Improved Differential-Linear Cryptanalysis of Orthros, Gleeok, ZIP-AES, and ZIP-GIFT

Yuanxin Wang, Chengcheng Chang, and Kai Hu

School of Cyber Science and Technology, Shandong University, Qingdao, China
wangyxin@mail.sdu.edu.cn, chengcheng.chang@mail.sdu.edu.cn,
kai.hu@sdu.edu.cn

Abstract. We improve differential-linear distinguishers and derive key-recovery attacks on reduced-round versions of the PRFs Orthros, Gleeok, ZIP-AES, ZIP-GIFT-64, and ZIP-GIFT-128. Relative to previous differential-linear analyses, we extend the reported coverage for Orthros from 7 to 9 rounds and for Gleeok-128 from 4 to 5 rounds. For Orthros and Gleeok-128, we obtain estimated correlation magnitudes $2^{-61.40}$ and $2^{-31.71}$, respectively. We also obtain distinguishers for 6-round Gleeok-256, $(3,3)$-round ZIP-AES, $(7,7)$-round ZIP-GIFT-64, and $(10,10)$-round ZIP-GIFT-128. For key recovery, we give a $(3,3)$-round differential-linear attack on ZIP-AES requiring approximately $2^{122.42}$ chosen plaintexts and $2^{122.88}$ ZIP-AES evaluations. Our $(10,10)$-round ZIP-GIFT-128 attack has estimated data, time, and memory complexities of 2^{93} chosen plaintexts, 2^{117} full PRF evaluations, and $2^{25.01}$ bits. We further obtain weak-key attacks on 9-round Orthros, 5-round Gleeok-128, and 6-round Gleeok-256, with key-space coverages of $2^{-2.87}$, 2^{-3}, and 2^{-1}, respectively. The Orthros key-recovery result reaches one round beyond the previous DL key-recovery coverage. These results use automated differential and linear trail search together with existing round-based correlation estimation. We enforce the compatibility constraints between branches and select input-side extensions that make their prescribed differences jointly reachable.

Keywords: Differential-linear cryptanalysis · Pseudorandom functions · Sum of permutations · Key recovery.

1 Introduction

Pseudorandom functions (PRFs) are fundamental building blocks in symmetric cryptography. A classical way to construct PRFs from pseudorandom permutations (PRPs) is to combine permutation evaluations so that the result is no longer invertible. Earlier PRP-to-PRF constructions, including the sum of permutations and EDM/EDMD-type designs, established beyond-birthday-bound security in suitable settings [6,22,10,23]. In the sum-of-permutations construction, the branches process a common input and their outputs are XORed. Parallel branch evaluation makes this construction attractive for low-latency PRFs.

Orthros [2] and Gleeok [1] instantiate this design with two and three branches, respectively. The final XOR hides the individual branch outputs, so attacks must exploit properties that remain observable in the full PRF. At ASIACRYPT 2024, the ZIP framework derived low-latency PRFs from existing block ciphers [14]. If $E = E_b \circ E_f$, the corresponding construction is ZIP-E$(P) = E_f(P) \oplus E_b^{-1}(P)$. Concrete instances include ZIP-AES and ZIP-GIFT. Nageler et al. [24] studied their resistance to differential, linear, and integral cryptanalysis and gave key-recovery attacks on reduced-round variants.

Table 1. Comparison of differential-linear distinguishers from this work and previous studies. The reported value is the absolute correlation.

Target	Rounds	Value	Ref.
Orthros	6	2^{-46}	[20]
	7	$2^{-42.44}$	[16]
	7	$2^{-42.25}$	Sec. 3.1
	9	$2^{-61.40}$	Sec. 3.1
Gleeok-128	4	$2^{-24.55}$	[9]
	4	$2^{-11.64}$	Sec. 3.3
	5	$2^{-31.71}$	Sec. 3.3
Gleeok-256	**6**	$2^{-68.13}$	Sec. B
ZIP-AES	**(3,3)**	$2^{-61.97}$	Sec. 4.1
ZIP-GIFT-64	**(7,7)**	$2^{-27.72}$	Sec. C
ZIP-GIFT-128	**(10,10)**	$2^{-52.75}$	Sec. 4.2

This paper establishes a differential-linear comparison for these branch-function PRFs, with improvements measured by round coverage, correlation, and compatible key-recovery extensions.

Differential-linear cryptanalysis combines a differential propagation over the first part of a primitive with a linear approximation over the remaining part [19]. Compared with pure differential or pure linear cryptanalysis, it is often more flexible: the attacker does not need to predict a full output difference, but only a biased parity of the output difference under a linear mask. This is particularly useful for branch constructions. Blondeau et al. [8] formalized the analysis of differential-linear correlations under an independence assumption. The Differential-Linear Connectivity Table (DLCT) [4] accounts for dependencies between the differential and linear parts. Subsequent methods include the Differential Algebraic Transitional Form (DATF) [21], multi-round DLCT analysis [15], and correlation estimation using truncated differentials [26].

Beyne [7] introduced a geometric approach to linear cryptanalysis. Within this framework, Hu et al. [18] developed a mix-basis treatment of combined attacks, including differential-linear cryptanalysis. More recently, the round-based

method of Hu et al. [17] gives more accurate estimates of differential-linear correlations by taking hull effects into account rather than relying only on a single trail.

These techniques allow us to revisit the reduced-round security of the full parallel-branch PRFs. Our objective is to extend the attacked rounds and obtain concrete key-recovery results. We use existing automated differential and linear trail search together with the round-based correlation estimates of Hu et al. [17]. The selected branch properties must correspond to a common external input difference and a common output mask. For key recovery, the same plaintext pair must also reach the prescribed internal differences through the guessed keyed prefixes. We account for these conditions when selecting the distinguishers used in the attacks.

Table 2. Comparison of our key-recovery attacks with previous key-recovery attacks. Here I, L, D, T-D, DL denote integral, linear, differential, truncated-differential and differential-linear attacks respectively. Memory is given in bits. Coverage denotes the fraction of the full key space covered by an attack. The table highlights round coverage, concrete complexities, and key-space coverage.

Target	Attack	Rounds	Data	Time	Memory	Coverage	Ref.
	D	6	2^{115}	2^{120}	2^{29}	$2^{-4.55}$	[20]
	T-D	7	$2^{75.06}$	$2^{117.03}$	$2^{83.07}$	2^{-5}	[28]
Orthros	DL	6	2^{40}	$2^{115.07}$	2^{47}	1	[16]
	DL	7	2^{95}	2^{106}	2^{29}	$2^{-4.55}$	[20]
	DL	8	2^{112}	$2^{119.58}$	2^{119}	1	[16]
	DL	8	$2^{101.64}$	2^{117}	$2^{21.04}$	$2^{-2.87}$	Sec. 3.2
	DL	**9**	$2^{117.96}$	$2^{122.98}$	$2^{21.04}$	$2^{-2.87}$	Sec. 3.2
Gleeok-128	I	7	2^{124}	$2^{133.6}$	–	1	[9]
	I	7	2^{124}	2^{132}	2^{131}	1	[9]
	I	8	2^{128}	2^{137}	–	1	[9]
	I	8	2^{128}	2^{129}	2^{135}	1	[9]
	DL	5	$2^{58.77}$	2^{250}	2^{17}	2^{-3}	Sec. 3.4
Gleeok-256	DL	6	$2^{145.12}$	2^{249}	2^{22}	2^{-1}	Sec. B
ZIP-AES	I	$(3,3)$	$\approx 2^{41.3}$	2^{48}	$\approx 2^{7}$	1	[24]
	I	$(4,2)$	2^{40}	2^{96}	2^{39}	1	[24]
	L	$(3,2)$	$2^{123.7}$	$2^{123.7}$	2^{119}	1	[24]
	D	$(3,2)$	2^{42}	2^{97}	2^{31}	1	[24]
	DL	$(3,3)$	$2^{122.42}$	$2^{122.88}$	2^{16}	1	Sec. 4.1
ZIP-GIFT-128	DL	$(10,10)$	2^{93}	2^{117}	$2^{25.01}$	1	Sec. 4.2

Our results. We improve the differential-linear cryptanalysis of Orthros, Gleeok, ZIP-AES, and ZIP-GIFT and obtain the following results.

The main improvements are measured within the differential-linear family: we extend the reported DL round coverage for Orthros and Gleeok-128, and obtain compatible full-PRF DL distinguishers for ZIP constructions with larger total branch-round coverage than the corresponding DL results for the underlying ciphers.

- **New and extended differential-linear distinguishers for Orthros and Gleeok.** We obtain a 9-round Orthros differential-linear distinguisher with estimated correlation magnitude $2^{-61.40}$, extending the 7-round result of Hadipour et al. [16] by two rounds. For Gleeok-128, we obtain a 5-round distinguisher with estimated correlation magnitude $2^{-31.71}$, extending the 4-round result of Chen et al. [9] by one round. We also obtain a 6-round Gleeok-256 distinguisher with estimated correlation magnitude $2^{-68.13}$. Table 1 compares our results with the previous differential-linear distinguishers of Li et al. [20], Hadipour et al. [16], and Chen et al. [9].
- **Differential-linear distinguishers for ZIP-AES and ZIP-GIFT.** We obtain full-PRF distinguishers for $(3, 3)$-round ZIP-AES, $(7, 7)$-round ZIP-GIFT-64, and $(10, 10)$-round ZIP-GIFT-128. Their estimated correlation magnitudes are $2^{-61.97}$, $2^{-27.72}$, and $2^{-52.75}$, respectively. Counting a ZIP round pair (r_f, r_b) by the total length $r_f + r_b$ of its two parallel branches, these distinguishers cover 6, 14, and 20 branch rounds. For comparison, the differential-linear distinguishers of Peng et al. [26] cover 5 rounds of AES, and those of Wang et al. [29] cover 12 rounds of GIFT-64 and 17 rounds of GIFT-128. These results show that differential-linear correlations survive the final XOR when the branch input differences and output masks are compatible.
- **Key recovery on ZIP-AES and ZIP-GIFT-128.** We obtain a differential-linear key-recovery attack on $(3, 3)$-round ZIP-AES. The attack requires approximately $2^{122.42}$ chosen plaintexts and $2^{122.88}$ full $(3, 3)$-round ZIP-AES evaluations. Table 2 summarizes the round coverage and concrete complexities of the resulting key-recovery attacks. For $(10, 10)$-round ZIP-GIFT-128, our attack has estimated data and time complexities of 2^{93} chosen plaintexts and 2^{117} full PRF evaluations, with $2^{25.01}$ bits of memory. These attacks use the input-side feasibility criterion of Flórez-Gutiérrez et al. [14], restated as Proposition 1. The distinguishers used for key recovery are selected to satisfy the required input-side compatibility conditions.
- **Weak-key recovery on Orthros and Gleeok.** We obtain key-recovery attacks on 9-round Orthros, 5-round Gleeok-128, and 6-round Gleeok-256. The attacks apply to key classes occupying fractions $2^{-2.87}$, 2^{-3}, and 2^{-1} of their respective key spaces. In particular, the 9-round Orthros attack requires approximately $2^{117.96}$ chosen plaintexts, $2^{122.98}$ 9-round evaluations, and $2^{21.04}$ bits of memory. It reaches one round beyond the 8-round attack of Hadipour et al. [16]. Table 2 highlights our key-recovery attacks, their round coverage, complexities, and key-space coverage. The additional ZIP-GIFT

attacks in Appendices C and D combine a differential-linear forward branch with a linear backward branch.

All results are summarized in Tables 1 and 2. All source code needed to reproduce the results in this paper is available at `https://anonymous.4open.science/r/SoP-Attack-A702/`.

Outline. The rest of this paper is organized as follows. Section 2 recalls the necessary background on differential-linear cryptanalysis and the branch-function constructions considered in this paper. Section 3 reports the corresponding results for Orthros and Gleeok. Section 4 applies the method to ZIP-AES and ZIP-GIFT. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 Differential-Linear Cryptanalysis

Langford and Hellman [19] introduced differential-linear (DL) cryptanalysis by combining a differential over F_0 with a linear approximation over F_1, where $F = F_1 \circ F_0$. Let p denote the differential probability, q the linear correlation, and C_F the resulting DL correlation. Under the standard independence assumption, C_F is approximated by $C_F \approx pq^2$. When this assumption does not hold, the estimate may deviate substantially from the actual DL correlation.

The Differential-Linear Connectivity Table (DLCT), introduced by Bar-On et al. [4], captures such dependencies. They used the following sandwich decomposition: $F = F_l \circ F_m \circ F_d$. A differential trail through F_d and a linear trail through F_l determine the input difference and output mask, respectively, of F_m. Here, p and q denote the differential-trail probability through F_d and the linear-trail correlation through F_l, respectively. Letting c_m denote the DL correlation of the middle part, the combined correlation is estimated as $C_F \approx p\, c_m\, q^2$. When F_m consists of a single S-box layer, the DLCT directly provides the corresponding DL correlation.

For multi-round middle components, Liu et al. [21] introduced the Differential Algebraic Transitional Form (DATF) to estimate DL biases. Hadipour et al. [15] extended the DLCT framework to multiple rounds from a boomerang perspective. More recently, Hu et al. [17] proposed a round-based approximation of DL correlations within Beyne's geometric framework [7].

Building on these results, we apply the sandwich decomposition to sum-of-permutations (SoP) constructions and use the round-based approximation of Hu et al. [17] to estimate the correlation of F_m. The differential trails determine the input differences to the middle part, whereas the linear trails determine its output masks.

2.2 Geometric Approach and Round-Based Approximation

We briefly review the round-based approximation of Hu et al. [17], which we use throughout this paper to estimate the middle correlations of our DL dis-

tinguishers. In Beyne's geometric approach [7], a multiset is represented by its correlation vector. For a nonempty finite multiset $\mathbb{S}$ over $\mathbb{F}_2^n$,

$$\mathrm{CV}(\mathbb{S})[u] = |\mathbb{S}|^{-1} \sum_{x \in \mathbb{S}} (-1)^{u^\top x}.$$

In DL cryptanalysis, the correlation associated with an input difference Δ and an output mask λ can be viewed as a coordinate of the correlation vector obtained from the input-pair multiset. Because exact propagation of this vector is infeasible for large n, Hu et al. approximate it round by round while retaining components associated with individual S-box positions.

In this representation, each coordinate is indexed by (v_0, v_1), where v_0 and v_1 are the value mask and difference mask, respectively. We adopt the zero-value-mask approximation and index each retained $(0, v)$-coordinate simply by v. Key and constant XOR additions do not affect these coordinates and are therefore omitted below.

Let the r-round middle part be $F_m = S_r \circ L_{r-1} \circ S_{r-1} \circ \cdots \circ L_1 \circ S_1$, $S_i = \|_{j=0}^{t-1} s_{i,j}$, The input difference and output mask of F_m are denoted by Δ and λ, respectively. Let ϖ_j denote the projection onto the j-th local component, and let τ_j denote the inclusion map that places a local mask in the j-th component and sets all other components to zero.

For each S_i, let $\sigma_{i-1}^{[j]}$ and $\gamma_i^{[j]}$ denote the local correlation vectors before and after the S-box $s_{i,j}$, respectively. The fixed input difference initializes these vectors as $\sigma_0^{[j]}[v] = (-1)^{v^\top \varpi_j(\Delta)}$, $v \in \mathbb{F}_2^{m_j}$.

For an m_j-bit S-box s, its correlation matrix [11] is $C^s[v, u] = 2^{-m_j} \mathrm{LAT}_s[u, v]$, where the LAT entries are unnormalized Walsh sums. Under the zero-value-mask approximation, the S-box update is

$$\gamma_i^{[j]}[v] = \sum_{u \in \mathbb{F}_2^{m_j}} \left(C^{s_{i,j}}[v, u]\right)^2 \sigma_{i-1}^{[j]}[u].$$

The linear layer is handled through transpose mask propagation. Because L_i may mix different local components, the joint coordinate is approximated by the product of its local coordinates:

$$\sigma_i^{[j]}[v] = \prod_{k=0}^{t-1} \gamma_i^{[k]} \left[\varpi_k\left(L_i^\top \tau_j(v)\right)\right], \quad 1 \le i < r.$$

After the last S-box layer, the middle correlation is estimated as

$$c_m = \prod_{j=0}^{t-1} \gamma_r^{[j]}[\varpi_j(\lambda)].$$

If F_m includes a final linear layer L_r, the same formula is applied with $L_r^\top \lambda$ in place of λ.

When Δ is sparse, the output differences of the first S-box layer can instead be enumerated explicitly, as in [17]. Let $p_1(\Delta')$ denote the probability of $\Delta \xrightarrow{S_1} \Delta'$ for a uniformly random input. For each feasible Δ', initialize the local vector by setting $\gamma_1^{[j]}[v] = (-1)^{v^\top \varpi_j(\Delta')}$, and then apply the remaining updates, starting with L_1 when $r > 1$. Denoting the resulting output estimate by $c(\Delta', \lambda)$, the weighted middle-correlation estimate is

$$c_m = \sum_{\Delta'} p_1(\Delta')\, c(\Delta', \lambda).$$

2.3 Structural DL Attacks on Sum-of-Permutations Constructions

The sum-of-permutations (SoP) construction was introduced by Bellare et al. at EUROCRYPT 1998 [6]. It evaluates several keyed permutations on the same input and XORs their outputs, as shown in Fig. 1. For an input $x \in \mathbb{F}_2^n$, the construction is defined by

$$F_K(x) = \bigoplus_{b=1}^{N} E_K^b(x), \qquad N \geq 2.$$

Here, $E_K^b : \mathbb{F}_2^n \to \mathbb{F}_2^n$ denotes the keyed permutation in the b-th branch, and K is the master key from which the branch round keys are derived. Each branch E_K^b is invertible, whereas their XOR F_K is not necessarily a permutation.

The appeal of SoP constructions lies in their parallel structure. Because the branches can be evaluated simultaneously, the critical path of a fully unrolled implementation is determined by the longest branch followed by the final XOR, rather than by the sum of all branch delays [2,1]. This property makes the paradigm suitable for latency-sensitive applications, such as memory encryption and pointer authentication. From a provable-security perspective, sums of independent random permutations can also achieve security beyond the birthday bound [22,5,25,13,12].

When the sandwich decomposition is applied to an SoP construction, the compatibility constraints depend on whether a single branch or the complete PRF is analyzed. A single branch can be analyzed without cross-branch constraints. For the complete PRF, however, all branches must share the same external input difference and a common output mask. Our search uses the same round partition in every branch, while allowing the internal differences and masks to vary across branches. Under the standard assumption that the branch contributions are statistically independent [14], the correlation of the complete PRF is estimated as the product of the branch correlations.

The final XOR prevents direct access to the individual branch outputs. We therefore consider input-side key-recovery extensions. Let P_K^b denote the added prefix in the b-th branch, and let Δ_d^b denote the required difference at the input of the corresponding branch distinguisher. For a given guess of the prefix subkeys, a queried pair $(x, x \oplus \Delta)$ is useful only if

$$P_K^b(x) \oplus P_K^b(x \oplus \Delta) = \Delta_d^b, \qquad 1 \leq b \leq N.$$

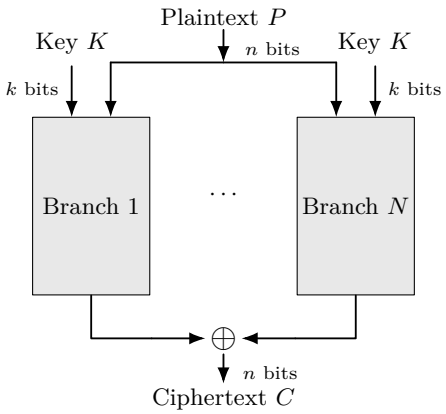

Fig. 1. An N-branch sum-of-permutations construction.

Thus, the required internal difference in every branch must be reachable from the same external difference Δ under the guessed prefix subkeys. The internal differences Δ_d^b may vary across branches, whereas the output mask must be common to all branches because it is applied after the final XOR.

Success probability. Following the normal approximation of Blondeau et al. [8], the success probability of a DL key-recovery attack with correlation C_{DL} is estimated as

$$P_S = \Phi\Big(\sqrt{N}\,|C_{\mathrm{DL}}| - \Phi^{-1}(1-2^{-a})\Big)\,, \tag{1}$$

where Φ is the cumulative distribution function of the standard normal distribution, N is the number of chosen plaintext pairs, and a is the advantage of the attack as defined by Selçuk [27]. Here $|C_{\mathrm{DL}}| = 2\,|p_{\mathrm{dl}} - \frac{1}{2}|$, where p_{dl} is the probability of the DL relation.

3 Differential-Linear Cryptanalysis of Orthros and Gleeok

This section first recalls the specifications of Orthros [2] and Gleeok [1], both of which are dedicated SoP constructions. We then apply the sandwich decomposition introduced in Section 2 to construct differential-linear distinguishers for these two primitives and extend the distinguishers to key-recovery attacks.

Specification of Orthros. Orthros is a 128-bit pseudorandom function with a 128-bit master key. It follows the sum-of-permutations paradigm and consists of two parallel 128-bit keyed permutations, denoted by E_K^1 and E_K^2. Given a plaintext $P \in \mathbb{F}_2^{128}$, the two branches process the same input independently, and their outputs are XORed to produce

$$\mathrm{Orthros}_K(P) = \mathrm{E}_K^1(P) \oplus \mathrm{E}_K^2(P).$$

Each branch is a 12-round SPN-based keyed permutation operating on a 128-bit state represented by 32 nibbles. The two branches share the same 4-bit S-box and matrix multiplication layer but use different permutation layers, round constants, and key schedules. From the common master key K, the bit-permutation-based key scheduling functions KSF_1 and KSF_2 generate the whitening key RK_0^b and the round keys RK_r^b, $1 \le r \le 12$, for Branch $b \in \{1, 2\}$. The corresponding whitening key is XORed with each branch state before the first round.

We index the rounds from 0 to 11. In rounds 0 to 3, each round consists of an S-box layer, the branch-dependent bit permutation P_{br}^1 or P_{br}^2, the matrix multiplication layer M, round-key addition, and round-constant addition. In rounds 4 to 10, the bit permutation is replaced by the branch-dependent nibble permutation P_{n}^1 or P_{n}^2, while the other operations remain unchanged. More precisely, the non-final round transformations are

$$\mathcal{R}_r^b = \begin{cases} \mathrm{AC}_{r+1}^b \circ \mathrm{AK}_{r+1}^b \circ M \circ P_{\mathrm{br}}^b \circ S, & 0 \le r \le 3, \\ \mathrm{AC}_{r+1}^b \circ \mathrm{AK}_{r+1}^b \circ M \circ P_{\mathrm{n}}^b \circ S, & 4 \le r \le 10. \end{cases}$$

Here, $b \in \{1, 2\}$ denotes the branch index. Since RK_0^b is the whitening key, zero-based round r uses RK_{r+1}^b and RC_{r+1}^b. In the final round $r = 11$, the permutation and matrix multiplication layers are omitted, leaving only the S-box layer followed by the additions of RK_{12}^b and RC_{12}^b.

Specification of Gleeok. Gleeok is a family of pseudorandom functions built from three parallel SPN-based keyed permutations. It has two variants, Gleeok-128 and Gleeok-256, with block sizes of 128 and 256 bits, respectively, and both variants use a 256-bit master key. We first describe the Gleeok-128 variant. Given an input $P \in \mathbb{F}_2^{128}$, three keyed permutations, denoted by G_K^1, G_K^2, and G_K^3, process the same input independently, and their outputs are XORed as

$$\mathrm{Gleeok}_K(P) = \mathrm{G}_K^1(P) \oplus \mathrm{G}_K^2(P) \oplus \mathrm{G}_K^3(P).$$

Each branch of Gleeok-128 is a 12-round SPN-based keyed permutation operating on a 128-bit state. The three branches share the same high-level round structure but use branch-dependent nonlinear layers, linear layers, round constants, and key schedules. The 256-bit master key K is loaded differently in the three branches and processed by three permutation-based key scheduling functions to generate the corresponding whitening and round keys. The whitening key is XORed with each branch state before the first round.

In Branches 1 and 2, the nonlinear layer applies the 3-bit S-box S_3 and the 5-bit S-box S_5 alternately, so that each consecutive 8-bit group is processed by $S_3 \parallel S_5$. Branch 3 instead applies 32 parallel copies of the 4-bit S-box S_4. We denote the nonlinear layer of Branch b by S^b, where $S^1 = S^2$ denotes the alternating $S_3 \parallel S_5$ structure and S^3 consists of parallel applications of S_4.

The linear layer of each branch consists of a three-input XOR layer θ^b followed by a bit permutation π^b. For an input $x \in \mathbb{F}_2^{128}$, the output of θ^b is defined by $(\theta^b(x))_i = x_{i+t_0^b} \oplus x_{i+t_1^b} \oplus x_{i+t_2^b}$, where $b \in \{1, 2, 3\}$ and all indices are

taken modulo 128. The subsequent bit permutation is defined by $(\pi^b(x))_i = x_{p^b i \bmod 128}$, where $p^1 = p^2 = 117$, $p^3 = 11$.

We again use zero-based round indices. For the non-final rounds 0 to 10, the round transformation of Branch b is

$$\mathcal{R}^b_r = \mathrm{AC}^b_{r+1} \circ \mathrm{AK}^b_{r+1} \circ \pi^b \circ \theta^b \circ S^b, \qquad 0 \le r \le 10,$$

where AK^b_{r+1} and AC^b_{r+1} denote the additions of RK^b_{r+1} and RC^b_{r+1}, respectively. The linear layer is omitted in the final round $r = 11$, whose transformation is $\mathcal{R}^b_{11} = \mathrm{AC}^b_{12} \circ \mathrm{AK}^b_{12} \circ S^b$.

Gleeok-256 retains the same three-branch, 12-round SoP structure but operates on a 256-bit state. Accordingly, Branches 1 and 2 apply 32 parallel $S_3 \parallel S_5$ groups, whereas Branch 3 applies 64 parallel copies of S_4. The branch-dependent linear layers and key schedules are defined over the 256-bit state. Branches 1 and 3 process $P \oplus RK^b_0$ directly, whereas Branch 2 first applies the initialization bit permutation π_{init} to the plaintext and then XORs the whitening key, so its first-round input is $\pi_{\mathrm{init}}(P) \oplus RK^2_0$.

3.1 DL Distinguishers for Orthros

We first search the two branches of Orthros independently and then impose the SoP compatibility constraints to obtain a distinguisher for the full Orthros PRF.

For a non-final round r of branch $b \in \{1, 2\}$, let P^b_r denote the corresponding permutation. Then the linear layer L^b_r can be denoted by $M \circ P^b_r$. Let v be an output mask of this layer. The corresponding input mask u is obtained by backward propagation as $u = (L^b_r)^\top v = (P^b_r)^\top M^\top v = (P^b_r)^{-1} M v$, where $(P^b_r)^\top = (P^b_r)^{-1}$ and $M^\top = M$. This relation provides the local S-box masks used in the round-based evaluation. Using this propagation rule, we obtain the branch-level distinguishers summarized in Table 3, where r_d, r_m, and r_l specify the round decomposition and $C^b_{\mathrm{DL}} = p\, c_m\, q^2$ is the estimated correlation of Branch b.

Table 3. The best branch-level differential-linear distinguishers for Orthros. Here, r_d, r_m, r_l denote the rounds of differential, middle and linear parts, respectively. p and q denote the probability and correlation of differential and linear trails while $|c_m|$ denotes the correlation of middle part.

Target	Rounds	r_d	r_m	r_l	p	$\lvert c_m \rvert$	$\lvert q \rvert$	$\lvert C^b_{\mathrm{DL}} \rvert$
Branch 1	10	4	4	2	2^{-25}	$2^{-6.67}$	2^{-4}	$2^{-39.67}$
Branch 2	10	4	4	2	2^{-22}	$2^{-9.37}$	2^{-4}	$2^{-39.37}$
	11	4	5	2	2^{-22}	$2^{-33.78}$	2^{-4}	$2^{-63.78}$

We next search for distinguishers of the full Orthros PRF while directly enforcing the SoP compatibility constraints described in Section 2.3. We found a

7-round distinguisher with correlation $2^{-42.25}$, which is close to that of the previously best 7-round distinguisher [16], whose correlation is $2^{-42.44}$. It consists of a two-round differential prefix with probability $p = 2^{-21}$ and a five-round middle part with estimated correlation $c_m = 2^{-21.25}$, yielding an overall correlation of $2^{-42.25}$. Let $\{i, x\}$ denote that the difference or mask of the i-th S-box is x. The input difference is $\{\{6, \texttt{0xf}\}, \{24, \texttt{0x1}\}\}$ and the middle difference is $\{\{28, \texttt{0x8}\}, \{30, \texttt{0x8}\}\}$ and $\{\{20, \texttt{0x2}\}, \{17, \texttt{0x2}\}\}$ respectively. The output mask is $\{5, \texttt{0xa}\}$. The consistency with the previous result provides further support for the validity of our search method. The best result is a 9-round differential-linear distinguisher consisting of three differential rounds, five middle rounds, and one linear round. Its correlation is estimated as $|C_{\mathrm{DL}}| \approx 2^{-35} \cdot 2^{-14.40} \cdot (2^{-6})^2 = 2^{-61.40}$, and the propagation is shown in Appendix A.1.

3.2 DL Key Recovery Attacks on Orthros

Since the linear layer of Orthros provides strong diffusion, prepending two or more rounds would involve too many first-round key bits and would lead to time complexity exceeding exhaustive search. The details are shown in Appendix A.3. Consequently, we prepend only one key-recovery round. Since the distinguisher found in Section 3.1 cannot reach the same input difference after one round of inverse key recovery extension, we therefore perform a new search in which the first round is reserved for key recovery and the common plaintext-boundary difference is imposed explicitly.

For an r_d-round differential prefix, the probability p is evaluated only over the last $(r_d - 1)$ rounds. The first round is reserved for key recovery, where the guessed key bits are used to identify plaintext pairs that produce the prescribed S-box output difference.

Characteristic. The search yields an 8-round DL distinguisher with estimated correlation $|C_{\mathrm{DL}}| \approx 2^{-53.59}$. The required input differences of the two branch distinguishers are

$$\Delta_{\mathrm{d}}^1 = \texttt{0000 0000 0000 0044 0000 0000 0000 0000},$$
$$\Delta_{\mathrm{d}}^2 = \texttt{0000 0000 2200 0000 0000 0000 0000 0000}.$$

Propagating these differences backward through the linear layers of the added round gives the required S-box output differences

$$\Delta_{\mathrm{out}}^1 = \texttt{0000 0000 0000 0000 0000 0000 0004 0400},$$
$$\Delta_{\mathrm{out}}^2 = \texttt{0000 0000 0000 0000 0000 0000 0008 0100}.$$

For suitable whitening-key guesses, the corresponding S-box transitions can be induced by the same plaintext pair in both branches. The common output mask is

$$\lambda_{\mathrm{out}} = \texttt{0000 0000 cc0c 0000 0000 0000 0000 0000}.$$

Key-recovery procedure. Prepending the reserved round gives a 9-round differential-linear key-recovery attack on Orthros, as shown in Fig. 2. The attack proceeds as follows.

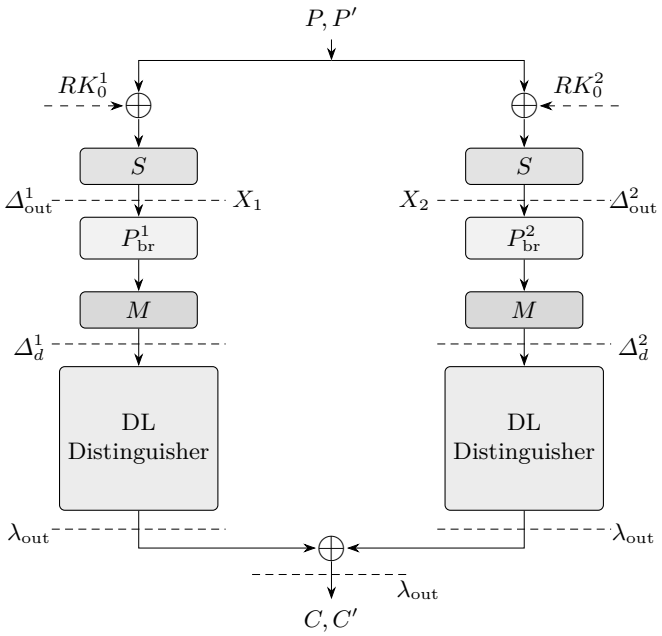

Fig. 2. Differential-linear key-recovery attack on 9-round Orthros.

1. Let A be the set of active plaintext bits involved in the prepended key-recovery round, where $|A| = N_a$. Form a plaintext structure by fixing all bits outside A and enumerating all 2^{N_a} values on A. For each P in the structure, query the oracle and store $C = \text{Orthros}_K(P)$, indexed by P, in a lookup table V_m.
2. Guess the involved whitening-key bits RK_0^1 of Branch 1. For each active S-box position i, enforce the prescribed output difference Δ^1_{out} by computing

$$X_1[i] = S(P[i] \oplus RK_0^1[i]), \qquad (X_1[i])' = X_1[i] \oplus \Delta^1_{\text{out}}[i],$$

   and deriving $P'[i] = S^{-1}((X_1[i])') \oplus RK_0^1[i]$. The resulting pair (P, P') satisfies the required S-box output-difference condition in Branch 1.
3. The whitening keys of both branches are derived from the same 128-bit master key. Hence, once the Branch 1 guess is fixed, N_c master-key bits also determine part of the involved Branch 2 key material. Enumerate only the remaining independent Branch 2 bits. For each candidate, test whether the same plaintext pair also induces the prescribed first-round S-box output difference in Branch 2:

$$S(P[j] \oplus RK_0^2[j]) \oplus S(P'[j] \oplus RK_0^2[j]) = \Delta^2_{\text{out}}[j].$$

   A candidate is discarded as soon as the condition fails for one active S-box position.

4. For every surviving candidate, retrieve $C' = E_K(P')$ from the structure table. Let k denote the corresponding N_t involved master-key bits. The counter $T_k[k]$ records the number of valid pairs associated with k, while $V_k[k]$ records the number of such pairs satisfying $\lambda_{\text{out}} \cdot C \oplus \lambda_{\text{out}} \cdot C' = 0$.
5. For each guess k with $T_k[k] > 0$, if $V_k[k]/T_k[k] - 1/2 > \theta_k$, accept k as a candidate, where θ_k is chosen according to the number $T_k[k]$ of pairs used for this guess.
6. For each retained candidate, verify each full key against two additional plaintext–output pairs and recover the remaining $(128 - N_t)$ master-key bits by exhaustive search.

Complexity analysis. For this attack, $N_a = 8$, $N_c = 3$ and $N_t = 13$ and the expected correlation of the differential-linear distinguisher is $|C_{\text{DL}}| \approx 2^{-53.59}$.

Each plaintext structure contains 2^{N_a} chosen plaintexts and therefore gives approximately 2^{2N_a-1} unordered plaintext pairs. For a fixed guess of the involved Branch 1 whitening-key bits, the prescribed S-box output difference in Branch 1 keeps a fraction 2^{-N_a} of these pairs, or equivalently gives 2^{N_a} ordered Branch 1-compatible pairs.

Furthermore, as pointed out by Flórez-Gutiérrez et al. [14], Orthros exhibits a weak-key phenomenon in differential-style key-recovery attacks, and the same phenomenon occurs here. The relevant weak-key class occupies a fraction $2^{-2.87}$ of the key space, as shown in Appendix A.2. For an 8-bit active input space, a full structure contains four ordered pairs, or two unordered pairs, satisfying the required S-box output differences in both branches in the least favorable valid case. We use this case for the complexity estimate below.

Using Eq. (1), we set $a = 8$ to balance statistical processing and exhaustive search. Taking $N \approx 2^{110.96}$ gives $P_S \approx 85.23\%$. Hence, $m \approx 2^{109.96}$ and $D = m2^8 + 2 \approx 2^{117.96}$ chosen plaintexts.

The exhaustive-search cost is 2^{128-a}. Counting one S-box layer as half a round, the time complexity is $T = 2^{13} \cdot (m2^8) \cdot \frac{8}{4 \cdot 32} \cdot \frac{1}{18} + 2^{128-a} \approx 2^{122.98}$ 9-round Orthros evaluations. Storing two arrays of 128-bit counters and one plaintext–output structure requires $M = 2 \cdot 128 \cdot 2^{13} + 256 \cdot 2^8 \approx 2^{21.04}$ bits of memory.

For the 8-round attack, the same first-round conditions and a 7-round correlation of $2^{-45.2}$ are used. Since at most four local key guesses are indistinguishable, at most 11 of the 13 involved key bits can be recovered in the least favorable case. We therefore set $a = 11$. Taking $N \approx 2^{94.64}$ gives $P_S \approx 85.31\%$. The data, time, and memory complexities are approximately $2^{101.64}$ chosen plaintexts, 2^{117} 8-round Orthros evaluations, and $2^{21.04}$ bits, respectively, with the same weak-key fraction $2^{-2.87}$.

3.3 DL Distinguishers for Gleeok-128

We focus on Gleeok-128 here and present the analysis of Gleeok-256 in Appendix B. As for Orthros, we first analyze the branches independently and then impose the SoP compatibility constraints for the full PRF.

For Branches 1 and 2, each $S_3 \parallel S_5$ pair is treated as an 8-bit local S-box

$$S_{35} : \mathbb{F}_2^8 \to \mathbb{F}_2^8, \qquad S_{35}(x_3 \parallel x_5) = S_3(x_3) \parallel S_5(x_5),$$

where $x_3 \in \mathbb{F}_2^3$ and $x_5 \in \mathbb{F}_2^5$. Hence, the state is divided into 16 local 8-bit components. Branch 3 is evaluated using its 32 native 4-bit S_4 components when analyzed independently. For the full-PRF search, each pair of adjacent S_4 boxes is grouped into an 8-bit local S-box

$$S_{44} : \mathbb{F}_2^8 \to \mathbb{F}_2^8, \qquad S_{44}(x_0 \parallel x_1) = S_4(x_0) \parallel S_4(x_1),$$

where $x_0, x_1 \in \mathbb{F}_2^4$. This grouping aligns the local decomposition across the three branches without changing the round function.

For Branch $b \in \{1, 2, 3\}$, let $L^b = \pi^b \circ \theta^b$ denote its linear layer. The backward propagation of an output mask v is $u = (L^b)^\top v = (\theta^b)^\top (\pi^b)^{-1} v$. Using the shift indices defined above, the transposed XOR layer is $\left((\theta^b)^\top v\right)_i = v_{i-t_0^b} \oplus v_{i-t_1^b} \oplus v_{i-t_2^b}$. All bit indices are taken modulo 128. Thus, backward mask propagation applies $(\pi^b)^{-1}$ followed by $(\theta^b)^\top$.

Using these models, we obtain 7-round DL distinguishers for Branches 1 and 2. Branch 3 admits a 12-round DL distinguisher with estimated correlation $\left|C_{\mathrm{DL}}^3\right| \approx 2^{-55.83}$. This result is consistent with Chen et al. [9], who identified an error in the original mask propagation through θ^3.

In Branches 1 and 2, S_3^i and S_5^i denote the 3-bit and 5-bit S-boxes of byte i, respectively. In Branch 3, S_4^{2i} and S_4^{2i+1} act on the high and low nibbles of byte i. After imposing the SoP compatibility constraints, we find a 4-round distinguisher consisting entirely of the middle part, with correlation $2^{-11.64}$. Its input difference is active only at S_5^0, with difference `0x08`, while the output mask is active only at S_5^1, with mask `0x04`. All unspecified S-box differences and masks are zero. This improves upon the previously best 4-round distinguisher [9], whose correlation is $2^{-24.55}$. The best full-PRF result is a 5-round DL distinguisher with round decomposition $(r_d, r_m, r_l) = (1, 4, 0)$, and estimated correlation $|C_{\mathrm{DL}}| \approx 2^{-31.71}$.

3.4 DL Key-Recovery Attack on Gleeok

We prepend one round to the 4-round DL distinguisher of Gleeok-128 to obtain a 5-round key-recovery attack. Since all three branches process the same plaintext, the added round must map a common external difference to the three prescribed internal differences. For the characteristic considered below, this requirement imposes linear conditions on the master key and restricts the attack to a weak-key class covering 2^{-3} of the key space. The resulting attack is illustrated in Fig. 3.

Characteristic. The four-round DL distinguisher has estimated correlation $|C_{\mathrm{DL}}| = 2^{-25.71}$.

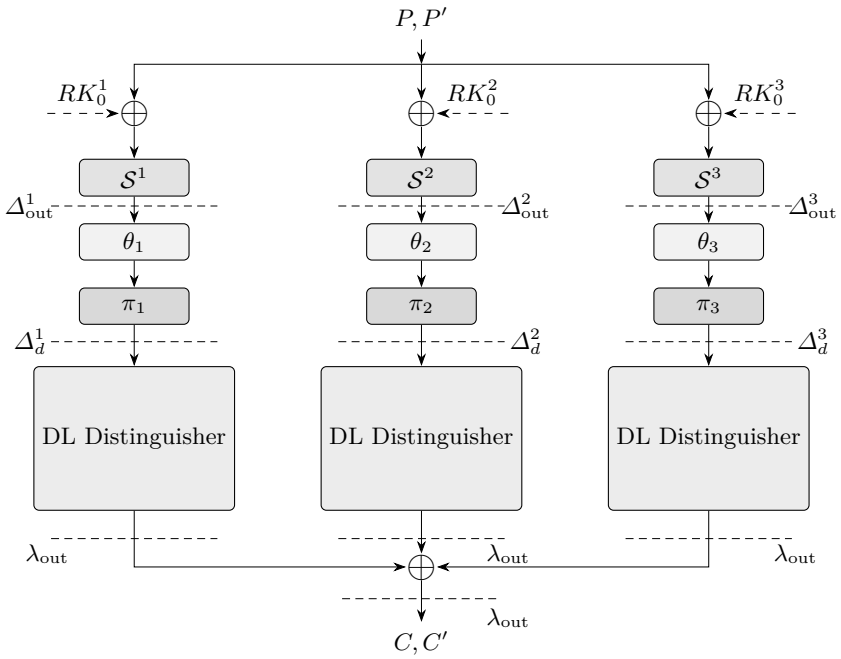

Fig. 3. One-round extension of the four-round DL distinguisher of Gleeok-128. The additions of RK_1^b and RC_1^b are absorbed into the distinguisher input.

The input differences of the 4-round distinguisher are

$$\begin{aligned}
\Delta^1_{\text{d}}: &\quad \Delta S_3^0 = \texttt{0x2},\ \Delta S_3^3 = \texttt{0x1},\ \Delta S_5^3 = \texttt{0x01},\\
\Delta^2_{\text{d}}: &\quad \Delta S_3^0 = \texttt{0x1},\ \Delta S_5^0 = \texttt{0x01},\ \Delta S_3^{13} = \texttt{0x2},\\
\Delta^3_{\text{d}}: &\quad \Delta S_4^2 = \texttt{0x1},\ \Delta S_4^{22} = \texttt{0x1},\ \Delta S_4^{28} = \texttt{0x1}.
\end{aligned}$$

To determine the differential transitions required in the prepended round, we propagate these differences backward through the branch-dependent linear layers $L_b = \pi_b \circ \theta_b$. This gives

$$\Delta^1_{\text{out}}:\ \Delta S_3^0 = \texttt{0x2}, \qquad \Delta^2_{\text{out}}:\ \Delta S_3^0 = \texttt{0x2}, \qquad \Delta^3_{\text{out}}:\ \Delta S_4^0 = \texttt{0x8},$$

where all unlisted S-box differences are zero and $L_b(\Delta^b_{\text{out}}) = \Delta^b_{\text{d}}$ for $b \in \{1, 2, 3\}$. The three branch distinguishers use the same output mask, $\lambda_{\text{out}} = \{\{13, \texttt{0xa0}\}\}$, which means the mask of the 13-th byte is `0xa0`.

Weak-key characterization. Let K be the master key and RK_0^b the whitening key of Branch b, where $b \in \{1, 2, 3\}$. The first S-box layer of Branch b takes $P \oplus RK_0^b$ as input. We consider plaintext pairs (P, P') with the common input difference

$$\Delta_P = P \oplus P' = \texttt{e0000000 00000000 00000000 00000000}.$$

This induces the transitions $\texttt{0x7} \rightarrow \texttt{0x2}$ in the active S_3 of Branches 1 and 2 and $\texttt{0xe} \rightarrow \texttt{0x8}$ in the active S_4 of Branch 3. Let χ_3 and χ_4 denote the corresponding admissible input sets, then we obtain

$$\chi_3 = \{x \mid S_3(x) \oplus S_3(x \oplus \texttt{0x7}) = \texttt{0x2}\} = \{\texttt{0x2}, \texttt{0x5}\},$$
$$\chi_4 = \{x \mid S_4(x) \oplus S_4(x \oplus \texttt{0xe}) = \texttt{0x8}\} = \{\texttt{0x3}, \texttt{0x4}, \texttt{0xa}, \texttt{0xd}\}.$$

For the same plaintext pair to satisfy the transitions in Branches 1 and 2, the active whitening-key slices must satisfy

$$RK_0^1 \oplus RK_0^2 \in \chi_3 \oplus \chi_3 = \{\texttt{0x0}, \texttt{0x7}\}.$$

Thus, for each of the 2^3 possible values of RK_0^1, two of the 2^3 values of RK_0^2 are compatible. To impose the Branch 3 condition, we extend RK_0^1 to four bits by defining $\overline{RK}_0^1 = RK_0^1 \parallel 0$. Compatibility with the active S_4 requires $\overline{RK}_0^1 \oplus RK_0^3 \in \{\texttt{0x0}, \texttt{0x1}, \texttt{0x6}, \texttt{0x7}, \texttt{0x8}, \texttt{0x9}, \texttt{0xe}, \texttt{0xf}\}$. Hence, eight of the 2^4 values of RK_0^3 are compatible with each RK_0^1. Since the three slices contain distinct master-key bits, the fraction of keys satisfying all three branch conditions is $1 \cdot \frac{2}{2^3} \cdot \frac{8}{2^4} = 2^{-3}$.

Key-recovery procedure. Among the 2^{10} local key guesses, 2^7 satisfy the three-branch compatibility conditions. For each of the eight possible plaintext pairs, $2 \cdot 2 \cdot 4 = 16$ compatible guesses select that pair and obtain identical counters, and the other guesses yield no valid pair. The threshold test therefore retains or rejects an entire compatible class.

Let $A = \{0, 1, 2, 3\}$ be the active plaintext-bit set, and let k denote a guess of the $N_t = 10$ involved master-key bits. Here, $RK_0^b(k)$ denotes the guessed whitening-key bits entering the active S-box of Branch b. The concrete key recovery procedure is shown in Algorithm 1.

Complexity analysis. Each structure contains 16 plaintexts and supplies exactly one pair for each compatible class, so $N = m$ for the correct weak-key class. Since each score represents 2^4 indistinguishable guesses, we set $a = 10 - 4 = 6$. With $|C_{\mathrm{DL}}| = 2^{-25.71}$, Eq. (1) gives $N = m \approx 2^{54.77}$. This gives $P_S \approx 85.07\%$. Including three additional plaintexts for verification, the data complexity is $D = 16m + 3 \approx 2^{58.77}$ chosen plaintexts.

For each structure, the filtering stage tests eight unordered pairs under all 2^{10} local key guesses. This requires $8m \cdot 2^{10} \approx 2^{67.77}$ local tests. Each test requires only the three active S-box computations.

For each retained equivalence class, the 2^4 local guesses must be combined with all 2^{246} values of the remaining master-key bits. Thus, the final exhaustive-search term is $2^4 \cdot 2^{246} = 2^{250}$ full-key trials. This term dominates the previous stages, so the time complexity is approximately $T \approx 2^{250}$ 5-round Gleeok-128 evaluations. The memory complexity is approximately 2^{17} bits, dominated by two arrays of 2^{10} 64-bit counters. The success probability and complexities are conditioned on the weak-key class, whose coverage is 2^{-3}.

Algorithm 1: Key recovery on 5-round Gleeok-128

Input: Encryption oracle E_K; number of structures m; target advantage a; plaintext difference Δ_P; target output differences Δ^b_{out} of the first S-box layer in each branch, $b \in \{1,2,3\}$; output mask λ_{out}

Output: The verified master key, or failure

Initialize $T[k] = V[k] = 0$ for all $k \in \{0,1\}^{N_t}$;

Query three distinct verification plaintexts and store the resulting plaintext–output pairs in $\mathcal{D}_{\text{ver}}$;

for $j = 1$ **to** m **do**

  Generate $\mathcal{P}_j$ by varying A and fixing the other bits uniformly;

  Store $L_j[P] \leftarrow E_K(P)$ for every $P \in \mathcal{P}_j$;

  foreach unordered pair $\{P, P'\} \subseteq \mathcal{P}_j$ satisfying $P \oplus P' = \Delta_P$ **do**

    foreach $k \in \{0,1\}^{N_t}$ **do**

      if the active S-box transition under $RK^b_0(k)$ has output difference Δ^b_{out} for every $b \in \{1,2,3\}$ **then**

        $T[k] \leftarrow T[k] + 1$;

        if $\lambda_{\text{out}} \cdot (L_j[P] \oplus L_j[P']) = 0$ **then** $V[k] \leftarrow V[k] + 1$;

$\mathcal{R} \leftarrow \left\{ k : T[k] > 0,\ 2\sqrt{T[k]}\left(\frac{V[k]}{T[k]} - \frac{1}{2}\right) > \Phi^{-1}(1 - 2^{-a}) \right\}$;

foreach $k \in \mathcal{R}$ and $z \in \{0,1\}^{256-N_t}$ **do**

  Form $\widehat{K}$ by assigning k to the involved master-key positions and z to the remaining positions;

  if $E_{\widehat{K}}(P) = C$ for every $(P, C) \in \mathcal{D}_{\text{ver}}$ **then return** $\widehat{K}$;

return failure;

4 Differential-Linear Cryptanalysis of ZIP Ciphers

The ZIP construction [14] is a two-branch sum-of-permutations construction for deriving low-latency pseudorandom functions from iterative block ciphers. Let $E = E_b \circ E_f$ be an n-bit block cipher, where E_f and E_b contain almost the same number of rounds. For $P \in \mathbb{F}_2^n$, ZIP-E is defined as

$$\text{ZIP-E}_{r_f, r_b}(P) = E_f(P) \oplus E_b^{-1}(P),$$

where r_f and r_b denote the number of rounds in the forward and backward branches, respectively. Both branches are keyed permutations, whereas their XOR need not be a permutation and is instead used as a PRF.

The use of the inverse permutation in the second branch is the main design principle of ZIP. As shown in [14], differential and linear trails of a sum $P \oplus Q$ are closely related to those of sequential compositions involving P, Q, and their inverses. By choosing the two branches from the forward- and inverse-round families of the same block cipher, ZIP can therefore inherit much of the existing cryptanalytic knowledge of the underlying cipher. Meanwhile, the two branches are evaluated in parallel, yielding a latency of roughly half that of the

corresponding sequential block cipher. The concrete instances considered here are ZIP-AES and the two GIFT-based variants ZIP-GIFT-64 and ZIP-GIFT-128.

Specification of ZIP-AES. ZIP-AES [14] is the AES-based instance of the ZIP construction. It is instantiated from AES-128 and operates on 128-bit inputs with a 128-bit master key. The full construction consists of two parallel five-round branches and is defined as

$$\text{ZIP-AES}_{5,5}(P) = \text{AES}_5(P) \oplus \text{AES}_5^{-1}(P).$$

More generally, the round-reduced variant $\text{ZIP-AES}_{r_f,r_b}$ is obtained by applying r_f forward AES rounds in the first branch and r_b inverse AES rounds in the second branch.

The forward branch uses the AES round function $R_{\text{AES}} = \text{MC} \circ \text{SR} \circ \text{SB}$, followed by round-key addition. Unlike standard AES, the MixColumns operation is retained in the last round of the forward branch. The backward branch applies the corresponding inverse round function $R_{\text{AES}}^{-1} = \text{SB}^{-1} \circ \text{SR}^{-1} \circ \text{MC}^{-1}$, and similarly retains the initial MC^{-1} operation.

Let $k_0 = k, k_1, \ldots, k_{10}$ be the round keys generated by the AES-128 key schedule. The forward branch is instantiated with $k_0, k_1, \ldots, k_5$, whereas the backward branch uses $k_6, k_7, \ldots, k_{10}, 0^{128}$.

Specifications of ZIP-GIFT-128 and ZIP-GIFT-64. ZIP-GIFT-128 [24,3] is the main GIFT-based instance considered in this paper. It is instantiated from GIFT-128 and operates on 128-bit inputs with a 128-bit master key. GIFT-128 consists of 40 rounds. It is decomposed as GIFT-128 $= E_b \circ E_f$, where E_f and E_b denote the forward and backward 20 rounds, respectively. Accordingly, the full ZIP-GIFT-128 construction is defined as

$$\text{ZIP-GIFT-128}_{20,20}(P) = \text{GIFT-128}_{20}(P) \oplus \text{GIFT-128}_{20}^{-1}(P).$$

ZIP-GIFT-64 is defined analogously from the 28-round GIFT-64 cipher. It operates on 64-bit inputs with a 128-bit master key, and its full construction consists of two 14-round branches, i.e., $\text{ZIP-GIFT-64}_{14,14}(P) = \text{GIFT-64}_{14}(P) \oplus \text{GIFT-64}_{14}^{-1}(P)$. In both variants, a forward round applies SubCells and PermBits before adding the round key, while the backward branch applies the inverse operations. The corresponding GIFT key schedules run in opposite directions, and neither variant uses pre-whitening. Following Nageler et al. [24], we use an equivalent-key representation to simplify the differential and linear propagation.

4.1 DL Key-Recovery Attack on ZIP-AES

We apply the differential-linear sandwich method from Section 2.1 to ZIP-AES. Under the compatibility constraints, we obtain a $(3, 3)$-round DL distinguisher with

$$\Delta_{\text{in}} = \texttt{0x00ca0000 00000000 00000000 00000000},$$
$$\lambda_{\text{out}} = \texttt{0x00000000 00000000 00000000 00880000}.$$

The correlations of the forward and backward branches are $|C_f| = 2^{-31.61}$, and $|C_b| = 2^{-30.36}$, respectively. Hence, the correlation of the resulting ZIP-AES distinguisher is $|C_{\mathrm{DL}}| = |C_f C_b| = 2^{-61.97}$.

Proposition 1 [14]. *Let* $\mathrm{F} = (P_2 \circ P_1) \oplus (Q_2 \circ Q_1)$. *Consider a DL key-recovery attack with input differences* δ_P *and* δ_Q *for* P_2 *and* Q_2*, respectively, and common output mask* α*. After guessing the required key material of* P_1 *and* Q_1*, the attack can work only if*

$$\big(\mathrm{Aut}_{P_2}(\delta_P, \alpha) \cdot \mathrm{Aut}_{Q_2}(\delta_Q, \alpha)\big)^{-2} < 2^n \cdot \mathrm{Prob}\big(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q\big).$$

Let p denote the probability of the differential propagation through the prepended round and C_{DL} the correlation of the remaining DL distinguisher. For ZIP-AES, Proposition 1 gives the criterion $p\,|C_{\mathrm{DL}}|^2 > 2^{-128}$.

Following this criterion, our search finds a $(2, 3)$-round DL distinguisher that can be extended by prepending one AES round to the forward branch. The resulting attack covers $(3, 3)$ rounds. The active input-byte difference is $\Delta_{\mathrm{in}} = \texttt{0x5b}$, and the distinguisher has correlation $|C_{\mathrm{DL}}| = 2^{-55.42}$. In the prepended round, the transition from the input difference `0x5b` to the S-box output difference `0xd9` holds with probability $p = 2^{-7}$. Figure 4 illustrates the resulting attack.

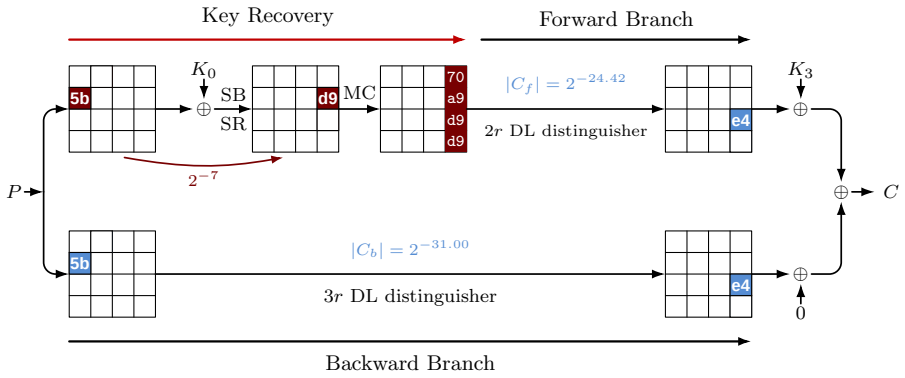

Fig. 4. Differential-linear key-recovery attack on ZIP-AES.

The external input difference Δ_{in} is applied to both branches. In the forward branch, it first propagates through the prepended round and then enters the two-round forward part of the DL distinguisher. In the backward branch, it directly enters the three-round backward part. The common output mask λ_{out} is applied after the final XOR.

Key-recovery procedure. The attack targets the first-round key byte $K_0[1]$. The procedure is as follows.

1. Select $2^{121.42}$ plaintexts P. For each P, form $P' = P \oplus \Delta_{\text{in}}$. Query the encryption oracle to obtain the corresponding output pair (C, C').
2. The S-box transition $\texttt{0x5b} \to \texttt{0xd9}$ is realized by the two pre-S-box values $\texttt{0x17}$ and $\texttt{0x4c}$. Therefore, each plaintext pair suggests the two candidates $P[1] \oplus \texttt{0x17}$ and $P[1] \oplus \texttt{0x4c}$ for $K_0[1]$.
3. For each suggested candidate k, increase T_k by one. If the corresponding output pair satisfies $\langle \lambda_{\text{out}}, C \oplus C' \rangle = 0$, increase V_k by one as well.
4. For each candidate k with $T_k > 0$, retain k in $\mathcal{R}$ if $V_k/T_k - 1/2 > \theta_k$, where θ_k is determined by T_k and the target advantage a.
5. For every $k \in \mathcal{R}$, exhaustively search the remaining 120 master-key bits and verify each resulting full-key candidate using three additional plaintext–output pairs.

Complexity analysis. The candidates $K_0[1]$ and $K_0[1] \oplus \texttt{0x5b}$ obtain identical counters. Therefore, the statistical phase can recover at most seven independent bits of $K_0[1]$, and we set $a = 7$. For a target success probability $P_S \approx 85.07\%$, Eq. (1) gives the required number of useful pairs as approximately $\approx 2^{114.42}$. Since only a fraction $p = 2^{-7}$ of the external pairs realizes the required transition in the prepended round, the number of external pairs is $N = p^{-1} \cdot 2^{114.42} \approx 2^{121.42}$. Including the three plaintexts used for final verification, the data complexity is $D = 2N + 3 \approx 2^{122.42}$ chosen plaintexts.

An advantage of $a = 7$ leaves $2^{128-7} = 2^{121}$ master-key candidates for the final exhaustive search. The counter updates require only constant work per processed pair and are dominated by data collection. Hence, the total time complexity is $T = D + 2^{121} \approx 2^{122.88}$ ZIP-AES evaluations.

The two counter arrays T and V each contain 2^8 entries. Using 128-bit counters, their total memory requirement is $2 \cdot 2^8 \cdot 128 = 2^{16}$ bits.

4.2 DL Key-Recovery Attack on ZIP-GIFT-128

Our search first yields a stand-alone $(10, 10)$-round DL distinguisher for ZIP-GIFT-128 with $r_d = 3$, $r_m = 7$, and correlation $|C_{\text{DL}}| = 2^{-52.75}$. Its input difference and output mask are

$$\Delta^d_{\text{in}} = \texttt{0x0020 0000 0010 0000 0000 0000 0000 0000},$$
$$\lambda^d_{\text{out}} = \texttt{0x0000 0000 0000 0000 0000 0000 1000 0000}.$$

For key recovery, we use an $(8, 8)$-round DL core and prepend two differential rounds to each branch, resulting in an attack on $(10, 10)$-round ZIP-GIFT-128.

Notation. Let P, P' be two 128-bit plaintexts and K the 128-bit master key. For a state X, let $X[i]$ denote bit i and $X[a..b]$ the value formed by bits $a, \ldots, b$, with $X[a]$ as the least significant bit. Thus, nibble j is $X[4j..4j + 3]$.

Let S be the GIFT S-box, S^{-1} its inverse, and let L be the GIFT-128 bit permutation. We index the first forward and backward rounds by 0 and -1, respectively, and denote the 128-bit round key of round r by RK_r.

Both branches start from $X_0 = P$. In the forward branch, X_i, Y_i, and EY_i denote the round input, the S-box output, and the linear-layer output, respectively. Since GIFT-128 has no pre-whitening, the first two rounds are

$$X_0 \xrightarrow{S} Y_0 \xrightarrow{L} EY_0 \xrightarrow{\oplus RK_0} X_1 \xrightarrow{S} Y_1 \xrightarrow{L} EY_1 \xrightarrow{\oplus RK_1} X_2.$$

In the backward branch, X_{-i+1} is the input of round $-i$, EX_{-i+1} is the inverse-linear-layer output, and Y_{-i} is the state after adding the equivalent round key. The first two rounds are

$$X_0 \xrightarrow{L^{-1}} EX_0 \xrightarrow{\oplus RK_{-1}} Y_{-1} \xrightarrow{S^{-1}} X_{-1} \xrightarrow{L^{-1}} EX_{-1} \xrightarrow{\oplus RK_{-2}} Y_{-2} \xrightarrow{S^{-1}} X_{-2}.$$

Characteristic. Let $\Delta_{\text{in}} = P \oplus P'$ be the common plaintext difference, and let $\Delta_d^f = \Delta X_2$ and $\Delta_d^b = \Delta X_{-2}$ be the input differences of the forward and backward eight-round DL distinguishers. Our search gives

$$\begin{aligned}\Delta_{\text{in}} &= \texttt{0000 0000 0000 0000 0000 0202 0000 0000},\\ \Delta_d^f &= \texttt{0000 0000 0100 0100 0000 0000 0000 0000},\\ \Delta_d^b &= \texttt{0000 0000 0000 0000 0000 0000 0000 0120}.\end{aligned}$$

Table 4 gives the two-round differential characteristics from Δ_{in} to Δ_d^f and Δ_d^b. Both eight-round DL distinguishers use the output mask

$$\lambda_{\text{out}} = \texttt{0000 0000 0000 0000 0020 0000 0000 0000},$$

where nibble 13 is 2 and all other nibbles are zero. The round-based method gives $|C_f| \approx 2^{-18.56}$ and $|C_b| \approx 2^{-20.83}$. Under the branch-independence approximation, the joint correlation is $|C_{\text{DL}}| = |C_f C_b| \approx 2^{-39.39}$.

The two prepended differential characteristics have joint probability $p = 2^{-19}$. Then we obtain $p|C_{\text{DL}}|^2 = 2^{-19}|C_{\text{DL}}|^2 \approx 2^{-97.77} > 2^{-128}$. Thus, the necessary data-availability condition is satisfied under the correlation model.

Plaintext structures. The structures vary the plaintext positions needed to test both prefix characteristics. In the forward branch, round 0 activates S_8 and S_{10}. After L, the active inputs of S_{10} and S_{26} in round 1 also depend on the outputs of S_9 and S_{11}. The forward characteristic therefore involves the four plaintext nibbles $P[32..35]$, $P[36..39]$, $P[40..43]$, and $P[44..47]$. We set $\mathcal{A}_f = \{32, 33, \ldots, 47\}$.

In the backward branch, the first inverse round involves S_0^{-1}, S_8^{-1}, S_{16}^{-1}, and S_{24}^{-1}. Through the inverse linear layer L^{-1}, their input bits come from the plaintext positions

$$\mathcal{A}_b = \{0, 33, 66, 99,\ 8, 41, 74, 107, 16, 49, 82, 115,\ 24, 57, 90, 123\}.$$

Since $\mathcal{A}_f \cap \mathcal{A}_b = \{33, 41\}$, the complete set of variable positions is $\mathcal{A} = \mathcal{A}_f \cup \mathcal{A}_b$, namely

$$\{32, 33, \ldots, 47\} \cup \{0, 8, 16, 24, 49, 57, 66, 74, 82, 90, 99, 107, 115, 123\}.$$

Table 4. $(2,2)$-round key-recovery characteristics of ZIP-GIFT-128 with joint probability 2^{-19}.

Branch	State	Difference
Forward	ΔX_0	00000000000000000000020200000000
	ΔY_0	0000000000000000000a0a00000000
	ΔX_1	00000a000000000000000a0000000000
	ΔY_1	00000100000000000000010000000000
	ΔX_2	00000000010001000000000000000000
Backward	ΔX_0	00000000000000000000020200000000
	ΔEX_0	00000000000000000000000200000002
	ΔX_{-1}	0000000000000000000000c00000006
	ΔEX_{-1}	0000000000000000000000000000c60
	ΔX_{-2}	00000000000000000000000000000120

For $\alpha, \beta \in \mathbb{F}_2^4$, define the input solution sets $\mathrm{XDDT}^f_{\alpha\to\beta} = \{x \in \mathbb{F}_2^4 : S(x) \oplus S(x \oplus \alpha) = \beta\}$, and $\mathrm{XDDT}^b_{\alpha\to\beta} = \{x \in \mathbb{F}_2^4 : S^{-1}(x) \oplus S^{-1}(x \oplus \alpha) = \beta\}$.

In the first forward round, S_8 and S_{10} must realize the transition $\mathtt{2} \to \mathtt{a}$. Since $\mathrm{XDDT}^f_{\mathtt{2}\to\mathtt{a}} = \{\mathtt{9}, \mathtt{b}\}$, we restrict $P[32..35]$, $P[40..43] \in \{\mathtt{9}, \mathtt{b}\}$.

In the second forward round, S_{10} and S_{26} must realize the transition $\mathtt{a} \to \mathtt{1}$, for which $\mathrm{XDDT}^f_{\mathtt{a}\to\mathtt{1}} = \{\mathtt{3}, \mathtt{5}, \mathtt{9}, \mathtt{f}\}$. Equivalently, x belongs to this set if and only if $x[0] = 1$ and $x[1] \oplus x[2] \oplus x[3] = 1$. The inputs of these two S-boxes are $X_1[40..43]$ and $X_1[104..107]$. From the linear layer and the round-key injection positions, we have $X_1[40] = Y_0[44]$, $X_1[104] = Y_0[36]$. We therefore impose $Y_0[36] = Y_0[44] = 1$.; the remaining parity conditions are tested under each key guess.

In the first backward round, S_0^{-1} and S_8^{-1} must realize the transitions $\mathtt{2} \to \mathtt{6}$ and $\mathtt{2} \to \mathtt{c}$, respectively. The corresponding solution sets are $\mathrm{XDDT}^b_{\mathtt{2}\to\mathtt{6}} = \{\mathtt{1}, \mathtt{3}, \mathtt{4}, \mathtt{6}\}$, and $\mathrm{XDDT}^b_{\mathtt{2}\to\mathtt{c}} = \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. The relevant high bits are unaffected by the equivalent round-key addition, so we impose $P[99] = Y_{-1}[3] = 0$, $P[107] = Y_{-1}[35] = 1$.

Under these restrictions, the 14 plaintext bits at positions $\{0, 8, 16, 24, 33, 41, 49, 57, 66, 74, 82, 90, 115, 123\}$ are chosen independently. Bits 33 and 41 select between `9` and `b` for $P[32..35]$ and $P[40..43]$, respectively. The remaining variable nibbles are $P[36..39]$ and $P[44..47]$. Each of these two nibbles ranges over $\{\mathtt{0}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{9}, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$, giving eight choices per nibble. All other plaintext bits are fixed within a structure. Hence, each structure contains $2^{14} \cdot 8^2 = 2^{20}$ plaintexts.

Since Δ_{in} is nonzero only at bits 33 and 41, $P' = P \oplus \Delta_{\mathrm{in}}$ lies in the same structure as P. We keep only representatives with $P[33] = 0$, yielding 2^{19} distinct pairs per structure.

Pair filtering. For each representative pair (P, P') with $P[33] = 0$, we test the remaining prefix conditions.

In the second forward round, these conditions are $X_1[41] \oplus X_1[42] \oplus X_1[43] = 1, X_1[105] \oplus X_1[106] \oplus X_1[107] = 1$. Since $X_1 = EY_0 \oplus RK_0$, the bit permutation

and round-key injection give

a	$X_1[40+a]$	$X_1[104+a]$
1	$Y_0[33] \oplus K[10]$	$Y_0[41] \oplus K[26]$
2	$Y_0[38] \oplus K[74]$	$Y_0[46] \oplus K[90]$
3	$Y_0[43]$	$Y_0[35]$

where a is the bit position within a nibble. For $P[32..35] \in \{\mathtt{9}, \mathtt{b}\}$ and $P[40..43] \in \{\mathtt{9}, \mathtt{b}\}$, the values $S(\mathtt{9}) = \mathtt{d}$ and $S(\mathtt{b}) = \mathtt{7}$, together with $P[33] = 0$, imply $Y_0[33] = 0$, $Y_0[35] = 1$, $Y_0[41] = P[41]$, $Y_0[43] = 1 \oplus P[41]$. Substituting these values into the two parity conditions yields

$$\begin{cases} K[10] \oplus K[74] = P[41] \oplus Y_0[38], \\ K[26] \oplus K[90] = P[41] \oplus Y_0[46]. \end{cases} \tag{2}$$

Thus, each pair determines its two compatible forward key parities.

In the first backward round, the inverse permutation and key schedule give

j	$Y_{-1}[4j]$	$Y_{-1}[4j+1]$	$Y_{-1}[4j+2]$	$Y_{-1}[4j+3]$
0	$P[0]$	$P[33] \oplus K[108]$	$P[66] \oplus K[48]$	$P[99]$
8	$P[8]$	$P[41] \oplus K[110]$	$P[74] \oplus K[50]$	$P[107]$
16	$P[16]$	$P[49] \oplus K[96]$	$P[82] \oplus K[52]$	$P[115]$
24	$P[24]$	$P[57] \oplus K[98]$	$P[90] \oplus K[54]$	$P[123]$

For each guess of the eight key bits in the table, require

$$\begin{cases} Y_{-1}[0..3] \in \mathrm{XDDT}^b_{2\to\mathtt{6}} = \{\mathtt{1}, \mathtt{3}, \mathtt{4}, \mathtt{6}\}, \\ Y_{-1}[32..35] \in \mathrm{XDDT}^b_{2\to\mathtt{c}} = \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}. \end{cases} \tag{3}$$

For surviving guesses, compute $X_{-1}[4j..4j+3] = S^{-1}(Y_{-1}[4j..4j+3])$ for $j \in \{0, 8, 16, 24\}$. Nibbles 16 and 24 have zero difference, but their values are needed in the next round. The two active inverse S-box inputs of the second backward round are

a	$Y_{-2}[4+a]$	$Y_{-2}[8+a]$
0	$X_{-1}[96]$	$X_{-1}[64]$
1	$X_{-1}[1] \oplus K[68]$	$X_{-1}[97] \oplus K[86]$
2	$X_{-1}[34] \oplus K[8]$	$X_{-1}[2] \oplus K[0]$
3	$X_{-1}[67]$	$X_{-1}[35]$

They must realize $\mathtt{6} \to \mathtt{2}$ and $\mathtt{c} \to \mathtt{1}$, respectively. Their solution sets are $\mathrm{XDDT}^b_{\mathtt{6}\to\mathtt{2}} = \{\mathtt{9}, \mathtt{a}, \mathtt{c}, \mathtt{f}\} = \{x : x[3] = 1,\ x[0] \oplus x[1] \oplus x[2] = 1\}$, and $\mathrm{XDDT}^b_{\mathtt{c}\to\mathtt{1}} = \{\mathtt{7}, \mathtt{b}\} = \{x : x[0] = x[1] = 1,\ x[2] \oplus x[3] = 1\}$. The first set requires $Y_{-2}[7] = 1$ and $Y_{-2}[4] \oplus Y_{-2}[5] \oplus Y_{-2}[6] = 1$. The second requires $Y_{-2}[8] = Y_{-2}[9] = 1$ and $Y_{-2}[10] \oplus Y_{-2}[11] = 1$. Substitution from the table gives

$$\begin{cases} X_{-1}[64] = X_{-1}[67] = 1, \\ K[68] \oplus K[8] = 1 \oplus X_{-1}[96] \oplus X_{-1}[1] \oplus X_{-1}[34], \\ K[86] = 1 \oplus X_{-1}[97], \\ K[0] = 1 \oplus X_{-1}[2] \oplus X_{-1}[35]. \end{cases} \tag{4}$$

Together, Eqs. (3) and (4) are necessary and sufficient for the backward characteristic. Although the filter contains 13 key-dependent bits and parities, exhaustive enumeration shows that every four guesses select the same pairs. The 2^{13} guesses therefore form 2^{11} classes represented by

$$\begin{aligned} K_{\mathrm{eq}} = \big(&K[48], K[50], K[52], K[54], K[86], K[96], K[98], \\ &K[10] \oplus K[74],\ K[26] \oplus K[90], \\ &K[0] \oplus K[108] \oplus K[110], \\ &K[0] \oplus K[8] \oplus K[68]\big). \end{aligned} \tag{5}$$

The 11 components of K_{eq} are independent. Hence there are 2^{11} equivalent-subkey candidates, each consistent with 2^{117} master keys.

For fixed K_{eq}, the backward conditions leave 2^6 choices for the 13 local variables. The two relations in Eq. (2) fix $Y_0[38]$ and $Y_0[46]$. Since the structure already imposes $Y_0[36] = Y_0[44] = 1$, each of the two output nibbles $Y_0[36..39]$ and $Y_0[44..47]$ has two fixed bits and hence 2^2 values. Since S is bijective, each corresponding plaintext nibble also has 2^2 values. Thus, every equivalent-subkey candidate retains $2^6 \cdot 2^2 \cdot 2^2 = 2^{10}$ pairs in one structure.

Key-recovery procedure. Let $\mathcal{K}_{\mathrm{eq}} = \mathbb{F}_2^{11}$ be the space of equivalent-subkey candidates. The attack uses only the observable PRF output $C(P)$ and proceeds as follows.

1. Build a lookup table for one generic structure. For each pair representative P with $P[33] = 0$, store the four values $\kappa \in \mathcal{K}_{\mathrm{eq}}$ under which the pair satisfies both prefix characteristics. The table is independent of the 98 fixed plaintext bits and the secret key.
2. Choose m plaintext structures by fixing the 98 bits outside $\mathcal{A}$ independently and uniformly. Query all 2^{20} plaintexts and form the 2^{19} pairs $P' = P \oplus \Delta_{\mathrm{in}},\ P[33] = 0$.
3. For each pair (P, P'), use the table to obtain its four compatible candidates. For each such κ, increment V_κ if $\langle \lambda_{\mathrm{out}}, C(P) \oplus C(P') \rangle = 0$. Each κ receives exactly 2^{10} pairs per structure, so its total sample size is $T_\kappa = m2^{10}$.
4. For each $\kappa \in \mathcal{K}_{\mathrm{eq}}$, if $V_\kappa/T_\kappa - 1/2 > \theta_\kappa$, retain κ, where θ_κ is chosen according to T_κ and the target advantage a.
5. For every retained κ, enumerate the 2^{117} consistent master keys and verify them with two additional plaintext–output pairs.

Complexity analysis. We set $a = 11$. With the DL correlation $|C_{\mathrm{DL}}| \approx 2^{-39.39}$, Eq. (1) gives $N \approx 2^{83.00}$. Thus, the correct equivalent-subkey class is retained with probability $P_S \approx 85.07\%$. Since each structure supplies 2^{10} pairs per candidate, $m = N/2^{10} \approx 2^{73.00}$. Including two verification plaintexts, the data complexity is $D = m2^{20} + 2 \approx 2^{93.00}$ chosen plaintexts.

The statistical phase performs $m2^{19} \cdot 4 = m2^{21} \approx 2^{94.00}$ counter updates. Under the wrong-key model, advantage a leaves 2^{11-a} equivalent-subkey classes

on average. The test does not force a unique winner: all classes above the threshold are searched, giving $2^{11-a} \cdot 2^{117} = 2^{128-a} = 2^{117}$ full-key trials. This term dominates, so the time complexity is $T \approx 2^{117}$ full $(10, 10)$-round ZIP-GIFT-128 evaluations.

The memory is dominated by the table for the pair filter. It has 2^{19} entries and stores four 16-bit indices per entry, requiring 2^{25} bits. The 2^{11} 128-bit counters require 2^{18} bits; T_κ need not be stored because it is fixed. Therefore, $M = 2^{25} + 2^{18} \approx 2^{25.01}$ bits.

Discussion. The main restriction is branch compatibility. Both branches receive the same plaintext pair, their DL output masks must match, and the guessed prefixes must map that pair to both prescribed internal differences. These joint conditions reduce the number of usable pairs and can make a strong distinguisher unsuitable for key recovery.

For comparison, Appendix C gives an $(8, 6)$-round attack on ZIP-GIFT-64 that uses a DL distinguisher in the forward branch and a linear approximation in the backward branch. This relaxes the joint input-difference condition, but its correlation contains the squared linear factor $|c_{\mathrm{L}}|^2$, and the 64-bit domain limits the available data. Its round coverage therefore remains below the differential and linear attacks of Nageler et al. [24].

5 Conclusion

We extend the reported differential-linear distinguishers for Orthros from 7 to 9 rounds and for Gleeok-128 from 4 to 5 rounds. We also obtain a 6-round distinguisher for Gleeok-256. For ZIP-AES, ZIP-GIFT-64, and ZIP-GIFT-128, our distinguishers cover $(3, 3)$, $(7, 7)$, and $(10, 10)$ rounds, respectively.

For key recovery, we give a differential-linear attack on $(3, 3)$-round ZIP-AES with data and time complexities of approximately $2^{122.42}$ chosen plaintexts and $2^{122.88}$ ZIP-AES evaluations. For $(10, 10)$-round ZIP-GIFT-128, the corresponding estimates are 2^{93} chosen plaintexts and 2^{117} full PRF evaluations. We further obtain weak-key attacks on 9-round Orthros, 5-round Gleeok-128, and 6-round Gleeok-256, covering fractions $2^{-2.87}$, 2^{-3}, and 2^{-1} of their respective key spaces.

These results use existing trail search and round-based correlation estimation. Key recovery additionally requires compatible branch masks and jointly reachable internal differences under the guessed subkeys. We also obtain an $(8, 6)$-round ZIP-GIFT-64 attack using a differential-linear forward branch and a linear backward branch. Extending the attacks to larger key classes and additional rounds remains open.

Our improvements are measured directly against previous differential-linear results: we extend the reported round coverage for Orthros and Gleeok and give compatible differential-linear distinguishers and key-recovery attacks for the ZIP instances.

Acknowledgements. We thank Chunning Zhou for independently checking our round-based correlation estimates and for pointing out that jointly treating the

masked S-boxes in the final round yields a refined correlation estimate for the five-round forward distinguisher used in the key-recovery attack in Appendix C.

References

1. Anand, R., Banik, S., Caforio, A., Ishikawa, T., Isobe, T., Liu, F., Minematsu, K., Rahman, M., Sakamoto, K.: Gleeok: A family of low-latency prfs and its applications to authenticated encryption. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2024**(2), 545–587 (2024). https://doi.org/10.46586/TCHES.V2024.I2.545-587
2. Banik, S., Isobe, T., Liu, F., Minematsu, K., Sakamoto, K.: Orthros: A low-latency PRF. IACR Trans. Symmetric Cryptol. **2021**(1), 37–77 (2021). https://doi.org/10.46586/TOSC.V2021.I1.37-77
3. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 321–345. Springer (2017). https://doi.org/10.1007/978-3-319-66787-4_16
4. Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: A new tool for differential-linear cryptanalysis. In: Advances in Cryptology – EUROCRYPT 2019, Part I. Lecture Notes in Computer Science, vol. 11476, pp. 313–342. Springer (2019). https://doi.org/10.1007/978-3-030-17653-2_11
5. Bellare, M., Impagliazzo, R.: A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. IACR Cryptol. ePrint Arch. **1999**, 24 (1999), http://eprint.iacr.org/1999/024
6. Bellare, M., Krovetz, T., Rogaway, P.: Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science, vol. 1403, pp. 266–280. Springer (1998). https://doi.org/10.1007/BFB0054132
7. 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. Lecture Notes in Computer Science, vol. 13090, pp. 36–66. Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_2
8. 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
9. Chen, S., Xie, P., Xu, S., Feng, X., Xiang, Z., Zeng, X.: Cryptanalysis of gleeok-128. Des. Codes Cryptogr. **94**(7), 145 (2026). https://doi.org/10.1007/S10623-026-01879-X
10. Cogliati, B., Seurin, Y.: EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 121–149. Springer (2016). https://doi.org/10.1007/978-3-662-53018-4_5

11. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings. Lecture Notes in Computer Science, vol. 1008, pp. 275–285. Springer (1994). https://doi.org/10.1007/3-540-60590-8_21
12. Dinur, I.: Tight indistinguishability bounds for the XOR of independent random permutations by fourier analysis. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14651, pp. 33–62. Springer (2024). https://doi.org/10.1007/978-3-031-58716-0_2
13. Dutta, A., Nandi, M., Saha, A.: Proof of mirror theory for $\xi_{\max} = 2$. IEEE Trans. Inf. Theory **68**(9), 6218–6232 (2022). https://doi.org/10.1109/TIT.2022.3171178
14. Flórez-Gutiérrez, A., Grassi, L., Leander, G., Sibleyras, F., Todo, Y.: General practical cryptanalysis of the sum of round-reduced block ciphers and ZIP-AES. In: Chung, K., Sasaki, Y. (eds.) Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part IX. Lecture Notes in Computer Science, vol. 15492, pp. 280–311. Springer (2024). https://doi.org/10.1007/978-981-96-0947-5_10
15. Hadipour, H., Derbez, P., Eichlseder, M.: Revisiting differential-linear attacks via a boomerang perspective with application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT. In: Advances in Cryptology – CRYPTO 2024, Part IV. Lecture Notes in Computer Science, vol. 14923, pp. 38–72. Springer (2024). https://doi.org/10.1007/978-3-031-68385-5_2
16. Hadipour, H., Todo, Y., Rahman, M., Eichlseder, M., Anand, R., Isobe, T.: Turning multiple key-dependent attacks into universal attacks. IACR Cryptol. ePrint Arch. **2025**, 1996 (2025), https://eprint.iacr.org/2025/1996
17. 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. Lecture Notes in Computer Science, vol. 16546, pp. 152–180. Springer (2026). https://doi.org/10.1007/978-3-032-25333-0_6
18. Hu, K., Zhang, C., Chang, C., Zhang, J., Wang, M., Peyrin, T.: Unlocking mix-basis potential: Geometric approach for combined attacks. 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. Lecture Notes in Computer Science, vol. 16004, pp. 293–334. Springer (2025). https://doi.org/10.1007/978-3-032-01901-1_10
19. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 17–25. Springer (1994). https://doi.org/10.1007/3-540-48658-5_3
20. Li, M., Sun, L., Wang, M.: Automated key recovery attacks on round-reduced orthros. In: Batina, L., Daemen, J. (eds.) Progress in Cryptology - AFRICACRYPT 2022: 13th International Conference on Cryptology in Africa, AFRICACRYPT

2022, Fes, Morocco, July 18-20, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13503, pp. 189–213. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-17433-9_9
21. Liu, M., Lu, X., Lin, D.: Differential-linear cryptanalysis from an algebraic perspective. In: Advances in Cryptology – CRYPTO 2021, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 247–277. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_9
22. Lucks, S.: The sum of prps is a secure PRF. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 470–484. Springer (2000). https://doi.org/10.1007/3-540-45539-6_34
23. Mennink, B., Neves, S.: Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 556–583. Springer (2017). https://doi.org/10.1007/978-3-319-63697-9_19
24. Nageler, M., Chakraborty, D., Scherer, S., Eichlseder, M.: Key recovery attacks on ZIP ciphers: Application to ZIP-AES and ZIP-GIFT. IACR Trans. Symmetric Cryptol. **2026**(2) (2026). https://doi.org/10.46586/TOSC.A0LMPAK8NNY
25. Patarin, J.: Introduction to mirror theory: Analysis of systems of linear equalities and linear non equalities for cryptography. IACR Cryptol. ePrint Arch. **2010**, 287 (2010), http://eprint.iacr.org/2010/287
26. Peng, T., Zhang, W., Weng, J., Ding, T.: New approaches for estimating the bias of differential-linear distinguishers. In: Advances in Cryptology – CRYPTO 2024, Part IV. Lecture Notes in Computer Science, vol. 14923, pp. 174–205. Springer (2024). https://doi.org/10.1007/978-3-031-68385-5_6
27. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008). https://doi.org/10.1007/S00145-007-9013-7
28. Taka, K., Ishikawa, T., Sakamoto, K., Isobe, T.: An efficient strategy to construct a better differential on multiple-branch-based designs: Application to orthros. In: Rosulek, M. (ed.) Topics in Cryptology - CT-RSA 2023 - Cryptographers' Track at the RSA Conference 2023, San Francisco, CA, USA, April 24-27, 2023, Proceedings. Lecture Notes in Computer Science, vol. 13871, pp. 277–304. Springer (2023). https://doi.org/10.1007/978-3-031-30872-7_11
29. Wang, S., Liu, M., Hou, S., Lin, D.: Differential-linear cryptanalysis of GIFT family and gift-based ciphers. IACR Commun. Cryptol. **1**(1), 13 (2024). https://doi.org/10.62056/A6N5TXOL7

A Additional Remarks on the Differential-Linear Attack on Orthros

A.1 Nine-Round Differential-Linear Characteristic for Orthros

This subsection gives the $(3+5+1)$-round differential-linear distinguisher used for the nine-round Orthros result, which means it is composed of 3 differential rounds, 5 middle rounds and 1 linear round respectively. The differential propagation of two branches are shown in Table 5.

Table 5. Three-round differential prefixes of the Orthros distinguisher.

Branch	Round	Difference	Value	Probability
1	1	ΔS^1_{in}	0000 0000 0000 0000 0000 0000 0002 0600	2^{-5}
		ΔS^1_{out}	0000 0000 0000 0000 0000 0000 0004 0400	
	2	ΔS^2_{in}	0000 0000 0000 0044 0000 0000 0000 0000	2^{-6}
		ΔS^2_{out}	0000 0000 0000 0082 0000 0000 0000 0000	
	3	ΔS^3_{in}	0000 0000 0000 0110 0000 0000 0000 0000	2^{-6}
		ΔS^3_{out}	0000 0000 0000 0120 0000 0000 0000 0000	
2	1	ΔS^1_{in}	0000 0000 0000 0000 0000 0000 0002 0600	2^{-6}
		ΔS^1_{out}	0000 0000 0000 0000 0000 0000 0008 0100	
	2	ΔS^2_{in}	0000 0000 2200 0000 0000 0000 0000 0000	2^{-6}
		ΔS^2_{out}	0000 0000 8800 0000 0000 0000 0000 0000	
	3	ΔS^3_{in}	0000 0000 0000 0000 0000 8800 0000 0000	2^{-6}
		ΔS^3_{out}	0000 0000 0000 0000 0000 4100 0000 0000	

Let $\{i,\mathtt{x}\}$ means that nibble i has the nonzero difference or mask $\mathtt{x}$, and all unlisted nibbles are zero. After the linear layer of Round 3, the two output differences propagate to the inputs of Round 4 as $\Delta^1_M = \{\{24,\mathtt{0x4}\},\{25,\mathtt{0x4}\}\}$ and $\Delta^2_M = \{\{9,\mathtt{0x1}\},\{11,\mathtt{0x1}\}\}$, respectively. Thus, the differential-prefix probabilities are $p^1_D = 2^{-5-6-6} = 2^{-17}$ and $p^2_D = 2^{-6-6-6} = 2^{-18}$, giving $p^1_D p^2_D = 2^{-35}$.

The common output mask of the distinguisher is

$$\lambda_{\text{out}} = \mathtt{0000\ 0000\ ee0e\ 0000\ 0000\ 0000\ 0000\ 0000}.$$

For the five-round middle part, we evaluate all compatible output-mask pairs under the fixed input differences Δ^1_m and Δ^2_m. Table 6 lists the 13 contributions retained by the round-based evaluation. Here $c_{m,i}$ denotes the combined middle correlation of the two branches for candidate i.

Table 6. Round-based correlation cluster over Rounds 4–8 of Orthros.

λ_m^1	λ_m^2	$\log_2 \lvert c_{m,i} \rvert$
{{0,0xe}}	{{13,0x6}}	−17.62
{{0,0xe}}	{{13,0xb}}	−17.62
{{0,0xe}}	{{13,0xe}}	−17.62
{{0,0xe}}	{{13,0x3}}	−17.62
{{0,0xb}}	{{13,0x6}}	−18.09
{{0,0xb}}	{{13,0xe}}	−18.09
{{0,0xb}}	{{13,0x3}}	−18.09
{{0,0x3}}	{{13,0x6}}	−18.41
{{0,0x3}}	{{13,0xb}}	−18.42
{{0,0x3}}	{{13,0xe}}	−18.42
{{0,0x6}}	{{13,0xb}}	−18.63
{{0,0x6}}	{{13,0xe}}	−18.63
{{0,0x6}}	{{13,0x3}}	−18.63

Summing these 13 contributions gives

$$|c_m| = \sum_{i=1}^{13} |c_{m,i}| = 4.661062 \times 10^{-5} \approx 2^{-14.40}.$$

Each path has correlation magnitude 2^{-6} and ends at the common mask λ_{out}. Therefore, the correlation of the complete nine-round distinguisher is $|C_{\text{DL}}| \approx 2^{-35} \cdot 2^{-14.40} \cdot (2^{-6})^2 = 2^{-61.40}$.

A.2 Weak-key of DL Key Recovery Attack on Orthros

For Orthros, let X_1 denote an arbitrary state after the first-round S-box layer, and let Δ_{out}^1 and Δ_{out}^2 be the two output differences. The following condition must hold in order for the attack to be valid:

$$S(S^{-1}(X_1) \oplus K) \oplus S(S^{-1}(X_1 \oplus \Delta_{\text{out}}^1) \oplus K) = \Delta_{\text{out}}^2.$$

Here, $k = RK_0^1 \oplus RK_0^2$ denotes the 4-bit subkey value associated with the active S-box in the key-recovery procedure. By exhaustively enumerating k and the corresponding input-output states, all possible state transitions involved in the 9-round differential-linear key-recovery attack on Orthros are obtained, as shown in Table 7.

Table 7. Possoble state transitions of Orthros through $S \circ S^{-1}$

	$\texttt{0x4} \to \texttt{0x8}$	$\texttt{0x4} \to \texttt{0x1}$
$k = \texttt{0x0}$	–	–
$k = \texttt{0x1}$	–	–
$k = \texttt{0x2}$	$(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0xd}, \texttt{0x5})$	–
$k = \texttt{0x3}$	$(\texttt{0x2}, \texttt{0x6}) \to (\texttt{0x0}, \texttt{0x8})$	$(\texttt{0x3}, \texttt{0x7}) \to (\texttt{0xd}, \texttt{0xc})$ $(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x6}, \texttt{0x7})$
$k = \texttt{0x4}$	–	–
$k = \texttt{0x5}$	$(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x1}, \texttt{0x9})$ $(\texttt{0xa}, \texttt{0xe}) \to (\texttt{0xf}, \texttt{0x7})$	–
$k = \texttt{0x6}$	–	–
$k = \texttt{0x7}$	$(\texttt{0x2}, \texttt{0x6}) \to (\texttt{0x8}, \texttt{0x0})$ $(\texttt{0xa}, \texttt{0xe}) \to (\texttt{0x7}, \texttt{0xf})$	–
$k = \texttt{0x8}$	–	$(\texttt{0xb}, \texttt{0xf}) \to (\texttt{0x2}, \texttt{0x3})$
$k = \texttt{0x9}$	–	$(\texttt{0x3}, \texttt{0x7}) \to (\texttt{0xc}, \texttt{0xd})$
$k = \texttt{0xa}$	$(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x5}, \texttt{0xd})$	–
$k = \texttt{0xb}$	–	$(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x7}, \texttt{0x6})$
$k = \texttt{0xc}$	–	–
$k = \texttt{0xd}$	$(\texttt{0x0}, \texttt{0x4}) \to (\texttt{0xf}, \texttt{0x7})$ $(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x9}, \texttt{0x1})$	–
$k = \texttt{0xe}$	–	$(\texttt{0xb}, \texttt{0xf}) \to (\texttt{0x3}, \texttt{0x2})$
$k = \texttt{0xf}$	$(\texttt{0x0}, \texttt{0x4}) \to (\texttt{0x7}, \texttt{0xf})$	–

For example, when $k = \texttt{0x0}$, there is no pair satisfying either the differential transition $\texttt{0x4} \to \texttt{0x8}$ or $\texttt{0x4} \to \texttt{0x1}$. When $k = \texttt{0x2}$, the only state pairs satisfying the transition $\texttt{0x4} \to \texttt{0x8}$ are

$$(\texttt{0x8}, \texttt{0xc}) \to (\texttt{0x5}, \texttt{0xd}), \qquad (\texttt{0xc}, \texttt{0x8}) \to (\texttt{0xd}, \texttt{0x5}).$$

In our 9-round key-recovery attack, two active S-boxes are involved, corresponding to 8 bits of k in total. For the full key space, these 8 bits can take all possible values uniformly. The attack is effective only when the subkey values used by both active S-boxes satisfy the required differential transitions. Therefore, the fraction of weak keys is $\frac{7}{16} \cdot \frac{5}{16} \approx 2^{-2.87}$.

A.3 DL Key Recovery on Orthros Considering 2 Rounds Pretended

We also consider prepending two rounds to a seven-round DL distinguisher. Our search yields the nine-round candidate shown in Figure 5. The required S-box

output differences in the two prepended rounds are

$$\Delta^1_{\mathrm{out},0} = \texttt{0x0000 0000 0000 0000 0000 0000 0004 0400},$$
$$\Delta^2_{\mathrm{out},0} = \texttt{0x0000 0000 0000 0000 0000 0000 0008 0100},$$
$$\Delta^1_{\mathrm{out},1} = \texttt{0x0000 0000 0000 0082 0000 0000 0000 0000},$$
$$\Delta^2_{\mathrm{out},1} = \texttt{0x0000 0000 8800 0000 0000 0000 0000 0000}.$$

Evaluating the two-round prefix involves eight nibbles of each whitening key RK^1_0, RK^2_0 and two nibbles of each first-round key RK^1_1, RK^2_1. Under the linear key schedules, these subkey bits depend on $N_t = 54$ distinct master-key bits, with

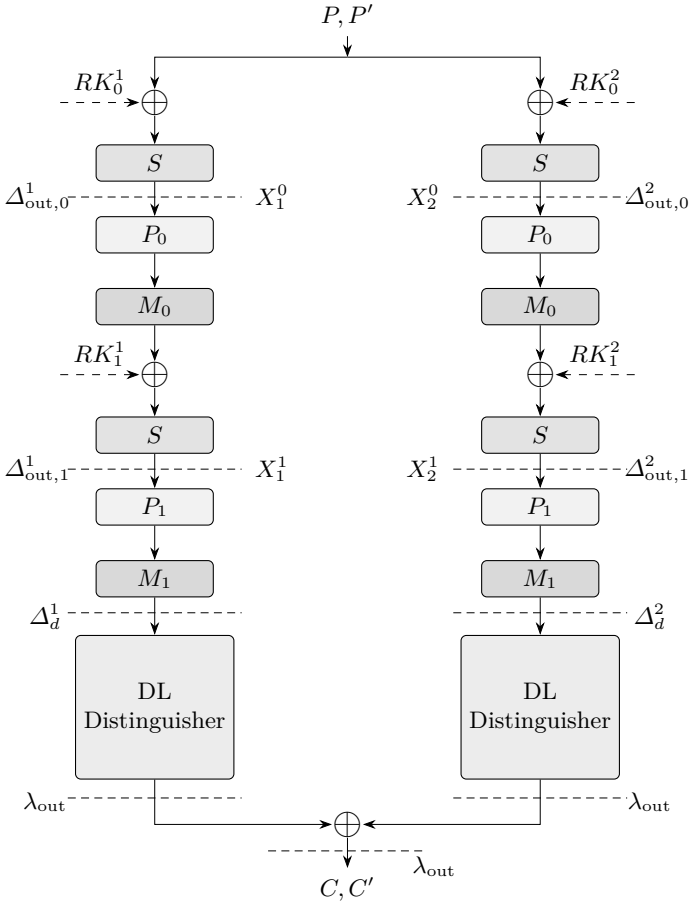

Fig. 5. Differential-linear key-recovery attack on 9-round Orthros with two prepended key-recovery rounds.

$N_c = 10$ common master-key positions between the two branch-key sets. The plaintext structures vary $N_a = 32$ bits, and the seven-round DL distinguisher has correlation magnitude $|C_{\mathrm{DL}}| \approx 2^{-41.59}$.

Let m be the number of structures. Each structure contains 2^{N_a} plaintexts and approximately 2^{2N_a-1} unordered pairs. For a fixed subkey guess, the Branch 1 condition retains a fraction 2^{-N_a}, and the Branch 2 condition retains a further fraction $2^8/2^{N_a}$. Thus, the number of usable pairs is $m \cdot 2^{2N_a-1} \cdot 2^{-N_a} \cdot \frac{2^8}{2^{N_a}} = 2^7 m$.

For a conservative lower bound, we set $a = 1$. Using Eq. (1) with $P_S = 85.23\%$ gives $N \approx 2^{83.31}$ usable pairs. Therefore, $m = N/2^7 \approx 2^{76.31}$, and the data complexity is $D = m2^{N_a} + 2 \approx 2^{108.31}$ chosen plaintexts.

Even if the work in the second prepended round is omitted, the time complexity is lower-bounded by

$$T \geq 2^{N_t} m 2^{N_a} \cdot \frac{N_a}{4 \cdot 32} \cdot \frac{1}{18} + 2^{128-a} \approx 2^{156.14}.$$

The two 2^{N_t}-entry arrays of 128-bit counters and one plaintext–output structure require $M = 2 \cdot 128 \cdot 2^{54} + 256 \cdot 2^{32} \approx 2^{62}$ bits. Since this lower bound already exceeds exhaustive search, the candidate does not yield an effective key-recovery attack. We therefore restrict the concrete Orthros attacks to one prepended round.

B DL Key-Recovery Attack on Gleeok-256

As for Gleeok-128, we first analyze the three branches separately and then impose the SoP compatibility conditions. The branch input differences must arise from one external plaintext difference, with the initialization of Branch 2 taken into account, and the output masks must coincide. Under the branch-independence approximation, the full-PRF correlation is the product of the branch correlations. Table 8 summarizes the results.

Table 8. Estimated DL correlation magnitudes for Gleeok-256. For Branch 3, p_d is the probability of the selected differential prefix.

Target	Rounds	r_d	r_m	r_l	p	$\lvert c_m \rvert$	$\lvert q \rvert$	$\lvert C_{\mathrm{DL}} \rvert$
Branch 1	6	0	6	0	–	$2^{-23.44}$	–	$2^{-23.44}$
Branch 2	6	0	6	0	–	$2^{-20.50}$	–	$2^{-20.50}$
Branch 3	9	2	7	0	2^{-28}	$2^{-84.17}$	–	$2^{-112.17}$
Full PRF	6	0	6	0	–	$2^{-68.13}$	–	$2^{-68.13}$

The six-round distinguishers for Branches 1 and 2 both use input difference $\{\{1, \texttt{0x01}\}\}$. Their output masks are $\{\{26, \texttt{0x4}\}\}$ and $\{\{35, \texttt{0x02}\}\}$, respectively; the indices refer to the 3-bit and 5-bit S-boxes. For Branch 3, a two-round

differential trail precedes a seven-round round-based evaluation. Using 4-bit S-box indices, the trail starts from

$$\Delta_{\text{in}}^3 = \{\{19, \texttt{0x2}\}, \{20, \texttt{0x2}\}, \{22, \texttt{0xf}\}, \{23, \texttt{0xf}\}, \\ \{58, \texttt{0x2}\}, \{59, \texttt{0x2}\}, \{61, \texttt{0x2}\}, \{62, \texttt{0x2}\}\}$$

and reaches $\Delta_{\text{mid}}^3 = \{\{55, \texttt{0x6}\}, \{60, \texttt{0x6}\}\}$ with probability 2^{-28}. The following seven rounds use output mask $\{\{34, \texttt{0x4}\}\}$ and have estimated correlation $2^{-84.17}$. The nine-round Branch 3 correlation is therefore $2^{-28} \cdot 2^{-84.17} = 2^{-112.17}$.

The full six-round PRF distinguisher has estimated correlation $2^{-68.13}$. Its external input difference is $\{\{6, \texttt{0x80}\}\}$, and its output mask is $\{\{26, \texttt{0x04}\}\}$; both indices refer to bytes. For key recovery, we prepend one round to the five-round distinguisher below.

Characteristic. The five-round DL correlation is $|C_{\text{DL}}| = 2^{-67.27}$. All indices are zero-based, and bits are numbered from the most significant bit. In Branches 1 and 2, S_3^i acts on the three most significant bits of byte i, while S_5^i acts on its five least significant bits. In Branch 3, S_4^{2i} acts on the high nibble of byte i, while S_4^{2i+1} acts on its low nibble.

The input differences of the five-round distinguisher are

$$\begin{aligned} \Delta_{\text{d}}^1 &: \quad \Delta S_5^0 = \texttt{0x01},\ \Delta S_3^2 = \texttt{0x1},\ \Delta S_3^3 = \texttt{0x1}, \\ \Delta_{\text{d}}^2 &: \quad \Delta S_3^{10} = \texttt{0x2},\ \Delta S_3^{11} = \texttt{0x2},\ \Delta S_5^{12} = \texttt{0x08}, \\ \Delta_{\text{d}}^3 &: \quad \Delta S_4^3 = \texttt{0x4},\ \Delta S_4^{26} = \texttt{0x4},\ \Delta S_4^{44} = \texttt{0x4}. \end{aligned}$$

Propagating these differences backward through the linear layers $L_b = \pi_b \circ \theta_b$ gives

$$\begin{aligned} \Delta_{\text{out}}^1 &: \quad \Delta S_5^{31} = \texttt{0x01}, \\ \Delta_{\text{out}}^2 &: \quad \Delta S_5^{28} = \texttt{0x01}, \\ \Delta_{\text{out}}^3 &: \quad \Delta S_4^{63} = \texttt{0x1}. \end{aligned}$$

All unlisted differences are zero, and $L_b(\Delta_{\text{out}}^b) = \Delta_{\text{d}}^b$ for $b \in \{1, 2, 3\}$. The common output mask is $\lambda_{\text{out}} = \{\{24, \texttt{0x10}\}\}$, where 24 is a byte index.

Weak-key characterization. Let RK_0^b be the whitening key of Branch b. Branches 1 and 3 process $P \oplus RK_0^b$, whereas Branch 2 processes $\pi_{\text{init}}(P) \oplus RK_0^2$, where $(\pi_{\text{init}}(P))[i] = P[41i \bmod 256]$.

We use plaintext pairs (P, P') with $\Delta_P = P \oplus P' = 0^{255} \,\|\, 1$. The initialization of Branch 2 maps bit 255 to bit 231. Hence the active S-boxes are S_5^{31}, S_5^{28}, and S_4^{63} in Branches 1, 2, and 3, respectively. Their required transitions are $\texttt{0x01} \to \texttt{0x01}$ in Branches 1 and 2 and $\texttt{0x1} \to \texttt{0x1}$ in Branch 3.

The corresponding input sets are

$$\begin{aligned} \chi_5 &= \{x : S_5(x) \oplus S_5(x \oplus \texttt{0x01}) = \texttt{0x01}\} \\ &= \{\texttt{0x02}, \texttt{0x03}, \texttt{0x06}, \texttt{0x07}, \texttt{0x0a}, \texttt{0x0b}, \texttt{0x0e}, \texttt{0x0f}\}, \\ \chi_4 &= \{x : S_4(x) \oplus S_4(x \oplus \texttt{0x1}) = \texttt{0x1}\} = \{\texttt{0x0}, \texttt{0x1}\}. \end{aligned}$$

Thus, χ_5 and χ_4 have the binary forms `0**1*` and `000*`, respectively.

Algorithm 2: Key recovery on 6-round Gleeok-256

Input: Encryption oracle E_K; number of structures $m \geq 1$; target advantage a; plaintext difference Δ_P; target output differences Δ^b_{out} of the first S-box layers; output mask λ_{out}

Output: The verified master key, or failure

Initialize $T[k] = V[k] = 0$ for all $k \in \{0,1\}^{N_t}$;
Query three distinct verification plaintexts and store the resulting plaintext–output pairs in $\mathcal{D}_{\text{ver}}$;
for $j = 1$ **to** m **do**
  Generate $\mathcal{P}_j$ by varying A and fixing the other bits uniformly;
  Store $L_j[P] \leftarrow E_K(P)$ for every $P \in \mathcal{P}_j$;
  foreach unordered pair $\{P, P'\} \subseteq \mathcal{P}_j$ satisfying $P \oplus P' = \Delta_P$ **do**
    foreach $k \in \{0,1\}^{N_t}$ **do**
      Derive the active bits of RK^b_0 from k according to the key schedule;
      for $b = 1$ **to** 3 **do**
        $Q_b \leftarrow (\mathcal{S}^b)^{-1}\big(\mathcal{S}^b(I^b(P) \oplus RK^b_0) \oplus \Delta^b_{\text{out}}\big) \oplus RK^b_0$;
        $P'_b \leftarrow (I^b)^{-1}(Q_b)$;
      if $P'_1 = P'_2 = P'_3 = P'$ **then**
        $T[k] \leftarrow T[k] + 1$;
        if $\lambda_{\text{out}} \cdot (L_j[P] \oplus L_j[P']) = 0$ **then** $V[k] \leftarrow V[k] + 1$;
$\mathcal{R} \leftarrow \left\{k : T[k] > 0,\ 2\sqrt{T[k]}\left(\frac{V[k]}{T[k]} - \frac{1}{2}\right) > \Phi^{-1}(1 - 2^{-a})\right\}$;
foreach $k \in \mathcal{R}$ and $z \in \{0,1\}^{256-N_t}$ **do**
  Form $\widehat{K}$ by assigning k to the involved master-key positions and z to the remaining positions;
  if $E_{\widehat{K}}(P) = C$ for every $(P, C) \in \mathcal{D}_{\text{ver}}$ **then return** $\widehat{K}$;
return failure;

Let $RK^b_0[i:j]$ denote bits i through j of RK^b_0, and let χ_5^{low} contain the four least significant bits of the elements of χ_5. Compatibility between Branches 1 and 3 requires

$$RK^1_0[252:255] \oplus RK^3_0[252:255] \in \chi_5^{\text{low}} \oplus \chi_4 = \chi_5^{\text{low}},$$

where set XOR contains all pairwise XORs. Thus, for each value of $RK^1_0[252:255]$, eight of the 16 values of $RK^3_0[252:255]$ are compatible.

The active S_5 of Branch 2 receives $P[91], P[132], P[173], P[214], P[255]$. The pattern `0**1*` fixes only $P[91]$ and $P[214]$, which are unconstrained by the other branches; hence Branch 2 adds no key restriction. Because the relevant Branch 1 and Branch 3 slices contain distinct master-key bits, the weak-key fraction is $1 \cdot 1 \cdot \frac{8}{2^4} = 2^{-1}$.

Key-recovery procedure. The prefix involves $N_t = 13$ master-key bits. However, changing $K[17], K[28], K[85], K[164], K[183]$, or $K[199]$ does not change

the selected pairs or their scores. Each score therefore represents a class of 2^6 indistinguishable guesses, which the threshold test retains or rejects together.

Let $A = \{91, 132, 173, 214, 251, 252, 253, 254, 255\}$ be the active plaintext-bit set. Each structure varies the nine bits in A. Let $I^1 = I^3 = \mathrm{id}$, $I^2 = \pi_{\mathrm{init}}$, and let $\mathcal{S}^b$ denote the S-box layer of Branch b. In the reconstruction, only the active S-boxes are evaluated; all other input bits are copied.

Complexity analysis. Each structure contains 2^9 plaintexts and supplies exactly four pairs for the correct weak-key class, so $N = 4m$. Since each score represents 2^6 indistinguishable guesses, we set $a = 13-6 = 7$. With $|C_{\mathrm{DL}}| = 2^{-67.27}$, Eq. (1) gives $N \approx 2^{138.12}$. This gives $P_S \approx 85.07\%$. Hence, $m = N/4 \approx 2^{136.12}$, and including three verification plaintexts, the data complexity is $D = 2^9 m + 3 \approx 2^{145.12}$ chosen plaintexts.

For each structure, the local procedure tests 2^8 unordered pairs under all 2^{13} local key guesses. This requires $m\,2^8\,2^{13} \approx 2^{157.12}$ local tests, each involving three active S-box reconstructions. Data collection requires approximately $2^{145.12}$ six-round Gleeok-256 evaluations.

For each retained equivalence class, the 2^6 indistinguishable local guesses are combined with the $2^{256-N_t} = 2^{243}$ assignments to the remaining master-key bits. Under the threshold model of Section 2, the final search requires $2^6 \cdot 2^{243} = 2^{249}$ full-key trials and dominates the preceding stages. Thus, the time complexity is $T \approx 2^{249}$ six-round Gleeok-256 evaluations.

The memory complexity is approximately 2^{22} bits, dominated by two arrays of 2^{13} 256-bit counters. Indexing the current outputs by the nine active plaintext bits requires another $2^9 \cdot 256 = 2^{17}$ bits. The success probability and complexities are conditioned on the weak-key class, whose coverage is 2^{-1}.

C DL Key Recovery on $(8, 6)$-round ZIP-GIFT-64 with One Linear Branch

We first find a $(7, 7)$-round differential-linear distinguisher of ZIP-GIFT-64 with $r_{\mathrm{m}} = 7$ and $|C_{\mathrm{DL}}| = 2^{-27.72}$. Its input difference is `0x0000000080000000` and its output mask is `0x0000000000000001`. Following the differential-and-linear key-recovery framework of Flórez-Gutiérrez et al. [14], we use a five-round DL distinguisher in the forward branch and a six-round linear trail in the backward branch. The forward distinguisher has correlation $|C_{\mathrm{DL}}| \approx 2^{-3.45}$, and the backward trail has correlation $|C_{\mathrm{L}}| = 2^{-11}$. The two components use the same output mask. Under the usual independence assumption, the correlation of the resulting $(5, 6)$-round hybrid distinguisher is $|C| \approx |C_{\mathrm{DL}}| \cdot |C_{\mathrm{L}}|^2 = 2^{-25.45}$. The forward input difference, backward input mask, and common output mask are $\Delta_{\mathrm{in}} =$ `0x00000c0000000000`, $\gamma =$ `0x0022040100000140`, and $\lambda =$ `0x0000000000000011`, respectively. The backward trail is given in Table 9.

We prepend the three-round truncated differential in Table 10 to the forward branch, obtaining an attack on $(8, 6)$-round ZIP-GIFT-64. The attack proceeds as follows.

Table 9. Six-round linear trail in the backward branch of ZIP-GIFT-$64_{5,6}$.

Symbol	Mask
γ	0022040100000140
ΓX_0^P	0000000506000300
ΓX_1	0000000804000200
ΓX_1^P	0000006000000080
ΓX_2	0000004000000010
ΓX_2^P	0000000000050000
ΓX_3	0000000000080000
ΓX_3^P	0000000000000800
ΓX_4	0000000000000100
ΓX_4^P	0000000100000000
ΓX_5	0000000c00000000
ΓX_5^P	0000000000000084
λ	0000000000000011

1. Select $2N$ plaintexts, organized into $2N/2^{32}$ structures. To construct one structure, choose an arbitrary intermediate state X_1^P and enumerate all 2^{32} values of the 32 bits that enter the eight active S-boxes in the second prepended round, while keeping the other bits fixed. Apply $\mathrm{S}^{-1} \circ \mathrm{P}^{-1}$ to each resulting state X_1^P to obtain the corresponding plaintext. For the truncated pattern in Table 10, the partner of every state belongs to the same structure; hence, each structure provides 2^{31} disjoint pairs and the complete data set provides N pairs.
2. Query the encryption oracle to obtain the outputs of all selected plaintexts after $(8, 6)$ rounds of ZIP-GIFT-64.
3. For each value of the 20-bit equivalent subkey $K_{\mathrm{eq}} = RK_1[\mathcal{I}_1] \parallel RK_2[\mathcal{I}_2]$, where the positions in $\mathcal{I}_1$ and $\mathcal{I}_2$ are marked in Table 10 and satisfy $|\mathcal{I}_1| = 16$ and $|\mathcal{I}_2| = 4$, partially encrypt the plaintexts through the 3-round prefix. The 16 bits from RK_1 determine the inputs of the eight active S-boxes in the second prepended round, while the four bits from RK_2 determine the inputs of the two active S-boxes in the third prepended round. The addition of RK_3 does not affect the difference entering the 5-round differential-linear distinguisher and therefore need not be guessed.
4. For each guess K_{eq}, identify within each structure the plaintext pairs that reach Δ_{in} at the input of the 5-round differential-linear distinguisher. By the structure construction, there are N such pairs for each guess.
5. For every selected plaintext pair (P, P') and the corresponding output pair (C, C'), compute $t = \langle C \oplus C', \lambda \rangle \oplus \langle P \oplus P', \gamma \rangle$. The second term accounts for the input mask of the backward linear trail. If $t = 0$, increment $V_{K_{\mathrm{eq}}}$.
6. For each K_{eq}, if $V_{K_{\mathrm{eq}}}/N - \frac{1}{2} > \theta$, retain K_{eq}, where θ is chosen according to N and the target advantage a.
7. For every retained K_{eq}, enumerate the 2^{108} consistent master keys and verify them with two additional plaintext–output pairs.

Table 10. Three-round truncated differential in the forward branch of ZIP-GIFT-64.

Symbol	Pattern
ΔP	`**** **** **** **** **** **** **** ****` `**** **** **** **** **** **** **** ****`
ΔX_1^S	`*00* **00 0**0 00** *00* **00 0**0 00**` `*00* **00 0**0 00** 100* *100 0**0 00**`
RK_1	`00 00 00 00 00 00 00 00 11 11 11 11 11 11 11 11`
ΔX_1^P	`0000 0000 0000 0000 0000 0000 0000 0000` `**** **** **** **** **** **** **** 11**`
ΔX_2^S	`0000 0000 0000 0000 0000 0000 0000 0000` `00*0 000* *000 0*00 00*0 000* 1000 0100`
RK_2	`00 00 00 00 00 00 11 11 00 00 00 00 00 00 00 00`
ΔX_2^P	`0000 0000 0000 0000 0000 0000 **** 11**` `0000 0000 0000 0000 0000 0000 0000 0000`
ΔX_3^S	`0000 0000 0000 0000 0000 0000 1000 0100` `0000 0000 0000 0000 0000 0000 0000 0000`
ΔX_3^P (Δ_{in})	`0000 0000 0000 0000 0000 1100 0000 0000` `0000 0000 0000 0000 0000 0000 0000 0000`

Complexity analysis. We set $a = 20$. With $|C| = 2^{-25.45}$, Eq. (1) gives $N \approx 2^{55.98}$ pairs and $P_S \approx 85.23\%$. Including two verification plaintexts, the data complexity is $D = 2N + 2 \approx 2^{56.98}$ chosen plaintexts.

For each of the 2^{20} equivalent-subkey guesses, partial encryption evaluates eight active S-boxes in the second prepended round and two in the third. Counting one full $(8, 6)$-round ZIP-GIFT-64 evaluation as $(8 + 6) \times 16 = 224$ S-box evaluations, this costs $T_{\text{partial}} = 2^{20} \cdot D \cdot \frac{10}{224} \approx 2^{72.50}$ equivalent $(8, 6)$-round evaluations. Under the wrong-key model, the threshold retains 2^{20-a} candidates on average. Since each candidate is consistent with 2^{108} master keys, the final search costs $2^{20-a} \cdot 2^{108} = 2^{128-a} = 2^{108}$ evaluations. Thus, $T = D + T_{\text{partial}} + 2^{128-a} \approx 2^{108}$. Processing one structure at a time requires 2^{32} 64-bit outputs and 2^{20} 64-bit counters. The memory complexity is therefore $M = 2^{38} + 2^{26} \approx 2^{38}$ bits.

D DL Key-Recovery Attack on (10, 8)-Round ZIP-GIFT-128 with a Linear Backward Branch

We apply the same differential-and-linear framework to ZIP-GIFT-128. The forward branch uses a seven-round DL distinguisher with input difference $\Delta_{\text{in}} =$ `0x00000000 0000000c 00000000 00000000` and correlation $|C_{\text{DL}}| = 2^{-17.09}$. The backward branch uses the eight-round linear trail in Table 11, whose correlation is $|C_{\text{L}}| = 2^{-18}$. Its input and output masks are

$$\gamma = \texttt{0x0088 0000 0000 4400 0000 0000 0000 1100},$$
$$\lambda = \texttt{0x0000 0000 0000 0000 000c 000c 0000 0000}.$$

Under the usual independence assumption, the correlation of the resulting $(7, 8)$-round hybrid distinguisher is $|C| \approx |C_{\text{DL}}| \cdot |C_{\text{L}}|^2 = 2^{-53.09}$.

Table 11. Eight-round linear trail in the backward branch of ZIP-GIFT-$128_{7,8}$.

Symbol	Mask
γ	00880000000044000000000000001100
ΓX_0^P	00000000000800080005000500000000
ΓX_1	00000000000100010008000800000000
ΓX_1^P	00000000000009000000000000000900
ΓX_2	00000000000005000000000000000500
ΓX_2^P	00000000000000000000050500000000
ΓX_3	00000000000000000000080800000000
ΓX_3^P	00000000000000000000080000000800
ΓX_4	00000000000000000000020000000100
ΓX_4^P	00000000000000000000000300000000
ΓX_5	00000000000000000000000200000000
ΓX_5^P	00000000000000000000000000000002
ΓX_6	0000000000000000000000000000000c
ΓX_6^P	00000000000000000000000000008400
ΓX_7	00000000000000000000000000001100
ΓX_7^P	00000000000000000001000100000000
λ	0000000000000000000c000c00000000

We prepend the three-round truncated differential in Table 12 to the forward branch, obtaining an attack on $(10, 8)$-round ZIP-GIFT-128. The procedure is as follows.

1. Select $2N$ plaintexts, organized into $2N/2^{32}$ structures. To construct one structure, choose an arbitrary intermediate state X_1^P, and enumerate all 2^{32} values of the 32 state bits entering the eight active S-boxes in the second prepended round, while keeping all other state bits fixed. Apply $\mathrm{S}^{-1} \circ \mathrm{P}^{-1}$ to each resulting intermediate state X_1^P to obtain the corresponding plaintext. For the truncated pattern in Table 12, the partner of every state belongs to the same structure. Hence, each structure provides 2^{31} disjoint pairs, and the complete data set provides N pairs.
2. Query the encryption oracle to obtain the outputs of all selected plaintexts after $(10, 8)$ rounds of ZIP-GIFT-128.
3. For each value of the 20-bit equivalent subkey $K_{\mathrm{eq}} = RK_1[\mathcal{I}_1] \parallel RK_2[\mathcal{I}_2]$, where the positions in $\mathcal{I}_1$ and $\mathcal{I}_2$ are marked in Table 12 and satisfy $|\mathcal{I}_1| = 16$ and $|\mathcal{I}_2| = 4$, partially encrypt the plaintexts through the 3-round prefix in the forward branch. The 16 bits from RK_1 determine the inputs of the eight active S-boxes in the second prepended round, while the four bits from RK_2 determine the inputs of the two active S-boxes in the third prepended round. The addition of RK_3 does not affect the difference entering the 7-round differential-linear distinguisher and therefore need not be guessed.
4. For each subkey guess K_{eq}, identify within each structure the plaintext pairs that reach Δ_{in} at the input of the 7-round differential-linear distinguisher. By the structure construction, there are N such pairs for every subkey guess.

Table 12. Three-round truncated differential in the forward branch of ZIP-GIFT-128.

Symbol	Pattern
ΔP	**** **** **** **** **** **** **** **** **** **** **** **** **** **** **** ****
	**** **** **** **** **** 11** **** **** **** **** **** **** **** **** **** ****
ΔX_1^S	*000 0*00 00*0 000* *000 0*00 00*0 000* *000 0*00 00*0 000* *000 0*00 00*0 000*
	*000 0*00 00*0 000* 1000 0100 00*0 000* *000 0*00 00*0 000* *000 0*00 00*0 000*
RK_1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00 00 00 00 00 00 00 00 11 11 11 11 11 11 11 11
ΔX_1^P	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000 **** **** **** **** **** 11** **** ****
ΔX_2^S	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000 *000 0*00 00*0 000* 1000 0100 00*0 000*
RK_2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 11 11
ΔX_2^P	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 **** 11**
ΔX_3^S	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1000 0100
ΔX_3^P (Δ_{in})	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1100
	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

5. For every selected plaintext pair (P, P') and its corresponding output pair (C, C'), compute $t = \langle C \oplus C', \lambda \rangle \oplus \langle P \oplus P', \gamma \rangle$. The second term accounts for the input mask of the backward linear trail. If $t = 0$, increment $V_{K_{\text{eq}}}$.
6. For each K_{eq}, if $V_{K_{\text{eq}}}/N - \frac{1}{2} > \theta$, retain K_{eq}, where θ is chosen according to N and the target advantage a.
7. For every retained K_{eq}, enumerate the 2^{108} consistent master keys and verify them with two additional plaintext–output pairs.

Complexity analysis. We set $a = 5$ to balance statistical processing and exhaustive search. With $|C| = 2^{-53.09}$, Eq. (1) gives $N \approx 2^{109.26}$ and $P_S \approx 85.23\%$. Including two verification plaintexts, the data complexity is $D = 2N+2 \approx 2^{110.26}$ chosen plaintexts.

For each of the 2^{20} equivalent-subkey guesses, partial encryption evaluates eight active S-boxes in the second prepended round and two in the third. Counting one full $(10, 8)$-round ZIP-GIFT-128 evaluation as $(10+8) \times 32 = 576$ S-box evaluations, this costs $T_{\text{partial}} = 2^{20} \cdot D \cdot \frac{10}{576} \approx 2^{124.41}$ equivalent $(10, 8)$-round evaluations. The threshold retains $2^{20-a} = 2^{15}$ candidates on average. Since each candidate is consistent with 2^{108} master keys, the final search requires $T_{\text{ex}} = 2^{20-a} \cdot 2^{108} = 2^{128-a} = 2^{123}$ evaluations. Therefore, the total time complexity is $T = D + T_{\text{partial}} + T_{\text{ex}} \approx 2^{124.87}$.

Processing one structure at a time requires 2^{32} 128-bit outputs and 2^{20} 128-bit counters. Hence, the memory complexity is $M = 2^{39} + 2^{27} \approx 2^{39}$ bits.