(N_s \cdot 2^{3w})$ operations.

The linear layer acts on state pairs as $\mathbf{L}^{\times 2}(x, \theta) = (Lx, L\theta)$. Its action on correlations sends the mask pair (v_0, v_1) to $(L^\top v_0, L^\top v_1)$. The linear layer generally mixes masks across cells. After this step, RB retains the coordinates supported on each cell and uses their product for masks supported on several cells. This is the only approximation made between rounds. Fixed round constants are handled by sign changes: XORing a constant $a \in \mathbb{F}_2^n$ into the state multiplies coordinate (v_0, v_1) by $(-1)^{v_0^\top a}$.

Selected output correlations can also be evaluated by propagating the last factored RB state through the remaining layers without further factorization. We write σ_r for the resulting implicit estimate of $\mathbf{CV}(\mathbb{C})$. The stored factored state has $\mathcal{O}(N_s \cdot 2^{2w})$ entries; the cost of a final joint query depends on the masks involved. For any requested output mask λ , the RB estimate of $\mathcal{E}_{\delta, \lambda}$ is $\sigma_r[(0, \lambda)]$. For our purposes, a single RB run can therefore supply the correlations for every λ in a selected mask subspace W . Section 3.1 converts these correlations into the joint probabilities of the affine output cosets.

2.4 Preimage attacks with DL approximations

Niu *et al.* showed how to turn a collection of highly biased DL approximations into a non-trivial speed-up of the brute-force preimage search. Their filtering framework is the starting point of our work, so we recall it briefly.

Basic idea. Suppose that (δ, λ) is a deterministic DL approximation of F satisfying $\lambda^\top (F(x) \oplus F(x \oplus \delta)) = 0$ for every x . For a target $O \in \mathbb{F}_2^n$, the naive preimage search evaluates F on each guess x and on each translate $x \oplus \delta$. The observation of [22] is that the shift $x \oplus \delta$ can be a preimage of O only if $\lambda^\top (F(x) \oplus O) = 0$; otherwise the evaluation of $F(x \oplus \delta)$ can be skipped. Each pair of guesses then costs about 1.5 evaluations of F on average when this test accepts half the base outputs. Several DL approximations can provide stronger filters.

General framework. Let $\mathbb{D} = \{\delta_0, \dots, \delta_{s-1}\} \subseteq \mathbb{F}_2^m$ be s distinct nonzero input differences. For each δ_i , let $A_i = \{\lambda_{i,0}, \dots, \lambda_{i,k_i-1}\}$ be a set of k_i linearly independent masks in $\mathbb{F}_2^n$ such that each $(\delta_i, \lambda_{i,j})$ is a DL approximation of F with

correlation $\mathcal{E}_{i,j} = \mathcal{E}_{\delta_i, \lambda_{i,j}} \neq 0$. The algorithm proceeds as follows. In each of N loops, draw $x \in \mathbb{F}_2^n$ at random, compute $y = F(x)$ and return if $y = O$; otherwise, for every δ_i , run a statistical test $\text{PreTest}(y, O, \delta_i, \Lambda_i)$ on the bits $\lambda_{i,j}^\top(y \oplus O)$, which returns 0 on acceptance and 1 on rejection, and only evaluate $F(x \oplus \delta_i)$ on the translates that survive the test. Writing

$$\mathcal{T}_{x,\mathbb{D}} = \{x \oplus \delta_i : \delta_i \in \mathbb{D}, \text{PreTest}(y, O, \delta_i, \Lambda_i) = 0\}$$

for the set of *translation survivors*, evaluating all survivors in N iterations costs $N(1 + \mathbb{E}_x |\mathcal{T}_{x,\mathbb{D}}|)$ evaluations of F on average, ignoring early termination.

Strictest strategy. The strictest strategy of Niu *et al.* [22] accepts a translate $x \oplus \delta_i$ iff $\lambda_{i,j}^\top(y \oplus O) = \zeta_{\mathcal{E}_{i,j}}$ for *every* j , where $\zeta_{\mathcal{E}} = 0$ for $\mathcal{E} > 0$ and $\zeta_{\mathcal{E}} = 1$ for $\mathcal{E} < 0$. Because the k_i masks are linearly independent, these conditions specify one output coset whose acceptance probability under the uniform-null model is $q_i = 2^{-k_i}$. Consequently, $\mathbb{E}_x |\mathcal{T}_{x,\mathbb{D}}| \approx \sum_{i=0}^{s-1} 2^{-k_i}$ and the total cost is

$$T \approx N \left(1 + \sum_{i=0}^{s-1} 2^{-k_i} \right). \quad (2)$$

They estimate the probability that a shifted preimage passes its filter by

$$p_i = \prod_{j=0}^{k_i-1} \left(\frac{1}{2} + \frac{|\mathcal{E}_{i,j}|}{2} \right).$$

With preimage density 2^{-n} and negligible probability of multiple hits within one iteration, the estimated single-iteration success probability is $2^{-n}(1 + \sum_i p_i)$. They choose $N \approx 2^n / (1 + \sum_i p_i)$ to obtain success probability $1 - e^{-1} \approx 0.63$, matching the generic search budget of 2^n evaluations in the same model.

Limitations and extensions. The product estimate for p_i above assumes independence among the DL events for each input difference. Moreover, the sampling analysis in [22, Section 3.1] assumes that candidate sets from different iterations are disjoint. Section 3 replaces the product estimate with joint affine-coset probabilities derived from correlations over complete mask subspaces. It also extends the filtering framework to multiple targets and analyzes independent uniform sampling without requiring candidate sets from different iterations to be disjoint. As an example, Section 4.2 shows how independent sampling supports KNOT-Hash attack parameters excluded by the disjointness requirement.

3 A Truncated Differential Framework for Preimage Attacks

In this section, we develop the TD-based preimage framework. Section 3.1 computes joint TD probabilities from DL correlations over a complete mask subspace, so that a filter can accept a union of output cosets while accounting for

the dependence between the individual DL events. Section 3.2 extends filtering from a single target to a set of targets, combining their acceptance sets in preprocessing so that one input evaluation serves all targets and each accepted shifted candidate requires only one confirmation. Section 3.3 analyzes independent uniform sampling of the query inputs, which gives the success probability after repeated iterations and removes the cross-iteration disjointness requirement of [22, Section 3.1]. It also derives the confirmation cost and the gain, which guide the selection of acceptance cosets in Section 3.4.

3.1 Constructing Joint TD Filters

We keep the notation of Section 2.2: $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$, a fixed nonzero input difference $\delta \in \mathbb{F}_2^m$, a mask subspace $W \subseteq \mathbb{F}_2^n$ of dimension k , and $U = W^\perp$. All probabilities in this subsection are over a uniformly sampled $x \in \mathbb{F}_2^m$.

Joint coset probabilities. For a representative $u \in \mathbb{F}_2^n$, the coset $c = u + U$ specifies one joint outcome of the k basis-mask tests. The quotient $\mathbb{F}_2^n/U$ is the set of these 2^k cosets. Define the probability of a coset c by

$$p_\delta(c) = \Pr_x[F(x) \oplus F(x \oplus \delta) \in c] = \Pr[\delta \xrightarrow{F} c].$$

The inverse Walsh transform on W gives the joint coset probabilities [6]:

$$p_\delta(u + U) = \frac{1}{2^k} \sum_{\lambda \in W} (-1)^{\lambda^\top u} \mathcal{E}_{\delta, \lambda}. \quad (3)$$

The right-hand side is independent of the representative u , since $\lambda^\top v = 0$ for every $\lambda \in W$ and $v \in U$. Equation (3) uses the correlations for all linear combinations of the basis masks. In the applications, we evaluate this identity with the estimated correlations: for an r -round function, the RB method of Section 2.3 produces the correlation vector σ_r , and $\mathcal{E}_{\delta, \lambda}$ is estimated by $\sigma_r[(0, \lambda)]$, whose index uses the zero mask on $F(x)$ and the mask λ on the output difference. The only approximation comes from the RB correlations, since Equation (3) is an identity; the selected TD probabilities are further refined experimentally.

Acceptance sets. Choose a set $\mathcal{C}_\delta \subseteq \mathbb{F}_2^n/U$ of output cosets and accept their union $\mathcal{A}_\delta = \bigcup_{c \in \mathcal{C}_\delta} c$. As $U = W^\perp$, the space $\mathbb{F}_2^n$ has $|W| = 2^k$ cosets of U , each containing 2^{n-k} elements. Since distinct cosets are disjoint, the TD probability of $\mathcal{A}_\delta$ and its acceptance rate for a uniform output difference are

$$\Pr_x[F(x) \oplus F(x \oplus \delta) \in \mathcal{A}_\delta] = \sum_{c \in \mathcal{C}_\delta} p_\delta(c), \quad \frac{|\mathcal{A}_\delta|}{2^n} = \frac{|\mathcal{C}_\delta|}{|W|}.$$

The strictest filter in [22] accepts one joint outcome of the selected DL conditions and estimates its probability as the product of the individual ones, assuming independence among the DL events. Our filter may therefore accept any set of cosets, which Section 3.4 selects using the resulting attack gain.

3.2 Multi-Target Preimage Filtering

Let $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$, and $\mathbb{S} = \{T_0, \dots, T_{K-1}\} \subseteq \mathbb{F}_2^n$ be a nonempty set of K distinct targets. The goal is to find $P \in \mathbb{F}_2^m$ such that $F(P) \in \mathbb{S}$. This extends the single-target framework of [22]. In the SKINNY-Hash application of Section 6, the targets are intermediate capacity states reached by the original message.

Differences and filters. Let $\mathbb{D} = \{\delta_0, \dots, \delta_{s-1}\} \subseteq \mathbb{F}_2^m$ be a set of s distinct nonzero input differences. For each $i \in \{0, \dots, s-1\}$, choose a mask subspace $W_i \subseteq \mathbb{F}_2^n$ of dimension k_i and put $U_i = W_i^\perp$. Select $l_i \in \{0, \dots, 2^{k_i}\}$ distinct cosets of U_i with representatives $\mathbb{M}_i = \{v_{i,0}, \dots, v_{i,l_i-1}\} \subseteq \mathbb{F}_2^n$. Their union is

$$\mathcal{A}_i = \mathbb{M}_i + U_i = \bigcup_{j=0}^{l_i-1} (v_{i,j} + U_i).$$

For $0 \leq j < l_i$, write $p_{i,j} = p_{\delta_i}(v_{i,j} + U_i)$ for the coset probability computed in Section 3.1. The outputs accepted against at least one target form

$$\Gamma_i = \bigcup_{T \in \mathbb{S}} (T \oplus \mathcal{A}_i). \quad (4)$$

Preprocessing. Choose a basis $\lambda_{i,0}, \dots, \lambda_{i,k_i-1}$ of W_i and define the linear map $\pi_i : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^{k_i}$ by $\pi_i(y) = (\lambda_{i,0}^\top y, \dots, \lambda_{i,k_i-1}^\top y)$. Its kernel is U_i , so $y \in \Gamma_i$ is equivalent to

$$\pi_i(y) \in \{\pi_i(T_a \oplus v_{i,j}) : 0 \leq a < K, 0 \leq j < l_i\}.$$

We store these projected values in a hash table $\mathcal{H}_i$, removing duplicates, and store the targets in $\mathcal{H}_\mathbb{S}$. A lookup in $\mathcal{H}_i$ tests all targets for the difference δ_i without enumerating the elements of U_i .

Shared query and confirmation. An iteration samples $x \in \mathbb{F}_2^m$ independently and uniformly, evaluates $y = F(x)$, and first checks whether $y \in \mathbb{S}$. If this direct test fails, it tests $\pi_i(y) \in \mathcal{H}_i$ for each difference. When $F(x \oplus \delta_i) = T_a$ for some $0 \leq a < K$, the tested value $y \oplus T_a$ equals $F(x) \oplus F(x \oplus \delta_i)$. Thus the TD filter can retain the candidate $x \oplus \delta_i$ using the already computed y . An accepted candidate is confirmed by evaluating $F(x \oplus \delta_i)$ and checking membership in $\mathbb{S}$. One base evaluation serves all s filters and all K targets. Each difference triggers at most one confirmation, even if several targets accept it.

Algorithm 1 summarizes this procedure. The preprocessing stores at most $K(1 + \sum_{i=0}^{s-1} l_i)$ table entries. After projection, each table lookup takes expected $O(1)$ time; there are at most s filter lookups per iteration. We count the evaluations of F separately from preprocessing, projections, and table lookups in the following analysis.

3.3 Success Probability, Evaluation Cost, and Gain

Independent sampling across iterations. Fix F , $\mathbb{S}$, $\mathbb{D}$, and the filters. For an integer $N \geq 1$, let $X_1, \dots, X_N$ be independent uniform samples from $\mathbb{F}_2^m$.

Algorithm 1 Multi-target preimage filtering

Require: Function $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$, targets $\mathbb{S}$, differences $\mathbb{D}$, maps π_i , and coset representatives $\mathbb{M}_i$.

Ensure: A message P with $F(P) \in \mathbb{S}$.

Use $K = |\mathbb{S}|$, $s = |\mathbb{D}|$, and $l_i = |\mathbb{M}_i|$.

```

1:  $\mathcal{H}_{\mathbb{S}} \leftarrow$  hash table of the distinct targets
2: for  $i = 0$  to  $s - 1$  do
3:    $\mathcal{H}_i \leftarrow$  hash table of the distinct values  $\pi_i(T_a \oplus v_{i,j})$ ,  $0 \leq a < K$ ,  $0 \leq j < l_i$ 
4: repeat
5:   Sample  $x$  independently and uniformly from  $\mathbb{F}_2^m$ 
6:    $y \leftarrow F(x)$ 
7:   if  $y \in \mathcal{H}_{\mathbb{S}}$  then
8:     return  $x$ 
9:   for  $i = 0$  to  $s - 1$  do
10:    if  $\pi_i(y) \in \mathcal{H}_i$  then
11:       $y' \leftarrow F(x \oplus \delta_i)$ 
12:      if  $y' \in \mathcal{H}_{\mathbb{S}}$  then
13:        return  $x \oplus \delta_i$ 
14: until a preimage is returned

```

For $1 \leq t \leq N$, define $I_t \in \{0, 1\}$ to be 1 if the iteration started at X_t returns a preimage, and 0 otherwise. Each I_t is a deterministic function of X_t , so the indicators are independent and identically distributed. Let $q = \Pr[I_t = 1]$ be the single-iteration success probability, and let $P_{\text{suc}}(N)$ denote the probability of finding a preimage within N iterations. Then

$$P_{\text{suc}}(N) = 1 - (1 - q)^N. \quad (5)$$

This identity permits intersections between the candidate sets $\{X_t\} \cup \{X_t \oplus \delta : \delta \in \mathbb{D}\}$ from different iterations. It removes the cross-iteration disjointness condition used in [22, Section 3.1]. Let μ be the expected number of evaluations of F in one iteration, including early termination. For $q > 0$, the expected work until success is μ/q , as proved in Appendix A. For small q , a budget of $N \approx 1/q$ iterations gives success probability approximately $1 - e^{-1}$.

Success within one iteration. For a uniformly sampled $X \in \mathbb{F}_2^m$ and $0 \leq i < s$, define $A = \{F(X) \in \mathbb{S}\}$, $C_i = \{F(X \oplus \delta_i) \in \mathbb{S}\}$, $H_i = \{F(X) \in \Gamma_i\}$, $B_i = C_i \cap H_i$. The direct hit is A ; the event B_i is a shifted hit retained by its filter. The iteration succeeds on $A \cup \bigcup_{i=0}^{s-1} B_i$. Assume that the search domain contains a preimage of some target. Define the preimage density $h = \Pr[A] > 0$, the conditional filter acceptance rate $\alpha_i = \Pr[H_i \mid C_i]$, and the unconditional rate $\beta_i = \Pr[H_i]$. Translation preserves uniform sampling, so $\Pr[C_i] = h$ and $\Pr[B_i] = h\alpha_i$. Let q_0 be the sum of the individual hit probabilities and R the sum of their pairwise intersection probabilities:

$$q_0 = h \left(1 + \sum_{i=0}^{s-1} \alpha_i \right), \quad R = \sum_{i=0}^{s-1} \Pr[A \cap B_i] + \sum_{0 \leq i < j < s} \Pr[B_i \cap B_j]. \quad (6)$$

The union bound and the second Bonferroni inequality give $\max\{h, q_0 - R\} \leq q \leq \min\{1, q_0\}$, so $q \approx q_0$ when $R \ll q_0$. Appendix A proves the bounds and gives a sufficient condition for this approximation. The correction concerns multiple hits within one iteration; it does not impose disjointness across iterations.

Filter probabilities in the application model. To estimate the rates, we use $h \approx K/2^n$ and model the output-difference distribution conditioned on a shifted candidate being a preimage by the joint TD distribution from Section 3.1. The acceptance probability against the matching target and the acceptance probability for a uniform output difference are then

$$P_{\text{TP}}^{(i)} = \sum_{j=0}^{l_i-1} p_{i,j}, \quad P_{\text{FP}}^{(i)} = \frac{|\mathcal{A}_i|}{2^n} = l_i \frac{|U_i|}{2^n}.$$

For multiple targets, we model the tests against non-matching targets as independent events with acceptance probability $P_{\text{FP}}^{(i)}$, independent of the matching-target test. This target-independence model differs from the independence assumption among DL conditions avoided in Section 3.1. The target model for SKINNY-Hash is specified in Section 6. Repeated target values are stored once. The following estimates omit such repetitions in the regime $K(K-1)/2^{n+1} \ll 1$, where their probability is negligible. Conditioned on a shifted hit, its matching-target test accepts with probability $P_{\text{TP}}^{(i)}$, while the other $K-1$ tests have acceptance probability $P_{\text{FP}}^{(i)}$. Consequently, the conditional acceptance rate is estimated by

$$\alpha_i \approx 1 - (1 - P_{\text{TP}}^{(i)})(1 - P_{\text{FP}}^{(i)})^{K-1}.$$

For a generic query, all K tests use this uniform model, giving $\beta_i \approx 1 - (1 - P_{\text{FP}}^{(i)})^K$. When the hit-overlap correction is negligible, they give

$$q \approx \frac{K}{2^n} \left(1 + \sum_{i=0}^{s-1} \left[1 - (1 - P_{\text{TP}}^{(i)})(1 - P_{\text{FP}}^{(i)})^{K-1} \right] \right). \quad (7)$$

Confirmation cost. If every accepted candidate is confirmed, an iteration costs $\mu_{\text{full}} = 1 + \sum_{i=0}^{s-1} \beta_i$ evaluations of F on average. Algorithm 1 stops upon success, so $0 \leq \mu_{\text{full}} - \mu \leq sq$. Indeed, early termination saves at most s evaluations and only occurs in a successful iteration. When this correction is negligible, the per-iteration cost estimate is

$$\mu \approx 1 + \sum_{i=0}^{s-1} \left[1 - (1 - P_{\text{FP}}^{(i)})^K \right]. \quad (8)$$

One confirmation is charged for each difference whose filter accepts, even when tests against several targets accept it.

Total cost and gain. For the fixed instance, independent uniform search takes $1/h$ evaluations on average, so the exact speed-up is $q/(h\mu)$. Using the estimates

above, we define the gain estimate

$$\text{Gain} = \frac{1 + \sum_{i=0}^{s-1} \left[1 - (1 - P_{\text{TP}}^{(i)})(1 - P_{\text{FP}}^{(i)})^{K-1} \right]}{1 + \sum_{i=0}^{s-1} \left[1 - (1 - P_{\text{FP}}^{(i)})^K \right]}. \quad (9)$$

The expected attack cost, measured in evaluations of F , is then

$$\text{Cost} = \frac{\mu}{q} \approx \frac{2^n}{K \text{Gain}}.$$

For a single target ($K = 1$), this reduces to

$$\text{Gain} = \frac{1 + \sum_{i=0}^{s-1} P_{\text{TP}}^{(i)}}{1 + \sum_{i=0}^{s-1} P_{\text{FP}}^{(i)}}.$$

Section 3.4 uses this gain to select the accepted output cosets.

3.4 Selecting Acceptance Cosets

Fix $i \in \{0, \dots, s-1\}$ and consider the filter for δ_i . Write $W = W_i$, $U = U_i$, and $p(c) = p_{\delta_i}(c)$ for each coset $c \in \mathbb{F}_2^n/U$. These probabilities are given by Section 3.1. For a single target, accepting a collection $\mathcal{C} \subseteq \mathbb{F}_2^n/U$ of cosets gives the model rates $P_{\text{TP}}(\mathcal{C}) = \sum_{c \in \mathcal{C}} p(c)$ and $P_{\text{FP}}(\mathcal{C}) = |\mathcal{C}|/|W|$. The false-positive model uses uniform output differences. We count the same confirmation cost for each accepted candidate.

Optimality at a fixed acceptance budget. Order the cosets so that $p(c_1) \geq \dots \geq p(c_{|W|})$. For an integer $0 \leq l \leq |W|$, every choice of l cosets has the same false-positive rate $l/|W|$. The set $\mathcal{C}_l = \{c_1, \dots, c_l\}$ maximizes the accepted TD probability. Replacing an accepted coset by an excluded coset of larger probability increases this mass without changing the cost. Ties may give several optimal sets. For $l = 0$, use $\mathcal{C}_0 = \emptyset$.

Choosing the budget by attack gain. Hold the filters at other differences fixed, and define the sums of their true-positive and false-positive rates by

$$P_{\text{rest}} = \sum_{0 \leq a < s, a \neq i} P_{\text{TP}}^{(a)}, \quad Q_{\text{rest}} = \sum_{0 \leq a < s, a \neq i} P_{\text{FP}}^{(a)}.$$

Let G_l be the single-target gain obtained by accepting $\mathcal{C}_l$:

$$G_l = \frac{1 + P_{\text{rest}} + \sum_{j=1}^l p(c_j)}{1 + Q_{\text{rest}} + l/|W|}. \quad (10)$$

We select l to maximize G_l . For $l < |W|$, adding the next coset improves the gain precisely when $p(c_{l+1}) > G_l/|W|$. The selection therefore balances accepted TD probability against confirmation cost.

Multiple targets. For $K > 1$, we evaluate each choice using the multi-target gain in Eq. (9). Under its target model, fixing l fixes $P_{\text{FP}}^{(i)}$, and increasing $P_{\text{TP}}^{(i)}$ cannot decrease the gain. The top- l sets therefore remain optimal at a fixed l within this model, while the maximizing l can change with K . For a specified target set, the acceptance union Γ_i in Eq. (4) determines the actual rates α_i and β_i . These rates and the hit-overlap correction can instead be used in the fixed-instance analysis of Section 3.3.

4 Preimage Attacks on KNOT-Hash

We apply the framework of Section 3 in the single-target case to KNOT-Hash [30], a permutation-based sponge hash family. A member is written as KNOT-Hash (n, b, r, r') , where n is the digest length, b the state size, and r and r' the absorbing and squeezing rates, respectively, with capacities $c = b - r$ and $c' = b - r'$. Appendix B.1 recalls the specification. For the three members with $r' > r$, the construction of Appendix B.2 recovers the squeezing capacity in about $2^{c'} = 2^{n/2}$ operations and then links it to the IV in about $2^{c/2} < 2^{n/2}$, so state recovery is the dominant cost and TD filtering reduces it. The remaining KNOT-Hash (256, 384, 128, 128) is outside our scope. An earlier preimage analysis of KNOT-Hash [25] did not beat the designers' claimed bound. To our knowledge, ours are the first attacks on reduced-round KNOT-Hash below this bound. Section 4.1 presents the attacks and results, and Section 4.2 uses an attack on 11-round KNOT-Hash (256, 256, 32, 128) to support an iteration budget excluded by Niu *et al.*'s disjointness requirement [22].

4.1 Attack Construction and Results

We write $S|_{\text{rate}}$ and $S|_{\text{cap}}$ for the two parts of a state S in the phase under consideration. In the attack description, $S|_{\text{rate}}||S|_{\text{cap}}$ lists the rate part first; the detailed masks in Appendix B use the explicit bit numbering given there.

Figure 1 shows the attack on KNOT-Hash (256, 256, 32, 128). Let $p_{256}[R]$ denote the R -round permutation. For a target digest $T_0||T_1$, listed in extraction order, fix the first 128-bit block T_0 and let $F(x)$ denote the 128-bit squeezing output of $p_{256}[R](T_0||x)$, where x is a sampled 128-bit squeezing capacity. Figure 1(a) applies the TD filters of Section 3 to find a capacity whose output equals T_1 . Figure 1(b) uses the meet-in-the-middle step of Appendix B.2 to link the IV to the recovered state. It matches the 224-bit absorbing capacity and determines the 32-bit middle message block.

The TD recovery in Figure 1(a) operates on the squeezing-phase permutation. We search for truncated differentials whose input difference lies in the capacity part and whose output condition is defined on the rate part of the squeezing state. We first search with a local 4-bit input difference $\Delta \in \{4, 8, \text{C}\}$ in the leftmost S-box, with all other S-boxes inactive. We then use cyclic shifts of the selected differences and masks. Let s count all input differences after these shifts. For each difference δ_i , $0 \leq i < s$, let k_i be the number of linearly independent

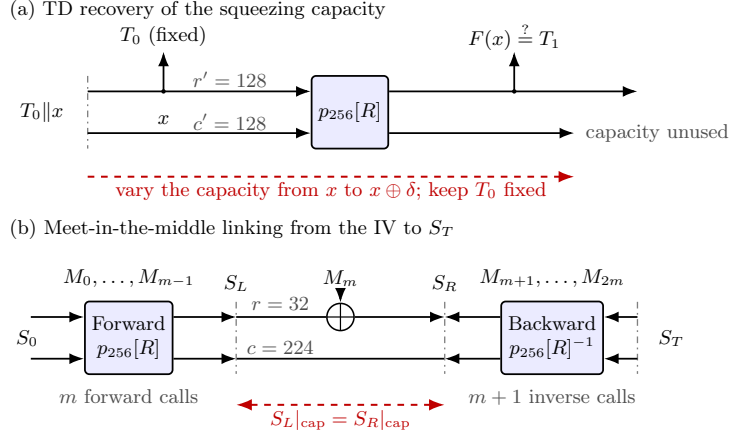


Fig. 1. Preimage attack on KNOT-Hash (256, 256, 32, 128). A capacity value x satisfying $F(x) = T_1$ gives the recovered state $S_T = T_0 || x$. In (a), each x is sampled independently and uniformly to form a candidate for S_T . The nonzero 128-bit difference δ changes only the capacity. In (b), S_0 is the IV and each side enumerates $m = \lceil (c/2)/r \rceil = 4$ of the padded message blocks $M_0, \dots, M_{2m}$. The states S_L and S_R lie immediately before and after the XOR of the middle block M_m . The backward traversal includes the inverse of the middle permutation.

output masks. They define 2^{k_i} affine output cosets, of which we retain the l_i with the highest estimated probabilities, with $P_{\text{FP}}^{(i)}$ as in Section 3.3, and we estimate $P_{\text{TP}}^{(i)}$ experimentally as the aggregate probability of the retained cosets.

Our results for the three KNOT-Hash members are summarized in Table 2, where the dominant time cost is $2^{n/2}/\text{Gain}$, and the baseline $2^{n/2}$ is the designers' claimed preimage bound [30, Table 6]. The table-based linking step of Appendix B.2 stores $\mathcal{O}(2^{c/2})$ states, but following [22, Section 7] it can instead use Floyd's cycle finding, so the attacks run in the reported times with negligible memory. For the primary member KNOT-Hash(256, 256, 32, 128), we obtain a 9-round preimage attack using three local input differences. For $\Delta = 4$, we select 12 linearly independent output masks, which define 2^{12} affine cosets, and retain 11 high-probability cosets as output truncations. For $\Delta = 8$ and $\Delta = \mathbb{C}$, we use 12 independent masks and select 6 and 19 high-probability cosets, respectively. The round-based estimates are invariant under cyclic shifts of the columns. For the three local input differences 4, 8, and $\mathbb{C}$, the reported attack uses $b/4 - 8 = 56$ cyclic shifts, giving 168 input differences. After refining the probabilities of the selected truncated differentials by experiments, we obtain $\text{Gain} = \frac{1+57.189}{1+0.492} = 2^{5.285}$. For 10 and 11 rounds, we only use the local input difference $\Delta = \mathbb{C}$ for the active S-box. In both cases, 12 linearly independent output masks are selected. We keep 107 affine cosets for 10 rounds and 206 affine cosets for 11 rounds. Since only one local input difference is used, we include all 64 cyclic shifts. The sum of measured aggregate probabilities is 18.051 for 10 rounds and 8.089

Table 2. Preimage attack results for **KNOT-Hash**. The number of input differences is $|\Delta|$ multiplied by the number of cyclic shifts, and the cyclic shifts of a local difference share its k_i masks and l_i retained cosets. Gain follows from Eq. (9), and Time is $2^{n/2}/\text{Gain}$, counted in evaluations of the reduced-round permutation.

KNOT-Hash	R	$ \Delta $	Δ	k_i	l_i	cyclic shifts	$\sum_i P_{\text{TP}}^{(i)}$	$\sum_i P_{\text{FP}}^{(i)}$	Gain	Time
KNOT-Hash (256, 256, 32, 128)	9	3	$\begin{smallmatrix} 4 \\ 8 \\ \mathbf{C} \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 12 \\ 12 \end{smallmatrix}$	$\begin{smallmatrix} 11 \\ 6 \\ 19 \end{smallmatrix}$	56	57.189	0.492	$2^{5.285}$	$2^{122.715}$
	10	1	$\mathbf{C}$	12	107	64	18.051	1.672	$2^{2.834}$	$2^{125.166}$
	11	1	$\mathbf{C}$	12	206	64	8.089	3.219	$2^{1.107}$	$2^{126.893}$
KNOT-Hash (384, 384, 48, 192)	9	3	$\begin{smallmatrix} 4 \\ 8 \\ \mathbf{C} \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 12 \\ 12 \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 7 \\ 17 \end{smallmatrix}$	88	121.446	0.773	$2^{6.109}$	$2^{185.891}$
	10	1	$\mathbf{C}$	12	69	96	28.695	1.617	$2^{3.504}$	$2^{188.496}$
	11	1	$\mathbf{C}$	12	143	96	12.151	3.352	$2^{1.596}$	$2^{190.404}$
KNOT-Hash (512, 512, 64, 256)	9	3	$\begin{smallmatrix} 4 \\ 8 \\ \mathbf{C} \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 12 \\ 12 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 1 \\ 1 \end{smallmatrix}$	120	360.000	0.088	$2^{8.374}$	$2^{247.626}$
	10	3	$\begin{smallmatrix} 4 \\ 8 \\ \mathbf{C} \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 12 \\ 12 \end{smallmatrix}$	$\begin{smallmatrix} 3 \\ 4 \\ 3 \end{smallmatrix}$	120	243.694	0.293	$2^{7.564}$	$2^{248.436}$
	11	3	$\begin{smallmatrix} 4 \\ 8 \\ \mathbf{C} \end{smallmatrix}$	$\begin{smallmatrix} 12 \\ 12 \\ 12 \end{smallmatrix}$	$\begin{smallmatrix} 6 \\ 3 \\ 34 \end{smallmatrix}$	120	73.882	1.260	$2^{5.050}$	$2^{250.950}$
	12	1	$\mathbf{C}$	12	94	128	22.618	2.938	$2^{2.585}$	$2^{253.415}$

for 11 rounds, giving gains $2^{2.834}$ and $2^{1.107}$, respectively. The same construction gives preimage attacks on 9-, 10-, and 11-round **KNOT-Hash** (384, 384, 48, 192) and on 9- to 12-round **KNOT-Hash** (512, 512, 64, 256). Detailed information on the selected output masks, differential-linear correlations, retained affine cosets, and truncated differential probabilities are given in Appendix B.

4.2 A KNOT-Hash Example Beyond Niu et al.’s Disjointness Bound

Our 11-round preimage attack on **KNOT-Hash** (256, 256, 32, 128) requires more iterations than the disjointness-based analysis of Niu *et al.* [22]. Fixing the first digest block leaves a 128-bit capacity to recover from 128 output bits, so the sampled domain and the search space have the same size. The difference set $\mathbb{D}$ consists of the 64 cyclic shifts of the local 4-bit input difference $\mathbf{C} = 1100_2$, which activates two capacity bits in one S-box column. We index the resulting differences by $0 \leq i < 64$. Each iteration therefore covers 65 candidate inputs, including the base input x .

The analysis of Niu *et al.* [22, Section 3.1] requires the candidate sets from different iterations to be disjoint. Since the capacity search space has 2^{128} elements, the number N of iterations must satisfy

$$65N \leq 2^{128}, \quad N \leq \left\lfloor \frac{2^{128}}{65} \right\rfloor \approx 2^{121.978}. \quad (11)$$

For this attack, the filters in Table 2 give $\sum_i P_{\text{TP}}^{(i)} = 8.089$. The probability model of Section 3.3 therefore gives a single-iteration success probability $q \approx q_0 = 2^{-128} \left(1 + \sum_{i=0}^{63} P_{\text{TP}}^{(i)}\right) = 9.089 \cdot 2^{-128}$. To reach success probability $1 - e^{-1} \approx 0.632$, we need

$$N \approx \frac{2^{128}}{1 + \sum_i P_{\text{TP}}^{(i)}} = \frac{2^{128}}{9.089} \approx 2^{124.816}. \quad (12)$$

This budget exceeds the upper bound in Eq. (11) by a factor of approximately 7.15 and is therefore not covered by their disjointness-based analysis.

Our analysis uses independent uniform sampling of the base inputs and remains valid when candidate sets from different iterations overlap, thereby accommodating the budget in Eq. (12), at which Eq. (5) gives the required success probability. Since $\sum_i P_{\text{FP}}^{(i)} = 64 \cdot 206/2^{12} = 3.21875$, the expected cost per iteration is approximately $1 + 3.21875$ evaluations of the reduced-round permutation. The resulting attack complexity is $(1 + 3.21875)N \approx 2^{126.893}$.

5 Preimage Attack on JH

This section applies the TD-based preimage framework of Section 3 in the single-target case to all four members of JH [27], a SHA-3 finalist built from a compression function with message feed-forward. We review previous preimage analyses in Section 5.1 and construct TDs across the two compression calls in Section 5.2. Sections 5.3 and 5.4 present attacks using single-word input differences and input-difference subspaces, respectively. Table 3 compares their results.

5.1 Previous Preimage Analyses

The preimage resistance of JH has been studied both at the generic mode level and at the reduced-round level. Mendel and Thomsen [21] and Bhattacharyya *et al.* [4] gave generic attacks on the JH-512 mode that construct correctly padded messages, with query complexities $2^{510.3}$ and 2^{507} , respectively. For reduced-round JH-512, Nourizadeh *et al.* [23] reported a 10-round preimage attack on the compression function with time complexity 2^{325} , and Li *et al.* [19] reported a 13-round meet-in-the-middle attack on a single compression call with time complexity 2^{320} and 2^{192} memory, omitting the mandatory padding block. Among the SAT-based analyses, Homsirikamol *et al.* [13] reported practical preimages for 2-round JH-256 with padding, the later 4-round result of Zaikin *et al.* [29] targets the JH-256 compression function, and Zaikin’s 5-round result [28] omits message padding.

Apart from [13], the round counts above refer to a single compression call. We apply TD filtering to the final digest of JH with the IV and padding defined by the specification, which forces the differential path to cross the two compression calls described below. The resulting attacks reach 3 rounds per compression call for JH-224/256/384 and 4 for JH-512.

5.2 Two-Block Setting and TD Construction

The four JH members share a compression function that applies the 42-round permutation E_8 to a 1024-bit chaining value H . In each compression function, the 512-bit message block is XORed into the left half before E_8 and fed forward into the right half afterward. The members differ in their initial chaining values and digest lengths $n \in \{224, 256, 384, 512\}$; the n -bit digest consists of the last n bits of the final chaining value.

For a nonempty block-aligned message, the padded input consists of $\ell \geq 1$ free message blocks $M^{(1)}, \dots, M^{(\ell)}$ followed by one fixed padding block. Let $H^{(j)}$ denote the chaining value after j blocks, with $H^{(0)}$ the specified IV. Figure 2 shows the last two compression calls. A difference $\delta \in \mathbb{F}_2^{512}$ injected into the last free block propagates through both calls before reaching the digest. For an R -round attack, each compression function uses the first R rounds of E_8 , so the differential path contains $2R$ rounds in total.

Throughout this section, we take $\ell = 1$ and use one free 512-bit message block x followed by its fixed padding block, starting from the IV specified for the corresponding member. Let $F : \mathbb{F}_2^{512} \rightarrow \mathbb{F}_2^n$ denote this two-block mapping. The message feed-forward changes the difference entering the padding call, while the filter acts on the final digest bits. We therefore construct TDs across both calls. The full specification is given in Appendix C.1.

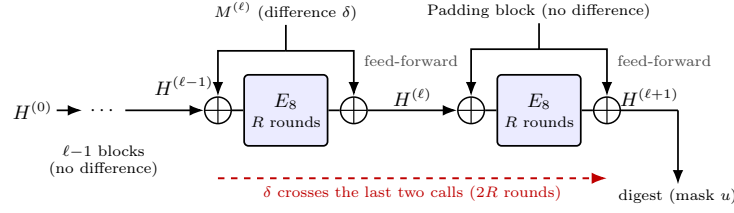


Fig. 2. The last two compression calls in the JH preimage attack. The difference enters the last free message block and crosses both calls, and $u \in \mathbb{F}_2^n$ is a mask on the digest. Our attacks set $\ell = 1$, so the abbreviated prefix is empty and $H^{(\ell-1)} = H^{(0)}$.

We search for $2R$ -round truncated differentials whose input difference δ lies in the free message block and whose output condition is defined by linear masks on the digest bits. For each candidate input difference, the RB method [14] estimates $2R$ -round DL correlations for masks supported on a single S-box at the output of the last S-box layer. When propagated backward through the final linear layer and word permutation, the digest masks used in our search are supported on one S-box for JH-512 and two S-boxes for JH-384, JH-256, and JH-224, since the JH-512 digest is the entire right half of the chaining value while the shorter digests retain only part of it. In the latter case, we approximate each correlation by the product of the two corresponding single-S-box estimates. These estimates guide candidate selection; the inverse Walsh

identity of Section 3.1 remains exact. The message feed-forward adds a known difference to the right half (δ for the free block and 0 for the padding block), which the RB method propagates together with the two permutations. The effect of the feed-forward on the state values is ignored for estimation. We validate the resulting TDs on the real two-block function F . The probability-1 conditions used below also follow from the structural verification (Appendix C.2).

5.3 Preimage Attacks with Single-Word Differences

In the word representation of E_8 , message injection changes the two most significant bits of each word, so a single active word admits the three nonzero local differences 4, 8, and C. With the 256 word positions this gives $s = 768$ input differences, each with k_i linearly independent output masks.

For JH-512 at $R = 4$ ($2R = 8$ rounds), 384 of the 768 input differences have $k_i = 60$ and the other 384 have $k_i = 64$, all masks having absolute correlation 1. The resulting affine truncated differentials have probability 1, so a single retained coset suffices and $P_{\text{FP}}^{(i)} = 2^{-k_i}$. By Eq. (9), the gain is $\text{Gain} = \frac{1+768}{1+(384 \cdot 2^{-60} + 384 \cdot 2^{-64})} \approx 2^{9.587}$. At $R = 4$, the shorter digests admit too few output masks and our search finds no truncated differentials with significant gain for JH-384, JH-256, and JH-224. We therefore report the result at $R = 3$ for all three members. Each of the same 768 input differences again admits a deterministic subspace of output masks with absolute correlation 1, whose dimension k_i ranges over 276 to 288 for JH-384, 184 to 192 for JH-256, and 152 to 176 for JH-224. In every case $\sum_i P_{\text{TP}}^{(i)} = 768$ while $\sum_i P_{\text{FP}}^{(i)}$ is negligible, so the gain is close to $1 + s = 769 = 2^{9.587}$ for all four members. These results are summarized in the *Input (single-word active)* columns of Table 3, where 2^n is the designers' claimed preimage bound.

With $s = 768$ input differences, the numerator $1 + \sum_i P_{\text{TP}}^{(i)}$ is at most 769, so the gain is bounded by 769. We next increase the number of candidates covered by one filter using an input-difference subspace.

Table 3. Preimage attack results for JH. Here R is the number of rounds in each compression call, (t, k) the input-difference and output-mask subspace dimensions, and Time is $2^n / \text{Gain}$ in evaluations of F .

Target	R	s	Input (single-word active)				Input subspace		
			$\sum_i P_{\text{TP}}^{(i)}$	$\sum_i P_{\text{FP}}^{(i)}$	Gain	Time	(t, k)	Gain	Time
JH-512	4	768	768	3.54×10^{-16}	$2^{9.587}$	$2^{502.413}$	(54, 64)	$2^{53.999}$	$2^{458.001}$
JH-384	3	768	768	4.74×10^{-81}	$2^{9.587}$	$2^{374.413}$	(158, 168)	$2^{157.999}$	$2^{226.001}$
JH-256	3	768	768	2.35×10^{-53}	$2^{9.587}$	$2^{246.413}$	(118, 128)	$2^{117.999}$	$2^{138.001}$
JH-224	3	768	768	2.53×10^{-44}	$2^{9.587}$	$2^{214.413}$	(102, 112)	$2^{101.999}$	$2^{122.001}$

5.4 Improvement Using an Input-Difference Subspace

Many of the TDs found for reduced-round JH share a common output condition. We select t linearly independent input differences $\delta_0, \dots, \delta_{t-1}$ among those found above, and let $\lambda_0, \dots, \lambda_{k-1} \in \mathbb{F}_2^n$ be k linearly independent masks having DL correlation +1 for all of them.⁶ Define

$$\overline{\mathbb{D}} = \text{Span}\{\delta_0, \dots, \delta_{t-1}\}, \quad W = \text{Span}\{\lambda_0, \dots, \lambda_{k-1}\}.$$

By Lemma 1 in Appendix C.2, all $2^t - 1$ nonzero differences in $\overline{\mathbb{D}}$ satisfy the same truncated condition $W^\perp$ with probability 1.

Filtering input cosets. Given a target digest $T \in \mathbb{F}_2^n$, the identity $F(x \oplus \delta) \oplus T = (F(x) \oplus T) \oplus (F(x \oplus \delta) \oplus F(x))$ and Lemma 1 place the last term in the linear subspace $W^\perp$ for all $x \in \mathbb{F}_2^{512}$ and $\delta \in \overline{\mathbb{D}}$, so $F(x \oplus \delta) \oplus T \in W^\perp$ if and only if $F(x) \oplus T \in W^\perp$. Thus a single evaluation of $F(x)$ and the one test $F(x) \oplus T \in W^\perp$ filter all 2^t messages of the coset $x \oplus \overline{\mathbb{D}}$ at once. We enumerate distinct input cosets and evaluate their remaining $2^t - 1$ messages only when the filter passes. The coset containing a preimage always passes, so $P_{\text{TP}} = 1$ for each of the $2^t - 1$ nonzero $\delta \in \overline{\mathbb{D}}$ and no preimage is missed.

Complexity. For an input coset $C = x \oplus \overline{\mathbb{D}}$, write $F(C) = \{F(z) : z \in C\}$ for its set of distinct digest values. A uniformly chosen target belongs to this set with probability $|F(C)|/2^n$ and passes the filter with probability 2^{-k} . For the complexity estimate, we assume $|F(C)| \approx 2^t$. For a fixed target, we use a heuristic model in which distinct input cosets pass the filter with probability 2^{-k} and contain a preimage with probability 2^{t-n} , with independent hit events across cosets. Under this model, a budget of approximately 2^{n-t} distinct input cosets gives success probability approximately $1 - e^{-1}$. Ignoring early termination, the expected cost per coset is $1 + (2^t - 1)2^{-k}$ evaluations of F , and we choose $t < k$ to keep the confirmation cost low. Relative to the 2^n generic-search budget, the resulting gain estimate is

$$\text{Gain} \approx \frac{1 + (2^t - 1)}{1 + (2^t - 1)2^{-k}} = \frac{2^t}{1 + (2^t - 1)2^{-k}}. \quad (13)$$

This gain estimate agrees with Eq. (9) for a single target when the difference set consists of all $2^t - 1$ nonzero elements of $\overline{\mathbb{D}}$, each with $P_{\text{TP}} = 1$ and uniform-null acceptance probability $P_{\text{FP}} = 2^{-k}$. Coset representatives can be enumerated using a complement of $\overline{\mathbb{D}}$, and each coset can be enumerated using a basis of $\overline{\mathbb{D}}$, so the attack requires negligible memory. The resulting (t, k) , gain, and time complexity for all four members are listed in the *Input subspace* columns of Table 3; for instance, 4-round JH-512 with $(t, k) = (54, 64)$ attains gain $2^{53.999}$, hence time complexity $2^{458.001}$. For JH-512, this budget covers all 2^{512-t} input cosets in the one-block message domain. Its success probability is approximately $1 - e^{-1}$ under the stated model.

⁶ Here k counts the independent masks used by the filter and need not be maximal. Additional common masks that arise when restricting the input subspace are deliberately left unused.

6 Second Preimage Attack on SKINNY-Hash

The inversion-based linking step used for KNOT-Hash in Section 4.1 does not transfer to SKINNY-Hash because its state transition $F_b : \mathbb{F}_2^b \rightarrow \mathbb{F}_2^b$ on the b -bit state is non-invertible. Moreover, the SKINNY-Hash designers single out second preimage as the security target in the parameter table of the specification [2]. We therefore apply the framework of Section 3 in the multi-target case to second preimage attacks that link a new message prefix to an absorption state of the target message. Section 6.1 describes this generic construction. Section 6.2 applies TD filtering to the linking step and analyzes its complexity. Section 6.3 presents the results for SKINNY-tk3-Hash. Appendix D treats SKINNY-tk2-Hash by the same template, and Appendix D.1 recalls the SKINNY-Hash specification.

6.1 Generic Second Preimage Construction

In the second preimage setting, the adversary is given a target message M together with its digest $H(M)$, and aims to find a distinct $M' \neq M$ with $H(M') = H(M)$. We describe the construction for SKINNY-tk3-Hash, whose state size is $b = 384$. Figure 3 summarizes the overall structure: the top sponge is the absorption of M under F_b , the middle box collects the intermediate capacity states that the adversary tries to hit, and the bottom sponge is the candidate prefix built by the adversary.

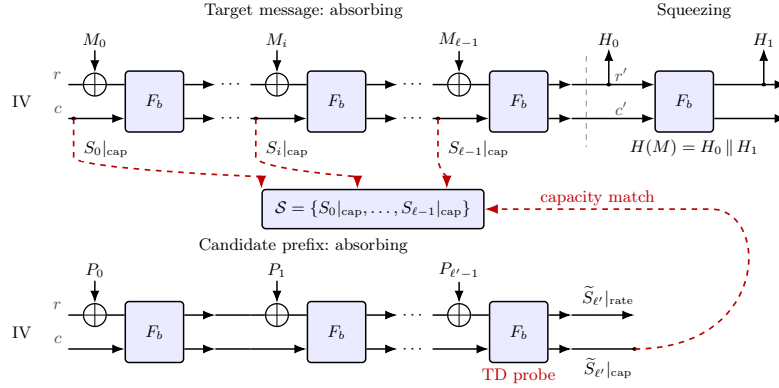


Fig. 3. Multi-target second preimage attack on a sponge hash. The squeezing rate and capacity are $r' = 128$ and $c' = b - r'$. The upper path records $S_i|_{\text{cap}}$ before absorbing each block M_i . The dashed capacity-match link tests whether the prefix $P_0 \parallel \dots \parallel P_{\ell'-1}$ has capacity output $\tilde{S}_{\ell'}|_{\text{cap}} \in \mathcal{S}$.

The target side (top sponge of Figure 3). Let r and $c = b - r$ denote the absorbing rate and capacity. We write $S|_{\text{rate}}$ and $S|_{\text{cap}}$ for the corresponding parts of a state S . The padding rule pad10^* appends a 1 bit followed by the

minimum number of zeros needed to reach a multiple of r bits. The padded target message M is parsed into ℓ blocks of r bits,

$$\text{pad10}^*(M) = M_0 \parallel M_1 \parallel \cdots \parallel M_{\ell-1}, \quad |M_i| = r,$$

which drive the absorbing phase from the IV S_0 through the state sequence

$$S_{i+1} = F_b(S_i \oplus (M_i \parallel 0^c)), \quad i = 0, 1, \dots, \ell - 1.$$

After absorbing, the squeezing phase emits the 256-bit digest $H(M)$.

Profiling: building $\mathcal{S}$ (middle box of Figure 3). The adversary first absorbs M once and records the capacity before each target block is absorbed,

$$\mathcal{S} = \{ S_i|_{\text{cap}} : 0 \leq i \leq \ell - 1 \}, \quad K = |\mathcal{S}| \leq \ell,$$

together with a corresponding index and rate part for each distinct capacity value. This costs ℓ evaluations of F_b and $\mathcal{O}(\ell)$ stored states. The estimates below omit repeated capacity values in the target sequence.

Linking: hitting $\mathcal{S}$ (bottom sponge of Figure 3). The adversary samples random candidate prefixes $P = P_0 \parallel P_1 \parallel \cdots \parallel P_{\ell'-1}$ of length $\ell' \geq \ell$ blocks⁷ and absorbs them from the same IV:

$$\tilde{S}_{i+1} = F_b(\tilde{S}_i \oplus (P_i \parallel 0^c)), \quad i = 0, 1, \dots, \ell' - 1, \quad \tilde{S}_0 = S_0.$$

A hit occurs when the final capacity $\tilde{S}_{\ell'}|_{\text{cap}}$ lies in $\mathcal{S}$. Under the uniform-capacity model, a candidate prefix hits one of the K distinct targets with probability $K/2^c$, giving an estimated $2^c/K$ trials per hit. This model treats the noninitial capacities $S_1|_{\text{cap}}, \dots, S_{\ell-1}|_{\text{cap}}$ as independent uniform values, independent of the current query, while the specified initial capacity $S_0|_{\text{cap}}$ is a fixed target; the gain estimates of Section 3.3 average over this model.

Second-preimage assembly after a capacity match. Suppose the hit lies at position j , i.e. $\tilde{S}_{\ell'}|_{\text{cap}} = S_j|_{\text{cap}}$ for some $0 \leq j \leq \ell - 1$. Define the rate-correction block

$$\widehat{M}_j = M_j \oplus \tilde{S}_{\ell'}|_{\text{rate}} \oplus S_j|_{\text{rate}}.$$

A direct check yields

$$F_b(\tilde{S}_{\ell'} \oplus (\widehat{M}_j \parallel 0^c)) = F_b(S_j \oplus (M_j \parallel 0^c)) = S_{j+1},$$

so the adversary's chain merges into the target chain at S_{j+1} , and both messages share the remaining absorbing blocks $M_{j+1}, \dots, M_{\ell-1}$ and the entire squeezing phase. The resulting padded block string is

$$B' = P_0 \parallel \cdots \parallel P_{\ell'-1} \parallel \widehat{M}_j \parallel M_{j+1} \parallel \cdots \parallel M_{\ell-1}.$$

⁷ This condition ensures that the constructed second preimage is longer than the given message and hence distinct from it.

If $j < \ell - 1$, the final padded block of the target message is unchanged. If $j = \ell - 1$, accept the hit only when $\widehat{M}_j \neq 0^r$; otherwise continue the search. The raw second preimage M' is obtained from B' by removing its last 1 bit and the zeros after it. Then $\text{pad10}^*(M') = B'$ and $H(M') = H(M)$. The condition $\ell' \geq \ell$ makes B' at least $\ell + 1$ blocks long, so $|M'| \geq \ell r > |M|$. Appendix D.2 quantifies the omitted padding-rejection probability in the cost model.

6.2 Multi-Target TD Filtering

The multi-target TD framework of Section 3 accelerates the linking step of Section 6.1. The filter acts on the last call to F_b , labelled “TD probe” on the candidate-prefix path in Figure 3. For a fixed state before this call, the filtered function maps the last message block to the capacity part of the output state. The fixed last-block domain has only 2^r inputs, so the search also refreshes the preceding prefix. Each batch uses a fresh $(\ell' - 1)$ -block prefix and B independently sampled last blocks. Appendix D.2 gives the batching algorithm, its success analysis, and the cost of generating prefixes. The set $\mathcal{S}$ plays the role of the target set $\mathbb{S}$, with $|\mathbb{S}| = K$. The TD search is constrained by the sponge geometry: input differences take the form $(\delta_r \| 0^c)$ with $\delta_r \in \mathbb{F}_2^r$, supported on the rate part the adversary controls, and output conditions act on the capacity part, since hits are tested in capacity.

Target selection in the round-reduced setting. The generic construction counts all ℓ recorded capacities. For the round-reduced SKINNY-Hash instances analyzed in this paper, we exclude the initial capacity $S_0|_{\text{cap}}$: the intermediate capacities $S_i|_{\text{cap}}$ for $i \geq 1$ lie in the image of the capacity projection of F_b , whereas the specified initial capacity $S_0|_{\text{cap}}$ need not be. All estimates below therefore use $K \leq \ell - 1$ targets, so instantiated attacks take $\ell \geq 2$.

Complexity. The reported cost counts calls to F_b for the TD probes and confirmations in the linking step. Ignoring the batching and padding corrections quantified in Appendix D.2, the uniform model gives the baseline estimate

$$T_{2\text{-pre}}(K) = \mathcal{O}\left(\frac{2^c}{K}\right), \quad 0 < K \leq \ell - 1,$$

that is, a $(c - \log_2 K)$ -bit cost. The specification states the corresponding second preimage bound as $(c - \log_2 \ell)$ bits [2]. The formal security claim is the more conservative half-capacity bound $2^{c/2}$. Eq. (9) gives the leading linking-cost estimate $2^c/(K \text{ Gain})$ with TD filtering.

6.3 Results on SKINNY-tk3-Hash

We present second-preimage attacks on 5, 6, and 7 rounds of the underlying tweakable block cipher in SKINNY-tk3-Hash. Table 4 summarizes the results.

The TD uses input differences in the 128-bit absorbing rate of F_{384} . Its output masks lie on the second of its three 128-bit branches, which is wholly contained

in the capacity. At $R = 5$ and $R = 6$ all s truncated differentials have probability 1, so $\sum_i P_{\text{TP}}^{(i)} = s$; the 6-round attack uses $s = 2040$ input differences with $k = 10$ linearly independent output masks each, giving $\text{Gain} = \frac{1+2040}{1+2040 \cdot 2^{-10}} \approx 2^{9.414}$ for $K = 1$. At $R = 7$, our search found no probability-1 truncated differential, so $\sum_i P_{\text{TP}}^{(i)} \ll s$ and the gain drops to $2^{1.783}$ for $K = 1$. All truncated-differential probabilities are experimentally validated with 2^{20} samples.

Figure 4 plots the model gain of the 6-round attack as K grows. The curve evaluates Eq. (9) with the common rates $(\sum_i P_{\text{TP}}^{(i)})/s$ and $(\sum_i P_{\text{FP}}^{(i)})/s$, which equal 1 and 2^{-10} for this deterministic instance. The attack still gives more than one bit of speedup for $K \leq 708$.

Table 4. Second preimage attack results for **SKINNY-tk3-Hash** at $K = 1$. Here R is the number of rounds of the underlying tweakable block cipher, s is the number of input differences, k is an upper bound on the number of linearly independent output masks per filter, and Time is $2^c/(K \text{ Gain})$ with $c = 256$, counted in evaluations of F_{384} .

R	s	k	$\sum_i P_{\text{TP}}^{(i)}$	$\sum_i P_{\text{FP}}^{(i)}$	Gain	Time
5	4080	14	4080	0.249	$2^{11.674}$	$2^{244.326}$
6	2040	10	2040	1.992	$2^{9.414}$	$2^{246.586}$
7	205	10	9.806	2.141	$2^{1.783}$	$2^{254.217}$

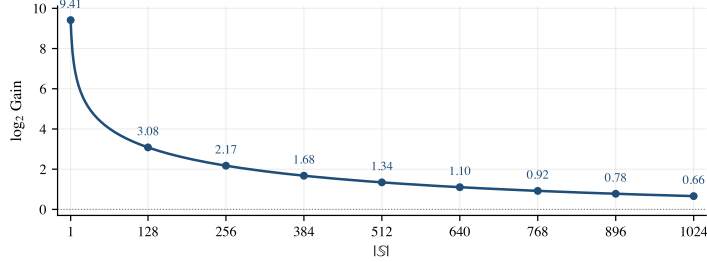


Fig. 4. Model gain of the 6-round second preimage attack on **SKINNY-tk3-Hash** as the number of targets $K = |\mathbb{S}|$ grows, under the homogeneous-rate approximation.

7 Truncated Differential Distinguishers for Xoodyak

The TD construction also gives distinguishers. On 6-round **Xoodyak** [9], we obtain an estimated sample complexity of 2^{70} pairs in the related-key setting. This is about 25 bits below the sample complexity we estimate for the prior distinguisher of Che and Tian [7] ($\approx 2^{95}$).

7.1 Xoodyak and its cryptanalysis

Xoodyak [9] is one of the ten finalists of the NIST LWC competition. It is a cryptographic primitive for hashing, authenticated encryption, and MAC computation, built around the 384-bit permutation Xoodoo [8].

State and round function. The Xoodoo state is arranged as a $4 \times 3 \times 32$ cube with entries $S[x][y][z]$. For fixed x, z the three bits $S[x][\cdot][z]$ form a *column*, and for fixed y the 128 bits $S[\cdot][y][\cdot]$ form a *plane*. Coordinates x, y , and z are taken modulo 4, 3, and 32, respectively. The input and output states of round r are denoted $S^{(r-1)}$ and $S^{(r)}$. Each step below updates its affected bits simultaneously from the state at the beginning of that step. Round r applies $\rho_{\text{east}} \circ \chi \circ \iota_r \circ \rho_{\text{west}} \circ \theta$. For the θ step, write $P[x][z] = \bigoplus_{y=0}^2 S[x][y][z]$ for the column parity.

$$\begin{aligned} \theta : \quad & S[x][y][z] = S[x][y][z] \oplus P[x-1][z-5] \oplus P[x-1][z-14], \\ \rho_{\text{west}} : \quad & S[x][1][z] = S[x-1][1][z], \quad S[x][2][z] = S[x][2][z-11], \\ \iota_r : \quad & S[0][0][z] = S[0][0][z] \oplus RC_r[z], \\ \chi : \quad & S[x][y][z] = S[x][y][z] \oplus (S[x][y+1][z] \oplus 1) \cdot S[x][y+2][z], \\ \rho_{\text{east}} : \quad & S[x][1][z] = S[x][1][z-1], \quad S[x][2][z] = S[x-2][2][z-8]. \end{aligned}$$

Here RC_r is the 32-bit round constant, with bit $RC_r[z]$. The constants are given in [8]. The only nonlinear step is χ , a quadratic map.

Keyed initialization. We use the keyed initialization considered in the DL and HDL analyses of [11, 15, 7]. Here the 128 bits $S^{(0)}[x][0][z]$ are initialized by a 128-bit key and the 256 bits $S^{(0)}[x][y][z]$ with $1 \leq y \leq 2$ by a 256-bit nonce. After one full Xoodoo invocation the first 192 bits of the resulting state are revealed and XORed into the first plaintext block.

Prior distinguishers. The prior initialization distinguishers compared here use DL or higher-order DL (HDL) approximations. For these results, a bias ε means that the tested parity is zero with probability $1/2 + \varepsilon$; its correlation is 2ε . Dunkelman and Weizman [11] gave the first 4-round DL distinguisher in the single-key setting (experimental bias $2^{-9.7}$) and a 5-round DL distinguisher in the related-key setting (experimental bias $2^{-9.36}$). Hu *et al.* [15] introduced the higher-order algebraic transitional form (HATF) to derive these biases theoretically and reached a 2nd-order HDL distinguisher for 5-round Xoodyak with bias 2^{-45} in the single-key setting. Che and Tian [7] extended the analysis to six rounds via an enhanced differential algebraic transitional form (DATF), obtaining a 2nd-order HDL distinguisher for 6-round Xoodyak with theoretical bias $\varepsilon = 2^{-48.47}$ in the related-key setting. The capacity of this binary statistic relative to uniform is $(2\varepsilon)^2$. Using the same inverse-capacity estimate as below gives $(2\varepsilon)^{-2} = 2^{94.94} \approx 2^{95}$ samples, up to constants depending on the chosen error probabilities. Each sample uses four related inputs with fixed second-order directions. This is the 6-round HDL result used in our comparison.

7.2 6-round TD via DL approximations and χ^2 distinguisher

We construct the 6-round truncated differential as the concatenation of a 2-round differential prefix with a 4-round TD obtained from DL approximations through the RB method. The resulting distribution is then distinguished from uniform by the χ^2 test.

The 2-round prefix trail. Let $Q = \rho_{\text{east}} \circ \chi$ and $L_j = \iota_j \circ \rho_{\text{west}} \circ \theta$ for round j . Write $f_j = L_{j+1} \circ Q$ for $1 \leq j \leq 5$. Then the 6-round permutation is

$$F = \underbrace{Q \circ f_5 \circ f_4 \circ f_3}_{F' \text{ (4 nonlinear layers)}} \circ \underbrace{f_2 \circ f_1 \circ L_1}_{G \text{ (2 nonlinear layers)}}.$$

The split is immediately before the third χ layer. Thus the linear steps preceding that layer belong to the prefix G .

We use a 2-round prefix of the designers' vortex construction [8, Section 7.6]. In the notation below, $x_t[i][j][k]$ denotes the bit at coordinate (i, j, k) of the difference state x_t , where x_{-1} is the input difference of G , x_0 is the difference after L_1 , and x_1, x_2 are the differences after f_1, f_2 , respectively. Each set below lists the positions of the nonzero bits:

$$\begin{aligned} x_{-1} &\xrightarrow{L_1} \{x_0[0][1][20], x_0[1][0][22], x_0[2][1][7], x_0[2][0][8], x_0[2][2][13], x_0[3][2][14]\} \\ &\xrightarrow{f_1} \{x_1[0][2][0], x_1[1][2][1], x_1[1][1][21], x_1[1][0][22], x_1[2][0][8], x_1[3][1][8]\} \\ &\xrightarrow{f_2} \{x_2[0][1][9], x_2[1][0][22], x_2[2][0][8], x_2[2][2][19], x_2[2][1][22], x_2[3][2][20]\} \end{aligned}$$

Both χ layers preserve the displayed difference and have six active columns. After each ρ_{east} step, every active column has even parity, so θ leaves the difference unchanged. Each χ layer imposes twelve affine conditions. Expressed in the state immediately after the first χ layer, the twenty-four conditions have rank 24. Thus the trail has exact probability $p_0 = 2^{-24}$ for a uniform input state. On differences, the round constant cancels, so $x_{-1} = (\rho_{\text{west}} \circ \theta)^{-1}(x_0)$. We take x_{-1} as the input difference of F and x_2 as the intermediate input difference of F' .

For the related-key experiment, write $x_{-1} = (\delta_K, \delta_N)$ according to the key and nonce planes, and let K be the fixed secret key. For each pair, independently sample uniform $a \in \mathbb{F}_2^{128}$ and $b \in \mathbb{F}_2^{256}$, and query $(K \oplus a, b)$ and $(K \oplus a \oplus \delta_K, b \oplus \delta_N)$ through the XOR-related-key oracle. The first state of each pair is then uniform in $\mathbb{F}_2^{384}$ even for fixed K . For N pairs, this experiment uses at most $2N$ related-key values.

The 4-round TD via RB and inverse Walsh. Let $e_{a,b,c} \in \mathbb{F}_2^{384}$ be the unit mask at state bit $S[a][b][c]$. Define the ordered masks

$$(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (e_{0,0,1}, e_{0,1,2}, e_{1,0,1}, e_{1,1,2}), \quad W = \text{Span}\{\lambda_0, \dots, \lambda_3\}.$$

All four positions lie in the 192 observable output bits. For $\lambda \in W$, write c_λ for the RB estimate of the DL correlation $\mathcal{E}_{x_2, \lambda}$ of F' , as defined in Section 3.1. Before the final ρ_{east} , these masks involve two χ columns. We evaluate the final

χ layer jointly on these columns from the RB state after three χ layers. The zero-mask correlation is $c_0 = 1$, and the four nonzero estimates are

$$c_{\lambda_0 \oplus \lambda_3 \oplus b_1 \lambda_1 \oplus b_2 \lambda_2} = -2^{-12}, \quad (b_1, b_2) \in \mathbb{F}_2^2.$$

All remaining nonzero-mask estimates are zero.

For a uniformly sampled state $X \in \mathbb{F}_2^{384}$, let

$$Z_j = \lambda_j^\top (F'(X) \oplus F'(X \oplus x_2)), \quad Z = (Z_0, Z_1, Z_2, Z_3) \in \mathbb{F}_2^4.$$

The event $Z = z$ specifies one output coset of $W^\perp$. Its estimated probability is

$$\hat{p}_4(z) = \frac{1}{16} \sum_{a \in \mathbb{F}_2^4} (-1)^{a^\top z} c_{\sum_{j=0}^3 a_j \lambda_j}.$$

Define the capacity of this estimated distribution relative to uniform by

$$\mathcal{C}_4 = 16 \sum_{z \in \mathbb{F}_2^4} (\hat{p}_4(z) - 1/16)^2 = \sum_{\lambda \in W \setminus \{0\}} c_\lambda^2 = 4 \cdot 2^{-24} = 2^{-22}.$$

This capacity aggregates squared correlations; it is distinct from the one-bit bias ε used in the prior-work comparison.

χ^2 distinguisher on six rounds. We use the composition model in which the prefix trail of probability p_0 gives the dominant contribution and multiplies each relevant suffix correlation by p_0 . The resulting capacity estimate is

$$\mathcal{C}_6 \approx p_0^2 \mathcal{C}_4 = (2^{-24})^2 \cdot 2^{-22} = 2^{-70}.$$

For N independent pair samples with input difference x_{-1} , let O_z count how often the four observed output-difference bits equal $z \in \mathbb{F}_2^4$. Pearson's statistic is

$$T_N = \sum_{z \in \mathbb{F}_2^4} \frac{(O_z - N/16)^2}{N/16}.$$

Under the uniform null model, T_N converges to a χ^2 distribution with $16 - 1 = 15$ degrees of freedom. Under the alternative with capacity $\mathcal{C}_6$, its expectation is $15 + (N - 1)\mathcal{C}_6$. For this fixed number of bins and fixed error probabilities, the sample complexity therefore scales as $N = \mathcal{O}(1/\mathcal{C}_6) = \mathcal{O}(2^{70})$ pairs in the related-key setting. This is about 25 bits below the sample complexity we estimate for the prior distinguisher of Che and Tian [7].

8 Conclusion

Preimage filtering with the joint TD distribution of multiple DL approximations needs no independence assumption among the DL events, because the inverse Walsh transform over a complete mask subspace is an identity. For the preimage

and second-preimage applications, we use RB estimates to select TDs and experimentally validate their probabilities. The framework also shares one evaluation across multiple targets and lets candidate sets overlap between iterations.

For the three **KNOT-Hash** members, our TD-based framework yields the first preimage attacks below the designers' claimed bounds for 9 to 12 rounds. For all four **JH** members, it covers 3 or 4 rounds per compression call under the padding rule of the specification, with speedups from 2^{54} to 2^{158} . For **SKINNY-Hash** target messages containing at least two padded blocks, it gives the first second-preimage attacks on 5- and 6-round **SKINNY-tk2-Hash** and 5- to 7-round **SKINNY-tk3-Hash**. The same construction also gives a 6-round related-key distinguisher on **Xoodyak**, with estimated sample complexity about 25 bits below the previous result.

We select input differences and mask subspaces heuristically, and then choose acceptance cosets by the procedure of Section 3.4. Jointly searching all three would need, for every candidate configuration, its DL correlations and coset distribution before its gain is known.

Acknowledgments

The first, third, fourth, and fifth authors are supported by the Singapore NRF Investigatorship grant NRF-NRFI08-2022-0013. During the preparation of this manuscript, the authors used ChatGPT (OpenAI) and Claude (Anthropic) to assist with drafting, restructuring, and language editing. The authors reviewed and revised the generated material and are responsible for the accuracy, originality, and final content of the paper. The scientific results, proofs, experiments, and interpretations were checked and approved by the authors.

References

1. Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: A new tool for differential-linear cryptanalysis. In: Ishai, Y., Rijmen, V. (eds.) *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 I. pp. 313–342. *Lecture Notes in Computer Science*, Springer (2019). https://doi.org/10.1007/978-3-030-17653-2_11
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: **SKINNY-AEAD** and **SKINNY-hash**. *IACR Trans. Symmetric Cryptol.* **2020**(S1), 88–131 (2020). <https://doi.org/10.13154/TOSC.V2020.IS1.88-131>
3. Beyne, T.: A geometric approach to linear cryptanalysis. In: Tibouchi, M., Wang, H. (eds.) *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security*, Singapore, December 6-10, 2021, Proceedings, Part I. pp. 36–66. *Lecture Notes in Computer Science*, Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_2
4. Bhattacharyya, R., Mandal, A., Nandi, M.: Security analysis of the mode of **JH** hash function. In: *Fast Software Encryption – FSE 2010*. *Lecture Notes in Computer Science*, vol. 6147, pp. 168–191. Springer (2010), https://www.isical.ac.in/~rishib_t/pubs/FSE2010-146.pdf

5. Blondeau, C., Leander, G., Nyberg, K.: Differential-linear cryptanalysis revisited. In: Cid, C., Rechberger, C. (eds.) *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*. pp. 411–430. *Lecture Notes in Computer Science*, Springer (2014). https://doi.org/10.1007/978-3-662-46706-0_21
6. Blondeau, C., Leander, G., Nyberg, K.: Differential-linear cryptanalysis revisited. *J. Cryptol.* **30**(3), 859–888 (2017). <https://doi.org/10.1007/S00145-016-9237-5>
7. Che, C., Tian, T.: Enhancing the DATF technique in differential-linear cryptanalysis. In: Hanaoka, G., Yang, B. (eds.) *Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part I*. pp. 352–382. *Lecture Notes in Computer Science*, Springer (2025). https://doi.org/10.1007/978-981-95-5018-0_12
8. Daemen, J., Hoffert, S., Assche, G.V., Keer, R.V.: The design of xoodoo and xoooff. *IACR Trans. Symmetric Cryptol.* **2018**(4), 1–38 (2018). <https://doi.org/10.13154/TOSC.V2018.I4.1-38>
9. Daemen, J., Hoffert, S., Peeters, M., Assche, G.V., Keer, R.V.: Xoodyak, a lightweight cryptographic scheme. *IACR Trans. Symmetric Cryptol.* **2020**(S1), 60–87 (2020). <https://doi.org/10.13154/TOSC.V2020.IS1.60-87>
10. Dong, X., Zhao, B., Qin, L., Hou, Q., Zhang, S., Wang, X.: Generic MitM attack frameworks on sponge constructions. In: Reyzin, L., Stebila, D. (eds.) *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*. pp. 3–37. *Lecture Notes in Computer Science*, Springer (2024). https://doi.org/10.1007/978-3-031-68385-5_1
11. Dunkelman, O., Weizman, A.: Differential-linear cryptanalysis on Xoodyak (2022), <https://csrc.nist.gov/Presentations/2022/differential-linear-cryptanalysis-on-xoodyak>, NIST Lightweight Cryptography Workshop 2022
12. Guo, C., Zhang, H., Lin, D.: Estimating differential-linear distinguishers and applications to CTC2. In: López, J., Wu, Y. (eds.) *Information Security Practice and Experience - 11th International Conference, ISPEC 2015, Beijing, China, May 5-8, 2015. Proceedings*. pp. 220–234. *Lecture Notes in Computer Science*, Springer (2015). https://doi.org/10.1007/978-3-319-17533-1_16
13. Homsirikamol, E., Morawiecki, P., Rogawski, M., Srebrny, M.: Security margin evaluation of SHA-3 contest finalists through SAT-based attacks. In: *Computer Information Systems and Industrial Management – CISIM 2012. Lecture Notes in Computer Science*, vol. 7564, pp. 56–67. Springer (2012). https://doi.org/10.1007/978-3-642-33260-9_4, <https://eprint.iacr.org/2012/421>
14. Hu, K., Niu, Z., Wang, M.: Round-based approximation of (higher-order) differential-linear correlation - A geometric approach perspective. In: Daemen, J., Thomé, E. (eds.) *Advances in Cryptology - EUROCRYPT 2026 - 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Rome, Italy, May 10-14, 2026, Proceedings, Part VI*. pp. 152–180. *Lecture Notes in Computer Science*, Springer (2026). https://doi.org/10.1007/978-3-032-25333-0_6
15. Hu, K., Peyrin, T., Tan, Q.Q., Yap, T.: Revisiting higher-order differential-linear attacks from an algebraic perspective. In: Guo, J., Steinfeld, R. (eds.) *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December*

- 4-8, 2023, Proceedings, Part III. pp. 405–435. Lecture Notes in Computer Science, Springer (2023). https://doi.org/10.1007/978-981-99-8727-6_14
16. Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (ed.) Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings. pp. 196–211. Lecture Notes in Computer Science, Springer (1994). https://doi.org/10.1007/3-540-60590-8_16
17. Leander, G., Tezcan, C., Wiemer, F.: Searching for subspace trails and truncated differentials. IACR Trans. Symmetric Cryptol. **2018**(1), 74–100 (2018). <https://doi.org/10.13154/TOSC.V2018.I1.74-100>
18. Li, H., He, L., Chen, S., Guo, J., Qiu, W.: Automatic preimage attack framework on ascon using a linearize-and-guess approach. IACR Trans. Symmetric Cryptol. **2023**(3), 74–100 (2023). <https://doi.org/10.46586/TOSC.V2023.I3.74-100>
19. Li, Z., He, Y., Yan, H., Zhao, L., Zhang, J.: MILP-based meet-in-the-middle preimage attack on the JH-512 hash function: breaking 13 rounds. Scientia Sinica Informationis **56**(4), 868–883 (2026). <https://doi.org/10.1360/SSI-2025-0389>, in Chinese
20. Liu, M., Lu, X., Lin, D.: Differential-linear cryptanalysis from an algebraic perspective. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III. pp. 247–277. Lecture Notes in Computer Science, Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_9
21. Mendel, F., Thomsen, S.S.: An observation on JH-512 (2008), https://ehash.is.ec.tugraz.at/uploads/d/da/Jh_preimage.pdf
22. Niu, Z., Hu, K., Sun, S., Zhang, Z., Wang, M.: Speeding up preimage and key-recovery attacks with highly biased differential-linear approximations. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV. pp. 73–104. Lecture Notes in Computer Science, Springer (2024). https://doi.org/10.1007/978-3-031-68385-5_3
23. Nourizadeh, S., Javanmardi, M., Sadeghiyan, B.: Cryptanalysis of the reduced-round version of JH. In: IST. pp. 1050–1054. IEEE (2012)
24. Qin, L., Hua, J., Dong, X., Yan, H., Wang, X.: Meet-in-the-middle preimage attacks on sponge-based hashing. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV. pp. 158–188. Lecture Notes in Computer Science, Springer (2023). https://doi.org/10.1007/978-3-031-30634-1_6
25. Rohit, R.: Preimage security of KNOT-Hash. Cryptology ePrint Archive, Paper 2019/424 (2019), <https://eprint.iacr.org/2019/424>
26. Sasaki, Y.: Memoryless unbalanced meet-in-the-middle attacks: Impossible results and applications. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8479, pp. 253–270. Springer (2014). https://doi.org/10.1007/978-3-319-07536-5_16
27. Wu, H.: The hash function JH (Jan 2011), https://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf, SHA-3 Round 3 Submission, version dated 16 January 2011
28. Zaikin, O.S.: Preimage attack on 5-round cryptographic hash function JH-256 via parallel SAT solver. Prikladnaya Diskretnaya Matematika. Supplement (18), 262–

- 264 (2025). <https://doi.org/10.17223/2226308X/18/56>, <https://www.mathnet.ru/eng/pdma728>, in Russian
29. Zaikin, O.S., Davydov, V.V., Kiryanova, A.P.: SAT-based analysis of SHA-3 competition finalists. *Numerical Methods and Programming* **25**(3), 259–273 (2024). <https://doi.org/10.26089/NumMet.v25r320>, <https://en.num-meth.ru/index.php/journal/article/view/1339>, in Russian
 30. Zhang, W., Ding, T., Yang, B., Bao, Z., Xiang, Z., Ji, F., Zhao, X.: KNOT: Algorithm specifications and supporting document (2019), <https://csrc.nist.gov/csrc/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/KNOT-spec.pdf>, submission to the NIST Lightweight Cryptography Project
 31. Zhao, B., Hou, Q., Qin, L., Dong, X.: Triangulating meet-in-the-middle attack. In: Kalai, Y.T., Kamara, S.F. (eds.) *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part V*. pp. 260–292. *Lecture Notes in Computer Science*, Springer (2025). https://doi.org/10.1007/978-3-032-01901-1_9

A Single-Iteration Hit Probability and Work

We prove the probability and cost bounds used in Section 3.3. Fix the function $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$, a nonempty target set $\mathbb{S} \subseteq \mathbb{F}_2^n$, the distinct nonzero differences $\mathbb{D} = \{\delta_0, \dots, \delta_{s-1}\} \subseteq \mathbb{F}_2^m$, and the TD filters. For each $0 \leq i < s$, let $\mathcal{A}_i$ be its acceptance set from Section 3.2 and write

$$\Gamma_i = \bigcup_{T \in \mathbb{S}} (T \oplus \mathcal{A}_i)$$

for the set of outputs that cause its filter to accept. For uniform $X \in \mathbb{F}_2^m$, define

$$\begin{aligned} A &= \{F(X) \in \mathbb{S}\}, & C_i &= \{F(X \oplus \delta_i) \in \mathbb{S}\}, \\ H_i &= \{F(X) \in \Gamma_i\}, & B_i &= C_i \cap H_i. \end{aligned}$$

Assume that the search domain contains a preimage of some target, and let

$$h = \Pr[A] > 0, \quad \alpha_i = \Pr[H_i \mid C_i], \quad \beta_i = \Pr[H_i].$$

Translation preserves the uniform distribution, so $\Pr[C_i] = \Pr[A] = h$ and $\Pr[B_i] = h\alpha_i$.

Single-iteration probability. The single-iteration success probability is $q = \Pr[A \cup \bigcup_{i=0}^{s-1} B_i]$. Set

$$q_0 = h \left(1 + \sum_{i=0}^{s-1} \alpha_i \right), \quad (14)$$

$$R = \sum_{i=0}^{s-1} \Pr[A \cap B_i] + \sum_{0 \leq i < j < s} \Pr[B_i \cap B_j]. \quad (15)$$

The union bound and the second Bonferroni inequality give $q_0 - R \leq q \leq q_0$. Since A is contained in the success event,

$$\max\{h, q_0 - R\} \leq q \leq \min\{1, q_0\}. \quad (16)$$

Thus R bounds the error caused by summing the individual hit probabilities when several candidates can be preimages.

A sufficient bound on hit overlaps. Suppose that, for some $\kappa \geq 0$ and every pair of distinct $a, b \in \{0\} \cup \mathbb{D}$,

$$\Pr[F(X \oplus a) \in \mathbb{S}, F(X \oplus b) \in \mathbb{S}] \leq \kappa h^2. \quad (17)$$

Each intersection in R is contained in a corresponding unfiltered joint hit event. There are $\binom{s+1}{2}$ such intersections, hence

$$R \leq \binom{s+1}{2} \kappa h^2, \quad \varepsilon := \frac{R}{q_0} \leq \frac{\binom{s+1}{2} \kappa h}{1 + \sum_{i=0}^{s-1} \alpha_i}. \quad (18)$$

Equation (17) bounds the joint hit probabilities for the instance under analysis. When $\varepsilon < 1$, Eq. (16) yields

$$(1 - \varepsilon)q_0 \leq q \leq q_0.$$

Combined with independent sampling and Eq. (5), a budget $N = \lceil ((1 - \varepsilon)q_0)^{-1} \rceil$ therefore gives success probability at least $1 - e^{-1}$, since $(1 - q)^N \leq e^{-qN}$.

Confirmation cost and expected work. Let μ be the expected number of evaluations of F in one iteration, including early termination. If all accepted candidates were evaluated, linearity of expectation would give the average cost $\mu_{\text{full}} = 1 + \sum_{i=0}^{s-1} \beta_i$. A failed iteration performs all these evaluations. A successful iteration saves at most s evaluations. Consequently,

$$\mu_{\text{full}} - sq \leq \mu \leq \mu_{\text{full}}. \quad (19)$$

With independent uniform inputs in successive iterations, iteration $t \geq 1$ is reached with probability $(1 - q)^{t-1}$. Its input is independent of the preceding failures, so its conditional expected cost is still μ . Summing over all iterations gives

$$\text{Cost} = \mu \sum_{t=0}^{\infty} (1 - q)^t = \frac{\mu}{q}. \quad (20)$$

With a budget of N iterations, the expected work is $\mu \sum_{t=0}^{N-1} (1 - q)^t = \mu(1 - (1 - q)^N)/q$. These costs count evaluations of F ; preprocessing and filter lookups are accounted for separately.

Independent uniform search has expected cost $1/h$ on the same instance, so the exact speed-up is $q/(h\mu)$. If $R < q_0$, Eqs. (16) and (19) give

$$\frac{q}{h\mu} \geq \frac{1 + \sum_{i=0}^{s-1} \alpha_i - R/h}{1 + \sum_{i=0}^{s-1} \beta_i}. \quad (21)$$

The ratio $(1 + \sum_{i=0}^{s-1} \alpha_i)/(1 + \sum_{i=0}^{s-1} \beta_i)$ is the resulting first-order estimate when R/q_0 and sq/μ_{full} are negligible.

B Preimage Attack Results on KNOT-Hash

B.1 The KNOT-Hash family

KNOT-Hash [30] is a permutation-based sponge hash family submitted to the NIST lightweight cryptography competition. A member is parameterized as KNOT-Hash (n, b, r, r') , where n is the digest length, b is the state size, r is the *absorbing* rate, and $r' \geq r$ is the *squeezing* rate. The absorbing capacity is $c = b - r$. All lengths are in bits. The parameter nr_h in Table 5 is the number of permutation rounds in the full specification. The specification defines four members, of which we target the three with $r' > r$ listed in Table 5.

Table 5. Parameters of the three targeted **KNOT-Hash** members. The primary member is **KNOT-Hash** (256, 256, 32, 128).

Member	n	b	c	r	r'	rounds nr_h
KNOT-Hash (256, 256, 32, 128)	256	256	224	32	128	68
KNOT-Hash (384, 384, 48, 192)	384	384	336	48	192	104
KNOT-Hash (512, 512, 64, 256)	512	512	448	64	256	140

Permutation. The state is arranged as a $4 \times (b/4)$ bit matrix. Let $p_b[R]$ denote the R -round permutation. Each round XORs a round constant into row 0, applies the same 4-bit S-box to every column (SubColumn), and cyclically shifts each row by its prescribed offset (ShiftRow). Within a column, the 4-bit value lists rows 3, 2, 1, and 0 from most to least significant. The S-box and shift offsets are specified in [30].

Sponge construction. **KNOT-Hash** uses $f = p_b[nr_h]$ with absorbing rate r and squeezing rate $r' \geq r$. The rate is taken from the least significant bits of the state. Padding appends a 1 and the minimum number of 0s needed to complete an r -bit block. The three targeted members start from the all-zero state. We list message and digest blocks in processing and extraction order, respectively. The padded message is parsed into v blocks $M_0 \parallel M_1 \parallel \dots \parallel M_{v-1}$ of r bits each. The absorbing phase XORs each M_i into the rate part and applies $p_b[nr_h]$. Let S be the current state and $S_{r'}$ its r' -bit squeezing output part. The squeezing phase then reads $H_0 \leftarrow S_{r'}$, applies $p_b[nr_h](S)$, reads $H_1 \leftarrow S_{r'}$, and continues until n/r' blocks have been produced (all four members satisfy $r' \mid n$). The digest is $H = H_0 \parallel H_1 \parallel \dots \parallel H_{n/r'-1}$, with H_i playing the role of T_i in Appendix B.2.

B.2 State Recovery and Meet-in-the-Middle Linking

The preimage attack on **KNOT-Hash** combines recovery of the state at the start of squeezing with meet-in-the-middle linking from the IV to that state. This two-stage construction is used by Niu *et al.* [22] and Dong *et al.* [10]. We recall the construction for digests spanning several output blocks; Figure 1 illustrates its application to **KNOT-Hash**.

State and digest notation. Let $f : \mathbb{F}_2^b \rightarrow \mathbb{F}_2^b$ be the underlying permutation. Write r and $c = b - r$ for the absorbing rate and capacity, and r' and $c' = b - r'$ for the squeezing rate and capacity. For either phase, $S|_{\text{rate}}$ and $S|_{\text{cap}}$ denote the corresponding parts of a state S . For this construction, assume $n > r'$ and parse the target digest as $T_0 \parallel \dots \parallel T_{\lceil n/r' \rceil - 1}$, each block of r' bits except possibly the last. In the attack notation, rate and capacity parts are listed in that order, regardless of their positions in the specification's serialized state. Let S_0 denote the specified initial state.

Recovering the squeezing state. Let S_T be the state at the start of squeezing. Its rate part is the known first digest block T_0 , so only its c' -bit capacity x remains to be recovered. For a guess x , start from $T_0 \parallel x$ and generate the remaining

output blocks using f . The target for this preimage search is $T_1 \parallel \cdots \parallel T_{\lceil n/r' \rceil - 1}$. We apply the single-target case of Algorithm 1 to this mapping, with TD filters derived from the squeezing-phase permutation. Input differences are confined to the capacity, and output conditions are imposed on the rate. A successful guess gives $S_T = T_0 \parallel x$.

Linking the IV to the recovered state. Choose m so that each side can enumerate at least $2^{c/2}$ valid block sequences. For the three targeted **KNOT-Hash** members, $m = \lceil c/(2r) \rceil = 4$ suffices. Let $M_0, \dots, M_{2m}$ denote the padded absorbing blocks. Suffixes are chosen so that their final block contains the padding bit. The middle block M_m links a forward computation from S_0 to a backward computation from S_T . The two sides are shown in Figure 1(b).

- *Forward table* $\mathbb{T}_L$. Enumerate prefixes $M_0, \dots, M_{m-1}$ and store $2^{c/2}$ resulting states S_L , immediately before the XOR of M_m .
- *Backward table* $\mathbb{T}_R$. Enumerate suffixes $M_{m+1}, \dots, M_{2m}$ and work backward from S_T . Invert the suffix calls and the middle permutation, and store $2^{c/2}$ states S_R , immediately after the XOR of M_m .

A match on all c capacity bits, $S_L|_{\text{cap}} = S_R|_{\text{cap}}$, is expected among the $2^{c/2} \times 2^{c/2}$ pairs. For a matching pair, set

$$M_m = S_L|_{\text{rate}} \oplus S_R|_{\text{rate}}.$$

Absorbing the resulting blocks from S_0 reaches S_T and produces the target digest. Removing the padding returns a preimage message. The table-based linking step uses $\mathcal{O}((m+1)2^{c/2})$ permutation evaluations and $\mathcal{O}(2^{c/2})$ stored states. Increasing both table sizes by a constant factor makes the linking success probability close to 1.

Niu *et al.* [22, Section 7] also state that the linking phase can be implemented with negligible memory using Floyd’s cycle finding, following Sasaki [26]. Figure 1(b) illustrates the table-based variant.

B.3 Preimage Attack Results on KNOT-Hash

We first use the RB method [14] to estimate DL correlations for differences active in one input S-box and masks supported on one output S-box. For each input difference, we select linearly independent output masks with high absolute correlations and consider their span. Correlations for masks supported on one output column are read from the precomputed estimates. For a mask supported on several columns, we approximate its correlation by the product of the corresponding single-column correlations. The inverse Walsh transform of the resulting correlation vector gives estimated probabilities for the output cosets. We select retained cosets using the gain criterion of Section 3.4 and refine their aggregate probabilities experimentally.

For 9-round **KNOT-Hash** (256, 256, 32, 128), the detailed results are given in Table 6. In both detailed tables, columns are indexed from the most significant side: $j = 0$ denotes the leftmost S-box and $0 \leq j < b/4$. Column 0 has local input

difference $\Delta \in \{4, 8, \mathbb{C}\}$; all other input columns have zero difference. The notation $[j, \gamma]$ specifies a mask on the output of column j before the final ShiftRow, with local mask $\gamma \in \{1, 2, 3\}$. Each table lists $k = 12$ linearly independent masks. Let $\lambda_0, \dots, \lambda_{11} \in \mathbb{F}_2^b$ be the corresponding masks after the final ShiftRow, in table order. They define 2^{12} affine output cosets. In these tables, $\omega = \omega_{11}\omega_{10} \cdots \omega_0$ denotes the 12-bit label of a coset, where for an output difference $z \in \mathbb{F}_2^b$ the bits $\omega_i \in \{0, 1\}$ specify the conditions

$$\lambda_i \cdot z = \omega_i, \quad 0 \leq i < 12.$$

Take the input difference with $\Delta = 4$ as an example. For the explicit bit conditions below, write the output difference as $z = (z_0, z_1, \dots, z_{255})$, with z_0 the most significant bit. Thus $z_{128}, \dots, z_{255}$ are the 128 squeezing-rate bits. The coset $\omega = 000010000000$ imposes:

$$\begin{aligned} z_{135} \oplus z_{200} &= 0, \\ z_{247} &= 0, \\ z_{182} \oplus z_{247} &= 0, \\ z_{159} \oplus z_{224} &= 0, \\ z_{200} &= 0, \\ z_{191} \oplus z_{192} &= 0, \\ z_{224} &= 0, \\ z_{175} \oplus z_{240} &= 1, \\ z_{174} \oplus z_{239} &= 0, \\ z_{239} &= 0, \\ z_{255} &= 0, \\ z_{190} \oplus z_{255} &= 0. \end{aligned}$$

The columns $\text{Pr}_{\text{theory}}$ and Pr_{exp} give the correlation-derived probability estimate and the directly sampled coset probability, respectively. ‘‘Corr.’’ gives the estimated correlation of each listed basis mask, whose row numbered $i + 1$ corresponds to λ_i . We use the experimentally estimated aggregate probabilities to evaluate the gain. Since 56 cyclic shifts are used for each truncated differential, the experimentally refined gain of the 9-round attack on KNOT-Hash (256, 256, 32, 128) is

$$\text{Gain} = \frac{1 + 56 \cdot (0.25125 + 0.16141 + 0.60857)}{1 + 56 \cdot (11 + 6 + 19)/2^{12}} \approx 2^{5.285}.$$

From Table 6, the aggregate correlation-derived probabilities $\text{Pr}_{\text{theory}}$ are smaller than the sampled aggregates Pr_{exp} for each of the three local differences. This discrepancy is caused by the use of the round-based approximation method in computing differential-linear correlations and by the independence assumption used when multiplying local contributions from several active S-boxes. When all

Table 6. Detailed results for 9-round **KNOT-Hash** (256, 256, 32, 128).

Input diff	Independent output masks			Selected affine cosets			
	No.	Mask $[j, \gamma]$	Corr.	No.	Coset ω	Pr_{theory}	Pr_{exp}
4	1	[8, 3]	0.572	1	000000000000	0.022	0.097
	2	[55, 1]	0.479	2	000010000000	0.010	0.021
	3	[55, 3]	0.479	3	100000000000	0.009	0.016
	4	[32, 3]	0.439	4	000000100000	0.009	0.037
	5	[8, 1]	0.432	5	010000000000	0.009	0.016
	6	[0, 3]	0.409	6	001000000000	0.008	0.011
	7	[32, 1]	0.386	7	000100000000	0.008	0.011
	8	[48, 3]	0.383	8	000001000000	0.007	0.011
	9	[47, 3]	0.367	9	000000010000	0.006	0.015
	10	[47, 1]	0.360	10	000000001000	0.006	0.008
	11	[63, 1]	0.341	11	110000000000	0.006	0.008
	12	[63, 3]	0.324	—	—	—	—
8	1	[8, 3]	0.590	1	000000000000	0.015	0.070
	2	[8, 1]	0.471	2	100000000000	0.009	0.034
	3	[0, 3]	0.414	3	001000000000	0.006	0.009
	4	[48, 3]	0.394	4	010000000000	0.006	0.014
	5	[32, 3]	0.378	5	000100000000	0.006	0.009
	6	[55, 1]	0.358	6	000000000100	0.006	0.024
	7	[32, 1]	0.351	—	—	—	—
	8	[55, 3]	0.351	—	—	—	—
	9	[47, 3]	0.320	—	—	—	—
	10	[47, 1]	0.310	—	—	—	—
	11	[48, 1]	0.282	—	—	—	—
	12	[16, 3]	0.218	—	—	—	—
C	1	[56, 3]	0.889	1	000000000000	0.143	0.361
	2	[56, 1]	0.857	2	100000000000	0.034	0.046
	3	[8, 3]	0.749	3	010000000000	0.034	0.046
	4	[0, 3]	0.714	4	001000000000	0.032	0.016
	5	[8, 1]	0.677	5	000100000000	0.024	0.013
	6	[0, 1]	0.632	6	000010000000	0.023	0.018
	7	[16, 1]	0.616	7	110000000000	0.023	0.026
	8	[16, 3]	0.605	8	000000100000	0.022	0.010
	9	[48, 3]	0.589	9	000001000000	0.022	0.018
	10	[48, 1]	0.517	10	001100000000	0.021	0.011
	11	[55, 1]	0.513	11	000000010000	0.019	0.005
	12	[55, 3]	0.513	12	000011000000	0.018	0.011
	—	—	—	13	000000001000	0.014	0.006
	—	—	—	14	000000101000	0.014	0.006
	—	—	—	15	000000000100	0.012	0.003
	—	—	—	16	000000010100	0.011	0.003
	—	—	—	17	101000000000	0.008	0.004
	—	—	—	18	011000000000	0.008	0.004
	—	—	—	19	000000000010	0.007	0.002

correlations in the mask span are computed exactly, the inverse Walsh transform gives exact coset probabilities.

For the 9-round attack on **KNOT-Hash** (512, 512, 64, 256), the conditions in Table 7 hold with probability 1. This also follows by propagating sets of possible local differences through the S-boxes and row shifts. For input difference 4 or 8 in column 0, columns 8, 17, 40, 41, 88, 118 have zero difference after the ninth S-box layer. For input difference C, the corresponding columns are 8, 17, 40, 41, 56, 88. Every listed mask therefore reads a zero difference. The estimated and experimentally measured probabilities are also 1, so the two gain estimates coincide.

Table 7. Detailed results for 9-round **KNOT-Hash** (512, 512, 64, 256).

Input diff	Independent output masks			Selected affine cosets			
	No.	Mask $[j, \gamma]$	Corr.	No.	Coset ω	Pr_{theory}	Pr_{exp}
4	1	[8, 1]	1.000	1	000000000000	1.000	1.000
	2	[8, 2]	1.000	—	—	—	—
	3	[17, 1]	1.000	—	—	—	—
	4	[17, 2]	1.000	—	—	—	—
	5	[40, 1]	1.000	—	—	—	—
	6	[40, 2]	1.000	—	—	—	—
	7	[41, 1]	1.000	—	—	—	—
	8	[41, 2]	1.000	—	—	—	—
	9	[88, 1]	1.000	—	—	—	—
	10	[88, 2]	1.000	—	—	—	—
	11	[118, 1]	1.000	—	—	—	—
	12	[118, 2]	1.000	—	—	—	—
8	1	[8, 1]	1.000	1	000000000000	1.000	1.000
	2	[8, 2]	1.000	—	—	—	—
	3	[17, 1]	1.000	—	—	—	—
	4	[17, 2]	1.000	—	—	—	—
	5	[40, 1]	1.000	—	—	—	—
	6	[40, 2]	1.000	—	—	—	—
	7	[41, 1]	1.000	—	—	—	—
	8	[41, 2]	1.000	—	—	—	—
	9	[88, 1]	1.000	—	—	—	—
	10	[88, 2]	1.000	—	—	—	—
	11	[118, 1]	1.000	—	—	—	—
	12	[118, 2]	1.000	—	—	—	—
c	1	[8, 1]	1.000	1	000000000000	1.000	1.000
	2	[8, 2]	1.000	—	—	—	—
	3	[17, 1]	1.000	—	—	—	—
	4	[17, 2]	1.000	—	—	—	—
	5	[40, 1]	1.000	—	—	—	—
	6	[40, 2]	1.000	—	—	—	—
	7	[41, 1]	1.000	—	—	—	—
	8	[41, 2]	1.000	—	—	—	—
	9	[56, 1]	1.000	—	—	—	—
	10	[56, 2]	1.000	—	—	—	—
	11	[88, 1]	1.000	—	—	—	—
	12	[88, 2]	1.000	—	—	—	—

C Specification and Subspace Closure for JH

C.1 The JH hash function family

JH [27] defines four members, JH-224, JH-256, JH-384, and JH-512, which share the compression function specified below and differ only in the initial chaining value and in the digest truncation length n .

Padding. Let $|M|$ denote the bit length of a message M . The padding appends a bit 1, followed by $383 + (-|M| \bmod 512)$ zero bits, and finally the 128-bit big-endian representation $\langle |M| \rangle_{128}$ of $|M|$. Thus at least 512 bits are appended, and the padded message splits into 512-bit blocks. For a single-block message ($|M| = 512$), the padding is the single block

$$P = 1 \parallel 0^{383} \parallel \langle 512 \rangle_{128}.$$

Compression function and digest. Let $H^{(i-1)} = H_L^{(i-1)} \parallel H_R^{(i-1)}$ be the 1024-bit chaining value and let $M^{(i)}$ be a 512-bit message block. Each half $H_L^{(i-1)}$ and $H_R^{(i-1)}$ has 512 bits, and $H^{(0)}$ is the specified IV for the member. Let E_8 be the 42-round permutation acting on the state as 256 4-bit words. The compression function of JH is defined as

$$H^{(i)} = E_8((H_L^{(i-1)} \oplus M^{(i)}) \parallel H_R^{(i-1)}) \oplus (0^{512} \parallel M^{(i)}).$$

Each round of E_8 consists of a parallel 4-bit S-box layer, the linear transformation L , and the word permutation P_8 (the two S-boxes S_0 and S_1 are selected according to the round constant). The message block $M^{(i)}$ is XORed into the *left* half of the chaining value before E_8 and fed forward into the *right* half afterward. After all padded blocks are processed, the n -bit digest is the last n bits of the final 1024-bit chaining value. The reduced-round attacks retain the specified IVs. They replace each call to E_8 by its first R rounds, including the linear layer of the last round.

C.2 Improved preimage attack using an input subspace

The input-coset filter of Section 5.4 uses a linear space of message differences with one common probability-1 output condition. The closure property below is a restatement, in terms of DL correlations, of Lemmas 1 and 2 of [17]; we give a self-contained proof and show how the common output condition extends from the selected input differences to their linear span.

Lemma 1. *Let $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ be a function. For input differences $\delta_0, \dots, \delta_{t-1} \in \mathbb{F}_2^m$ and output masks $\lambda_0, \dots, \lambda_{k-1} \in \mathbb{F}_2^n$, define $\overline{\mathbb{D}} = \text{Span}\{\delta_0, \dots, \delta_{t-1}\}$ and $W = \text{Span}\{\lambda_0, \dots, \lambda_{k-1}\}$, with orthogonal complement $W^\perp = \{u \in \mathbb{F}_2^n : \lambda^\top u = 0 \text{ for all } \lambda \in W\}$. If the DL correlation of F satisfies $\mathcal{E}_{\delta_i, \lambda_j} = +1$ for all $0 \leq i < t$ and $0 \leq j < k$, then*

$$F(x \oplus \delta) \oplus F(x) \in W^\perp \quad \text{for all } x \in \mathbb{F}_2^m \text{ and } \delta \in \overline{\mathbb{D}}.$$

Equivalently, $\Pr[\delta \xrightarrow{F} W^\perp] = 1$ for every nonzero $\delta \in \overline{\mathbb{D}}$.

Proof. By definition, the DL correlation is

$$\mathcal{E}_{\delta_i, \lambda_j} = 2^{-m} \sum_{z \in \mathbb{F}_2^m} (-1)^{\lambda_j^\top (F(z \oplus \delta_i) \oplus F(z))}.$$

Hence $\mathcal{E}_{\delta_i, \lambda_j} = +1$ is equivalent to

$$\lambda_j^\top (F(z \oplus \delta_i) \oplus F(z)) = 0,$$

for every $z \in \mathbb{F}_2^m$, $0 \leq i < t$, and $0 \leq j < k$, which gives

$$F(z \oplus \delta_i) \oplus F(z) \in W^\perp.$$

Fix an arbitrary input $x \in \mathbb{F}_2^m$ and a difference $\delta \in \mathbb{D}$, and write $\delta = \bigoplus_{i=0}^{t-1} a_i \delta_i$ with $a_i \in \mathbb{F}_2$. Define $x_0 = x$ and $x_{i+1} = x_i \oplus a_i \delta_i$ for $0 \leq i < t$, so that $x_t = x \oplus \delta$. For each i , the output difference

$$F(x_{i+1}) \oplus F(x_i) = \begin{cases} F(x_i \oplus \delta_i) \oplus F(x_i), & \text{if } a_i = 1, \\ 0, & \text{if } a_i = 0, \end{cases}$$

which always belongs to $W^\perp$. Therefore, the output difference for the input difference δ is

$$F(x \oplus \delta) \oplus F(x) = F(x_t) \oplus F(x_0) = \bigoplus_{i=0}^{t-1} (F(x_{i+1}) \oplus F(x_i)) \in W^\perp,$$

since all intermediate terms belong to the linear subspace $W^\perp$.

For the reduced-round JH instances of Section 5, we verify the hypothesis $\mathcal{E}_{\delta_i, \lambda_j} = +1$ by conservatively propagating the support of potentially active 4-bit words through both compression calls, including the message feed-forward, and stopping at the last S-box layer. Pulling a digest mask back through the final word permutation and linear layer expresses it on the words at that point; the selected masks are supported only on words that the propagation proves inactive, so the corresponding correlations equal +1 for every message.

D Specification and Second-Preimage Attack Results on SKINNY-Hash

D.1 The SKINNY-Hash family

SKINNY-Hash [2] is a function-based sponge hash family with two members, SKINNY-tk2-Hash and SKINNY-tk3-Hash. Table 8 lists their digest size n , state size b , absorbing rate r and capacity $c = b - r$, and squeezing rate r' and capacity $c' = b - r'$. The members also use different tweakable block ciphers in their state-transition functions.

Table 8. Parameters of the two SKINNY-Hash members. SKINNY-tk2-Hash uses an absorbing rate $r = 32$ that differs from its squeezing rate $r' = 128$; SKINNY-tk3-Hash uses $r' = r = 128$. All values are in bits.

Member	n	b	r	c	r'	c'	internal function
SKINNY-tk2-Hash	256	256	32	224	128	128	F_{256}
SKINNY-tk3-Hash	256	384	128	256	128	256	F_{384}

Internal function F_b . The state transition $F_b : \mathbb{F}_2^b \rightarrow \mathbb{F}_2^b$ of SKINNY-Hash is a non-invertible function built from $d = b/128$ parallel calls to a tweakable block cipher: SKINNY-128-256 for F_{256} ($d = 2$) and SKINNY-128-384 for F_{384} ($d = 3$). The entire state x is used as the tweak for each call. Its consecutive 128-bit components are denoted $TK_1, \dots, TK_d$. Writing $\tilde{E}_x(\cdot)$ for encryption under tweak x , define the fixed plaintext blocks

$$C_0 = 0^{128}, \quad C_1 = 0^7 \| 1 \| 0^{120}, \quad C_2 = 0^6 \| 1 \| 0^{121}.$$

The specification then fixes

$$\begin{aligned} F_{384}(x) &= \tilde{E}_x(C_0) \| \tilde{E}_x(C_1) \| \tilde{E}_x(C_2), \\ F_{256}(x) &= \tilde{E}_x(C_0) \| \tilde{E}_x(C_1). \end{aligned}$$

Thus the first byte of C_j has value j , and the remaining bytes are zero. The construction is illustrated in Figure 5.

Sponge construction. SKINNY-Hash starts from $S_0 = 0^r \| 1 \| 0^{c-1}$. The padding appends a bit 1 followed by the minimum number of zeros needed to reach a multiple of r bits. The padded message is parsed into ℓ blocks of r bits each and absorbed into the rate part under F_b . Squeezing extracts the leading 128 bits of the state as a digest block, applies F_b once, and extracts another 128 bits; the digest is $H = H_0 \| H_1$ in both members. For SKINNY-tk2-Hash, the rate boundary switches between phases (absorbing rate $r = 32$, squeezing rate $r' = 128$); for SKINNY-tk3-Hash the boundary is the same ($r' = r = 128$).

D.2 Prefix Sampling and Padding

For a fixed prefix preceding the TD probe, the last-block map has 2^r inputs and 2^c possible capacity outputs. Since $r < c$, many such maps have no preimage of the target set. We therefore refresh the preceding prefix rather than sample one last-block map indefinitely.

Choose $\ell' \geq \max\{\ell, \lceil c/r \rceil + 2\}$ and a batch size B . Each batch independently samples $\ell' - 1$ uniform message blocks and absorbs them from the IV, at a cost of $\ell' - 1$ calls to F_b . Call this parent prefix V . At its final state, run B TD-filtering trials with independently uniform last blocks, stopping at the first valid second preimage. If all trials fail, sample a new parent prefix. The full ℓ' -block prefix domain has at least 2^{c+2r} strings, while the parent-prefix domain has at

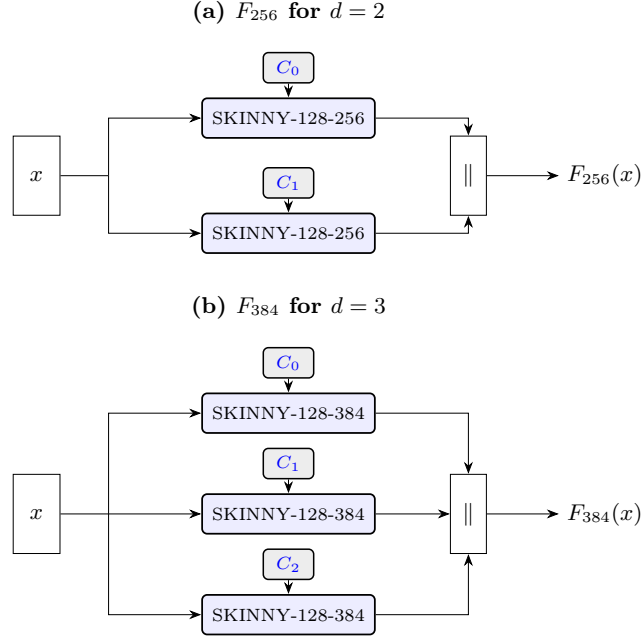


Fig. 5. Internal function F_b of SKINNY-Hash. The state x is used as the tweakkey for d parallel calls of the underlying tweakable block cipher on the fixed plaintexts $C_0, \dots, C_{d-1}$, and the d ciphertexts are concatenated to form $F_b(x)$. (a) F_{256} used in SKINNY-tk2-Hash ($d = 2$, SKINNY-128-256); (b) F_{384} used in SKINNY-tk3-Hash ($d = 3$, SKINNY-128-384).

least 2^{c+r} strings. The estimates below model the full-prefix capacity outputs as uniform; the domain size alone does not assert uniformity for the concrete function.

Fix F_b , the target message, and the filters. Let q_V and μ_V be the success probability and expected cost of the probe and confirmations of one trial for parent V , including the padding check of Section 6.1. Define

$$\bar{q} = \mathbb{E}_V[q_V], \quad q_2 = \mathbb{E}_V[q_V^2], \quad \bar{\mu} = \mathbb{E}_V[\mu_V].$$

Conditional on V , the trials are independent. Thus the exact batch success probability is

$$Q_B = \mathbb{E}_V[1 - (1 - q_V)^B], \quad B\bar{q} - \binom{B}{2}q_2 \leq Q_B \leq B\bar{q}.$$

The independently sampled batches are also independent for the fixed attack instance, even if their candidate prefixes overlap. Their expected cost is at most $\ell' - 1 + B\bar{\mu}$. Consequently, whenever the denominator is positive, the expected

linking work satisfies

$$T \leq \frac{(\ell' - 1)/B + \bar{\mu}}{\bar{q} - (B - 1)q_2/2}. \quad (22)$$

This approaches $\bar{\mu}/\bar{q}$ when prefix generation and the within-batch correction are small.

To quantify these corrections, let L_V count the last blocks whose capacity outputs belong to the K distinct targets, and put $a = (s + 1)2^{-r}$ and $M_2 = \mathbb{E}_V[L_V(L_V - 1)]$. Each target preimage can be reached from at most $s + 1$ base blocks, so $q_V \leq aL_V$ and $q_V^2 \leq aq_V + a^2L_V(L_V - 1)$. Hence

$$q_2 \leq a\bar{q} + a^2M_2.$$

The leading estimate uses the sparse-occupancy condition $aM_2 \ll \bar{q}$, together with a batch size satisfying $\ell' \ll B$ and $B(s + 1) \ll 2^r$. These conditions make both corrections small. For comparison, independent uniform capacity outputs give $\mathbb{E}[L_V] = \rho = K2^{r-c}$ and $M_2 \leq \rho^2$; this is a benchmark for the occupancy condition, not an independence assumption on all outputs of the biased function F_b . Equation (22) keeps the finite-batch cost explicit; the reported gain values use the leading filtering term. Under the uniform-capacity model, a fixed target contributes the same 2^{-c} hit probability as any other fixed capacity value. The multi-target acceptance products remain the estimates of Section 3.3, not an independence claim about the actual target trajectory.

The padding check rejects only a hit at $j = \ell - 1$ for which $\widehat{M}_j = 0^r$. Under the additional uniform-full-state model, one candidate has both the required final capacity and this one forbidden rate value with probability $2^{-(c+r)}$. A trial contains at most $s + 1$ candidates, so discarding these hits decreases its average success probability by at most $(s + 1)2^{-(c+r)}$ in that model. The leading estimates omit this term. For all earlier target positions, the original final padded block is retained and no padding rejection is needed.

D.3 Second-Preimage Attack Results on SKINNY-Hash

This appendix gives the second-preimage attack instances on **SKINNY-Hash** that are not worked out in Section 6. Each instance gives the TD search parameters, the gain for one target ($K = 1$), and its dependence on K under the target model of Section 3.3. We write $\sum p$ and $\sum q$ for the sums of true-positive and uniform-null acceptance probabilities over the selected input differences. Gains are multiplicative; their base-2 logarithms measure the speed-up in bits. The curves use the homogeneous-rate approximation $p_i = (\sum p)/s$ and $q_i = (\sum q)/s$. This reproduces the gain at $K = 1$ from the displayed sums; for nondeterministic filters, it approximates the nonlinear multi-target formula. The single-target results for **SKINNY-tk3-Hash** and **SKINNY-tk2-Hash** are summarized in Tables 4 and 9, respectively.

5-round attack on SKINNY-tk3-Hash. The TD uses input differences in the 128-bit absorbing rate of F_{384} and output masks on its second branch $\tilde{E}_x(C_1)$,

which lies entirely in the capacity. The 5-round attack uses $s = 4080$ input differences; for each, we select $k = 14$ linearly independent output masks with absolute correlation 1, which together fix one affine truncated differential of probability 1. For $K = 1$ the Gain is

$$\text{Gain} = \frac{1 + 4080}{1 + 4080 \cdot 2^{-14}} \approx 2^{11.674}.$$

As K grows, the Gain decreases, as shown in Figure 6. The attack still gives more than two bits of speedup for $K \leq 4709$.

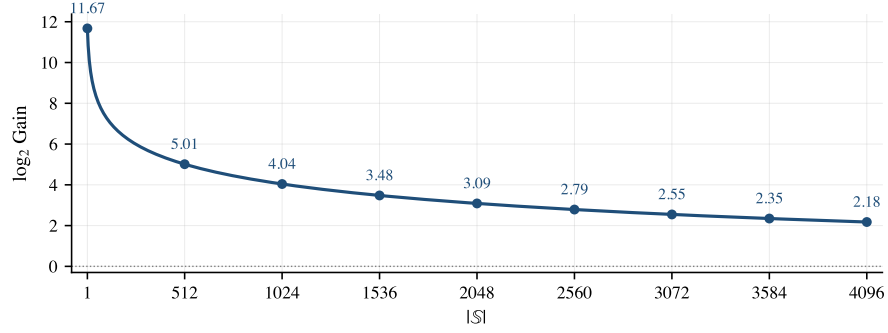


Fig. 6. Model gain of the 5-round second-preimage attack on **SKINNY-tk3-Hash** as the number of targets $K = |S|$ grows, under the homogeneous-rate approximation.

7-round attack on SKINNY-tk3-Hash. We use $s = 205$ input differences; for each, we select at most $k = 10$ linearly independent output masks and retain the high-probability affine cosets. For $K = 1$ the Gain is

$$\text{Gain} = \frac{1 + 9.806}{1 + 2.141} \approx 2^{1.783}.$$

The model curve is shown in Figure 7. The attack gives more than one bit of speedup only for $K \leq 3$, and the Gain approaches the no-gain baseline as K grows further.

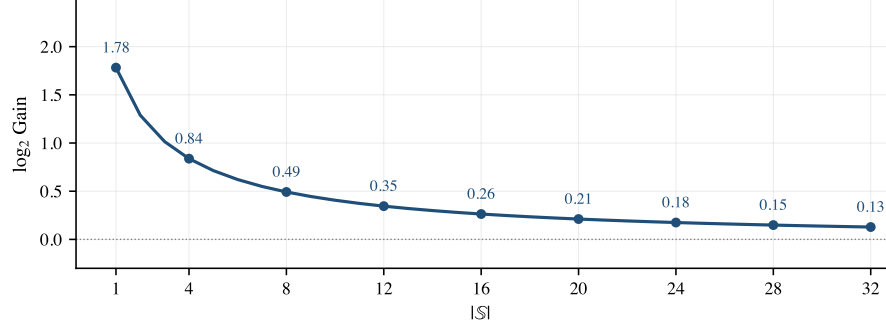


Fig. 7. Model gain of the 7-round second-preimage attack on **SKINNY-tk3-Hash** as the number of targets $K = |S|$ grows, under the homogeneous-rate approximation.

Table 9. Second-preimage attack results for **SKINNY-tk2-Hash** at $K = 1$. Here R is the number of rounds of the underlying tweakable block cipher, s is the number of input differences, and k is an upper bound on the number of independent output masks per filter. Time is $2^c / (K \text{ Gain})$ with $c = 224$, counted in evaluations of F_{256} .

R	s	k	$\sum p$	$\sum q$	Gain	Time
5	1020	14	1020	0.062	$2^{9.909}$	$2^{214.091}$
6	130	11	7.353	1.959	$2^{1.497}$	$2^{222.503}$

5-round attack on SKINNY-tk2-Hash. For **SKINNY-tk2-Hash**, the truncated differential is constructed in the related-tweakey setting, with the input difference injected into the first four 8-bit cells of TK_1 , corresponding to the 32-bit absorbing rate. Output masks are restricted to capacity bits. The 5-round attack uses $s = 1020$ input differences; for each, we select $k = 14$ linearly independent output masks with absolute correlation 1, fixing one affine truncated differential of probability 1. For $K = 1$ the Gain is

$$\text{Gain} = \frac{1 + 1020}{1 + 1020 \cdot 2^{-14}} \approx 2^{9.909}.$$

As K grows, the Gain decreases, as shown in Figure 8. The attack retains more than two bits of speedup for $K \leq 4697$.

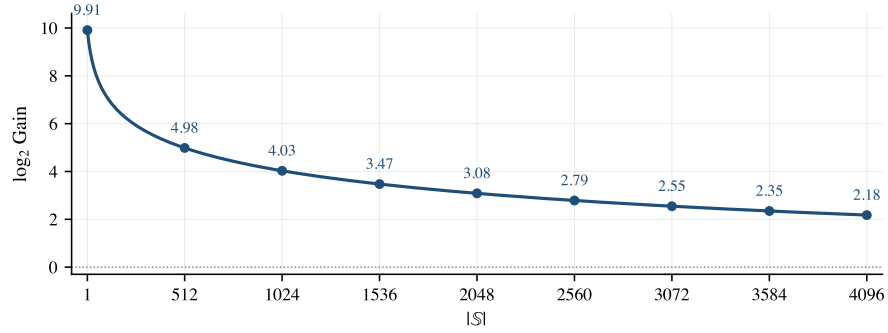


Fig. 8. Model gain of the 5-round second-preimage attack on *SKINNY-tk2-Hash* as the number of targets $K = |S|$ grows, under the homogeneous-rate approximation.

6-round attack on *SKINNY-tk2-Hash*. We use $s = 130$ differences in the related-tweakey setting. For each, we select $k = 11$ linearly independent output masks and retain the high-probability affine cosets. For $K = 1$ the Gain is

$$\text{Gain} = \frac{1 + 7.353}{1 + 1.959} \approx 2^{1.497}.$$

The model curve is shown in Figure 9. The speed-up exceeds one bit only for $K \leq 2$, and decays rapidly for larger K .

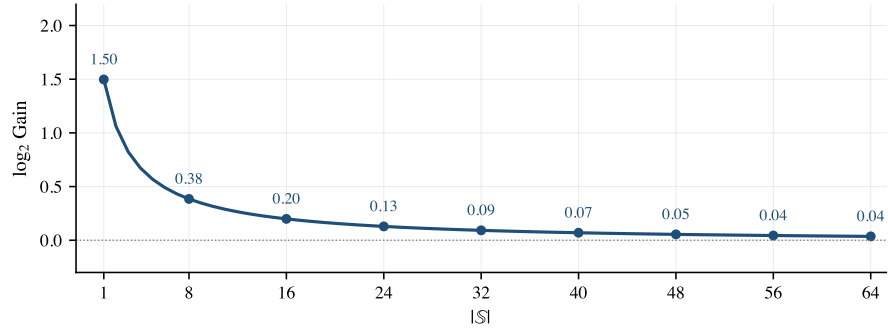


Fig. 9. Model gain of the 6-round second-preimage attack on *SKINNY-tk2-Hash* as the number of targets $K = |S|$ grows, under the homogeneous-rate approximation.