Cryptanalysis of Gleeok-128

Siwei Chen[1,2], Peipei Xie[1,2], Shengyuan Xu[2,3*], Xiutao Feng[4], Zejun Xiang[1,2], Xiangyong Zeng[2,5]

[1]School of Cyber Science and Technology, Hubei University, Wuhan, Hubei, China.
[2]Key Laboratory of Intelligent Sensing System and Security, Ministry of Education, Hubei University, Wuhan, Hubei, China.
[3]Department of Fundamental Courses, Shandong University of Science and Technology, Taian, Shandong, China.
[4]Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China.
[5]Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan, Hubei, China.

*Corresponding author(s). E-mail(s): xushengyuan@sdust.edu.cn;
Contributing authors: chensiwei_hubu@163.com;
xiepeipei@stu.hubu.edu.cn; fengxt@amss.ac.cn;
xiangzejun@hubu.edu.cn; xzeng@hubu.edu.cn;

Abstract

Gleeok is a family of low-latency keyed pseudorandom functions (PRF) including two variants called Gleeok-128 and Gleeok-256, which are based on three parallel SPN-based keyed permutations whose outputs are XORed to produce the final output. Both Gleeok-128 and Gleeok-256 employ a 256-bit key, with block sizes of 128 and 256 bits, respectively. Due to this multi-branch structure, evaluating its security margin and mounting valid key-recovery attacks present non-trivial challenges. In this paper, we present the first comprehensive third-party cryptanalysis of Gleeok-128, including a two-stage MILP-based framework for constructing branch-wise and full-cipher differential-linear (DL) distinguishers, and a dedicated key-recovery framework based on integral distinguishers for multi-branch designs. Our analysis yields 7-/7-/8-/4-round DL distinguishers for Branch1/Branch2/Branch3/Gleeok-128 with squared correlations $\mathbf{2^{-88.12}}/\mathbf{2^{-88.12}}/\mathbf{2^{-38.73}}/\mathbf{2^{-49.04}}$. All distinguishers except the one targeting

the PRF outperform the best distinguishers given in the design document. Moreover, by tightening algebraic degree bounds, we obtain 9-/9-/7-round integral distinguishers for the three branches and a 7-round distinguisher for the full PRF, extending the existing ones proposed in the original design document by 3/3/2 rounds and 2 rounds, respectively. Furthermore, the newly explored integral distinguishers enable the key-recovery attacks on Gleeok-128: a 7-round attack in the non-full-codebook setting and an 8-round attack in the full-codebook setting. In addition, we uncover a flaw in the original linear security evaluation of Branch3, showing that it can be distinguished over all 12 rounds with data complexity $\mathbf{2^{48}}$, and propose optimized linear-layer parameters that significantly strengthen its linear resistance without sacrificing diffusion. These results advance the understanding of Gleeok-128's security and provide generalized methods for analyzing multi-branch cipher designs.

Keywords: Gleeok-128, Pseudorandom Function, Security Evaluation, Key-recovery Attacks

1 Introduction

In recent years, the growing demand for real-time applications, such as 5G communications, Trusted Execution Environments (TEEs), telemedicine, and vehicle-to-everything (V2X) systems, has made cryptographic latency a critical performance factor. In such latency-sensitive contexts, cryptographic operations often lie on the critical path, and even small delays can degrade system responsiveness. To address this, several low-latency symmetric designs have been proposed, including PRINCE [1], QARMA [2], MANTIS [3], and Orthros [4]. Orthros, for example, is a 128-bit keyed pseudorandom function (PRF) constructed as the sum of two keyed permutations to achieve security through structural composition. However, most existing designs support only 128-bit keys, which may be insufficient for emerging applications requiring higher security levels. Against this backdrop, Gleeok was recently introduced as a family of low-latency PRFs supporting 256-bit keys while maintaining sub-nanosecond latency [5].

Gleeok is a recently proposed family of pseudorandom functions designed to achieve ultra-low latency while supporting 256-bit keys. Drawing inspiration from Orthros, its core construction is based on three parallel 128-bit branches: the first two are tailored to provide resistance against statistical attacks, particularly differential cryptanalysis, while the third branch is intended to counter algebraic attacks. The outputs of the three branches are XORed to form the final ciphertext. In addition to the standard variant, Gleeok also includes wide-block versions using three 256-bit branches. Currently, there is no third-party analysis of Gleeok, and analysis frameworks for multi-branch cipher structures remain limited.

The original design document of Gleeok evaluates its security against various classes of classical attacks, including differential and linear cryptanalysis, impossible differential attacks, integral attacks, and other related methods. Their analysis concludes

that the best known attacks on Gleeok-128 and Gleeok-256 are 5-round integral distinguishers rather than key-recovery attacks. Since decrypting the last rounds from the ciphertext would require guessing 256 bits for Gleeok-128 or 512 bits for Gleeok-256, incorporating a key-recovery phase based on these distinguishers was declared to be an open problem. Furthermore, to date, no third-party cryptanalysis has been conducted to independently validate or challenge these security claims.

Motivated by these gaps, this work aims to revisit the security claims of Gleeok. In particular, we aim to investigate whether longer and more effective distinguishers can be constructed, and whether practical key-recovery attacks are feasible despite its multi-branch structure. Answering these questions is essential for a comprehensive understanding of Gleeok's security margin and for advancing cryptanalysis methodologies for similar designs.

1.1 Our Contributions

In this paper, we conduct the first comprehensive third-party cryptanalysis of Gleeok-128, targeting both its individual branches and the overall PRF structure. Our work integrates differential-linear (DL) and integral cryptanalysis as well as a novel key-recovery framework tailored for multi-branch designs. The main contributions are summarized as follows:

- **Two-stage MILP-based framework for branch-wise DL distinguishers.** We present a generic two-stage MILP-based framework for the automated search and precise evaluation of high-quality DL distinguishers. Using this framework, we identify a diverse set of efficient distinguishers for Gleeok-128: for Branch1 and Branch2, we construct 7-round DL distinguishers with squared correlation $2^{-88.12}$, improving the data complexity over the previously best-known 7-round boomerang distinguishers; for Branch3, we obtain an 8-round DL distinguisher with squared correlation $2^{-38.73}$; for the full Gleeok-128, we derive a 4-round distinguisher with squared correlation $2^{-49.04}$. These DL distinguishers are compared with the existing ones in Table 1. In addition, lower-round results are experimentally verified, with measured correlations closely matching the MILP-based estimates, confirming the accuracy of our framework.
- **Algebraic-degree-based integral distinguishers and key-recovery framework.** We apply an algebraic-degree-bound approach for constructing longer and more efficient integral distinguishers than those obtained via division property. This method yields 9-, 9-, and 7-round integral distinguishers for Branch1, Branch2, and Branch3, respectively, and a 7-round distinguisher for Gleeok-128. Building on these results, we design a key-recovery framework tailored for multi-branch ciphers that avoids guessing intermediate states of all branches by prepending only one round before the distinguisher. Using this framework, we mount a 7-round key-recovery attack in the non-full-codebook setting with 2^{124} chosen plaintexts and time complexity $2^{133.6}$ (further reduced to 2^{132} with precomputation), and an 8-round attack in the full-codebook setting that recovers all 256 key bits with time complexity 2^{129} and memory 2^{133} bytes. These represent the first key-recovery attacks on Gleeok-128 to date. The comparison of our newly explored distinguishers with the previous

ones are summarized in Table 1, and details of our key-recovery attacks are shown in Table 2.

- **Revised linear security evaluation and optimized linear-layer parameters.** We identify a modeling error in the original security evaluation of Branch3, where the linear mask propagation through the θ operation was incorrectly specified. Correcting this reveals that Branch3 can be distinguished over all 12 rounds with expected data complexity 2^{48}, indicating significantly weaker linear resistance than originally claimed. To mitigate this weakness, we perform an exhaustive search over θ/π parameter sets that preserve full diffusion, identifying six improved parameter classes with markedly lower maximal squared correlations in the early rounds. Among these, the Improved Class B (see Table C11) achieves the stronger resistance to linear cryptanalysis and differential cryptanalysis without compromising the original design rationale.

Table 1: The distinguishing attacks on Gleeok-128 and its underlying branches. The full version has 12 rounds.

Cipher	Round	Type	Data	Ref.
Branch1/2	6	Integral	2^{127}	[5]
	6	Integral	$\mathbf{2^{65}}$	Sect. 4.1
	7	Boomerang	2^{100}	[5]
	7	Differential-linear	$\mathbf{2^{88.12}}$	Sect. 3.2
	9	Integral	$\mathbf{2^{126}}$	Sect. 4.1
Branch3	5	Integral	2^{127}	[5]
	5	Integral	$\mathbf{2^{113}}$	Sect. 4.1
	8	Differential	2^{64}	[5]
	8	Linear	2^{64}	[5]
	8	Differential-linear	$\mathbf{2^{38.73}}$	Sect. 3.2
	8	Linear	$\mathbf{2^{32}}$	Sect. 5
	12	Linear	$\mathbf{2^{48}}$	Sect. 5
Gleeok-128	4	Differential	$\geq 2^{126}$	[5]
	4	Linear	$\geq 2^{124}$	[5]
	4	Differential-linear	$\mathbf{2^{49.04}}$	Sect. 3.2
	5	Integral	2^{127}	[5]
	5	Integral	$\mathbf{2^{113}}$	Sect. 4.1
	7	Integral	$\mathbf{2^{127}}$	Sect. 4.1

1.2 Organization of This Paper

This paper is organized as follows: Section 2 introduces the necessary notations, the specification of Gleeok-128, and a brief overview of the cryptanalytic techniques used in this work. In Section 3, we proposes a framework to explore DL distinguishers and

Table 2: The key-recovery attacks on Gleeok-128. The full version has 12 rounds.

Round	Time	Memory	Data	Ref.
7	$2^{133.6}$	Negligible	2^{124} CP	Sect. 4.3
7	2^{132}	2^{129} Bytes	2^{124} CP	Sect. 4.3
8	2^{137}	Negligible	2^{128} CP	Sect. 4.3
8	2^{129}	2^{133} Bytes	2^{128} CP	Sect. 4.3

apply it to construct distinguishers for each branch and the PRF. Section 4 constructs new integral distinguishers based on algebraic degree analysis and designs a general framework that achieves the longest known key-recovery attacks on Gleeok-128 to date. In addition, due to the flaws in the original linear security evaluation, Section 5 explores the optimization of linear-layer parameters for Branch3. Finally, Section 6 concludes the paper.

2 Preliminaries

In this section, we start by providing the specification of Gleeok-128. Then we review the DL cryptanalysis and summarize the key concepts of integral cryptanalysis relevant to our work.

2.1 Specification of Gleeok

Gleeok is a PRF family with two variants: Gleeok-128 and Gleeok-256. Both variants use a 256-bit key and operate on 128-bit and 256-bit input blocks, respectively. Internally, the input message M is first split and copied into three parallel branches, denoted as Branch1, Branch2, and Branch3, which are processed by independent keyed permutations of 128 or 256 bits each. The final ciphertext is computed by XORing the outputs of all three branches, as shown in the following equation:

$$C = \mathsf{Branch1}(X_1) \oplus \mathsf{Branch2}(X_2) \oplus \mathsf{Branch3}(X_3),$$

where X_1, X_2, X_3 are the initial internal states copied from the plaintext M. An overview of the overall structure is illustrated in Figure 1 (a).

Here we only introduce the detail of Gleeok-128 and omit Gleeok-256. Each branch of Gleeok-128 is designed as an SPN-based structure, with 12 rounds consisting of the following operations: a nonlinear layer (Sboxes), a 3-input XOR operation θ, a bitwise permutation π, a round key addition RK_{xor}, and a round constant addition RC_{xor}. The round function R of each branch can be represented as:

$$R = RC_{xor} \circ RK_{xor} \circ \pi \circ \theta \circ S.$$

Figure 1 (b) shows the overview of the round function.

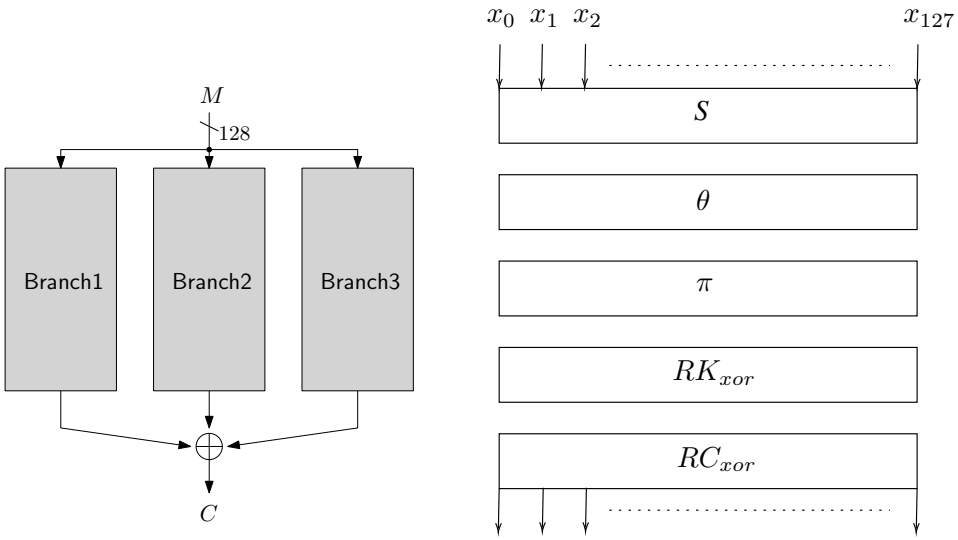

Fig. 1: (a) Overall structure and (b) round function of Gleeok-128.

Sbox Layer (S). Different Sboxes are used in different branches. In Branch1 and Branch2, the 3-bit Sbox S_3 and the 5-bit Sbox S_5 are applied alternately to the internal state as follows:

$$\begin{aligned} X \leftarrow & (S_3(x_0||x_1||x_2)||S_5(x_3||x_4||x_5||x_6||x_7)|| \\ & \cdots||S_3(x_{120}||x_{121}||x_{122})||S_5(x_{123}||x_{124}||x_{125}||x_{126}||x_{127})). \end{aligned}$$

In Branch3, a 4-bit Sbox S_4 is used uniformly, and the state is updated as:

$$\begin{aligned} X \leftarrow & (S_4(x_0||x_1||x_2||x_3)||S_4(x_4||x_5||x_6||x_7)|| \\ & \cdots||S_4(x_{120}||x_{121}||x_{122}||x_{123})||S_4(x_{124}||x_{125}||x_{126}||x_{127})). \end{aligned}$$

The truth tables of the three Sboxes can be seen in Table A1 of Supplementary Material.

3-input XOR (θ). The θ operation computes each bit as a XOR of three input bits at different positions:

$$x_i \leftarrow x_{i+t_0} \oplus x_{i+t_1} \oplus x_{i+t_2},$$

where the parameters of t_0, t_1, and t_2 vary across branches, summarized in Table 3.

Permutation (π). The permutation π is a branch-dependent bit-wise permutation, defined as:

$$x_i \leftarrow x_{i \cdot p \bmod 128}.$$

Here, p denotes a branch-dependent constant, shown in Table 3.

Key Addition (RK_{xor}). In each round, the internal state is XORed with the round key corresponding to the current branch and round number.

Table 3: The parameters of θ and π for Gleeok-128

Param.	t_0	t_1	t_2	p
Branch1	12	31	86	117
Branch2	4	23	78	117
Branch3	7	15	23	11

Constant Addition (RC_{xor}). At round r, the internal states of Branch1, Branch2, and Branch3 are XORed with the corresponding round constant RC_r^i, where $i \in \{1, 2, 3\}$. The round constants are defined as

$$RC_r^i = \mathrm{MSB}_{128}\Big((\pi - 3) \ll (r \times 128) + (i \times 12 \times 128)\Big),$$

where MSB_{128} returns the most significant 128 bits.

Key Scheduling Function. The 256-bit master key is first divided into two 128-bit parts $K_0 = (k_0 \parallel \cdots \parallel k_{127})$ and $K_1 = (k_{128} \parallel \cdots \parallel k_{255})$. For Branch1, K_0 and K_1 are used directly; for Branch2, the two halves are swapped; and for Branch3, the halves are rotated so that $K_0 = (k_{64} \parallel \cdots \parallel k_{191})$ and $K_1 = (k_{192} \parallel \cdots \parallel k_{255} \parallel k_0 \parallel \cdots \parallel k_{63})$. In the scheduling process KSF_i for Branch$i$, as outlined in Algorithm 1, K_0 and K_1 are alternately applied to generate round keys. Note that the RK_0^i is the whitening key of Branch$i$. The permutation parameter pk_i is fixed to 29, 51, and 107 for Branch1, Branch2, and Branch3, respectively.

Algorithm 1 Procedure of KSF_i

Input: Initial K_0 and K_1
Output: Round keys $(RK_0^i, RK_1^i, \ldots, RK_R^i)$
for $r \leftarrow 0$ **to** R **do**
    $(k_0 \parallel k_1 \parallel \cdots \parallel k_{127}) \leftarrow K_{r \bmod 2}$
    for $j \leftarrow 0$ **to** 127 **do**
        $k_j^* \leftarrow k_{pk_i \cdot j \bmod 128}$
    end for
    $K_{r \bmod 2} \leftarrow (k_0^* \parallel k_1^* \parallel \cdots \parallel k_{127}^*)$
    $RK_r^i \leftarrow K_{r \bmod 2}$
end for
return $(RK_0^i, RK_1^i, \ldots, RK_R^i)$

2.2 Differential-Linear Cryptanalysis

DL cryptanalysis, introduced by Langford and Hellman [6], combines differential and linear techniques to form a two-stage distinguisher. In the first stage, an input difference Δ_I is propagated with probability p through the early rounds of the cipher. The second stage applies a linear approximation with correlation q from an intermediate point to the output. When the two stages align properly in the middle of the cipher, the resulting distinguisher can be more powerful than either technique alone. Under standard assumptions, the distinguishing advantage is approximately pq^2, requiring $O(p^{-2}q^{-4})$ chosen plaintexts.

To improve modeling accuracy and capture dependencies between subcomponents, Bar-On et al. [7] proposed the Differential-Linear Connectivity Table (DLCT) at EUROCRYPT 2019, which is defined as

$$\mathrm{DLCT}_F(\Delta, \lambda) = \#\{X \in \mathbb{F}_2^n | \lambda \cdot F(X) = \lambda \cdot F(X \oplus \Delta)\} - 2^{n-1}.$$

Formally, the cipher is decomposed as $E = E_l \circ E_m \circ E_d$, as shown in Figure 2, instead

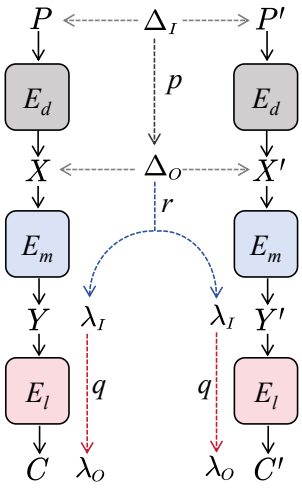

Fig. 2: The structure of DL distinguishers

of the original two-stage structure. Here, E_d is covered by a differential $\Delta_I \to \Delta_O$ with probability p and E_l is covered by a linear approximation $\lambda_I \to \lambda_O$ with correlation q. The correlation from Δ_O to λ_I through the middle component E_m can be computed by DLCT:

$$\mathrm{C}[\Delta_O \xrightarrow{E_m} \lambda_I] = \frac{\mathrm{DLCT}_{E_m}(\Delta_O, \lambda_I)}{2^{n-1}} = r.$$

Under the independence assumption between E_d, E_m, and E_l, the expected correlation of the DL distinguisher is evaluated by prq^2, and the corresponding data complexity remains $O(p^{-2}r^{-2}q^{-4})$. To be more precise, the correlation estimation, combined with

the theoretical method proposed by Blondeau et al. [8], can be represented as:

$$\mathrm{C}[\Delta_I \xrightarrow{E} \lambda_O] = \sum_{\Delta_O, \lambda_I} \Pr[\Delta_I \xrightarrow{E_d} \Delta_O] \cdot \mathrm{C}[\Delta_O \xrightarrow{E_m} \lambda_I] \cdot (\mathrm{C}[\lambda_I \xrightarrow{E_l} \lambda_O])^2. \quad (1)$$

In practice, DL cryptanalysis often employs MILP-based methods to search for a best single DL trail, i.e.,

$$\max_{\Delta_I, \Delta_O, \lambda_I, \lambda_O} \{\Pr[\Delta_I \xrightarrow{E_d} \Delta_O] \cdot \mathrm{C}[\Delta_O \xrightarrow{E_m} \lambda_I] \cdot (\mathrm{C}[\lambda_I \xrightarrow{E_l} \lambda_O])^2\}, \quad (2)$$

with its correlation estimated based on the DDT, LAT, and DLCT values of each involved Sbox. However, multiple trails may share a given input difference and output mask pair. According to Equation (1), the sum of correlations over all such trails more accurately reflects the total correlation of the distinguisher, which we refer to as the exact correlation. While this aggregation captures the combined effect of all valid trails, it still relies on specific assumptions and thus serves as a theoretical approximation. Nevertheless, for convenience, we refer to this aggregated value as the exact correlation throughout the remainder of this paper.

2.3 Integral Cryptanalysis

Integral cryptanalysis is a powerful technique inspired by the Square attack [9], and was formally introduced by Knudsen and Wagner [10]. It exploits the behavior of a block cipher under a structured set of plaintexts to reveal statistical biases in intermediate states. Specifically, an integral distinguisher is constructed by varying a subset of input bits-called active bits-across all possible values, while fixing the remaining bits. After a few rounds, certain output bits may exhibit the *balanced* property, meaning their XOR sum over all plaintexts is zero, regardless of the key.

Formally, let E be an r-round block cipher with n-bit block size. Let $\mathbb{D}$ be a set of 2^d plaintexts differing in d active bits. If the i^{th} bit of the output satisfies

$$\bigoplus_{PT \in \mathbb{D}} CT_i = 0,$$

where $CT = E^r(PT, K)$, then bit i is said to have a *balanced integral property*. Such properties can be used to distinguish r-round E from an ideal cipher.

An effective approach to constructing integral distinguishers is based on algebraic degree estimation. Suppose the algebraic degree of an r-round block cipher is upper bounded by d_u, and the number of active input bits is d. When $d > d_u$, the output after r rounds cannot form a full-degree polynomial. In this case, the cipher has the integral distinguisher with 2^d chosen plaintexts. This method has been applied to evaluate ciphers such as Keccak and Luffa [11]. Several algebraic degree estimation techniques can be used for this purpose, including Boura and Canteaut's and Carlet's formulas [12, 13], estimations based on the division property [14], and numeric mapping [15]. Among them, the division property based estimation is considered the most precise [16].

3 Differential-Linear Cryptanalysis of Gleeok-128

In [5], the authors give a comprehensively security analysis on Gleeok-128 as well as all the branches from the resistance on distinguishing attacks perspective. As a consequence, the longest valid distinguisher for Gleeok-128 is the integral distinguisher that covers 5 out of 12 rounds. Regarding the single branch, the boomerang distinguishers of Branch1 and Branch2 outperform the integral, linear, and differential distinguishers. The main reason is that the short differential characteristics have high probability and the probability will decrease rapidly with the round growing. Naturally, combining two short but high-probability differential characteristics can derive a good boomerang one. Inspired by this observation, we in this section will adopt another combined cryptanalytic method, DL cryptanalysis, to explore improved distinguishers for branches and the overall PRF.

3.1 A Two-Stage Framework to Explore DL Distinguishers

In this part, we introduce a two-stage framework for the automatic search of DL distinguishers. This framework is applicable to not only the single branch but also the PRF, which can be easily implemented by MILP-aided and SAT-aided tools. In the first stage, we aim to find a series of good DL trail candidates. In the second stage, for each trail, we apply a clustering-based technique to estimate its exact correlation and select the best one as our DL distinguisher. In the following context, we will illustrate in detail the framework applied to the single branch. As for its application to the PRF, we only need to combine the processes of three branches under some conditions.

Stage one: search for optimal DL trails. In this stage, we convert the DL trail search of single branch (i.e., Branch1/Branch2/Branch3 in Gleeok-128) to an MILP[1] problem according to Equation (2). Specifically, we regard the single branch as an SPN-based block cipher and split the R-round cipher to three sub-ciphers: the top one covers R_d rounds, the middle one covers only one round and the bottom sub-cipher covers R_l rounds, where $R = R_d + 1 + R_l$. For the sake of simplification, we denote the three sub-ciphers by E_d, E_m and E_l, respectively. Next we need to model each one separately by MILP:

- The sub-model $\mathcal{M}_d$ describes the R_d-round differential propagation. In this sub-model, an integer variable P is introduced to represent the differential probability $2^{-\mathrm{P}}$;
- The sub-model $\mathcal{M}_l$ describes the R_l-round linear mask propagation, and the involved integer variable C_l represents the linear approximation correlation $2^{-\mathrm{C}_l}$;
- The sub-model $\mathcal{M}_m$ consists of the linearly characterized DLCTs of Sbox layer and the propagation of linear mask through θ and π operations. The integer variable C_m denotes the DL correlation $2^{-\mathrm{C}_m}$.

[1] In our experiments, the MILP-based approach performed better than the SAT-based one in the second stage, e.g., computing the aggregated correlation of a 5-round DL pair of Branch1 took ¡1s with MILP (Gurobi) but ≈46s with SAT. This mainly results from Gurobi's multi-threaded optimization and parameter tuning. Considering overall runtime and model consistency, we recommend using MILP throughout, though combining SAT for the first stage and MILP for the second may further improve efficiency at the cost of integration complexity.

Specially, the output difference from E_d propagates directly to E_m's Sbox layer, the output linear mask of π operation of E_m becomes the input linear mask for E_l's Sbox layer. Thus, we can integrate them to an entire model $\mathcal{M}$ by regarding $\mathcal{M}_m$ as a link of $\mathcal{M}_d$ and $\mathcal{M}_l$, describing the DL trail with round combination $(R_d, 1, R_f)$. According to Equation (2), our aim is to maximize the product of the differential probability for E_d, the correlation for E_m, and the squared correlation for E_l, i.e., $2^{-\mathrm{P}} \cdot 2^{-\mathrm{C}_m} \cdot 2^{-2\mathrm{C}_l}$. Consequently, the model's objective function is

$$\text{minimize: } \mathrm{P} + \mathrm{C}_m + 2 \cdot \mathrm{C}_l.$$

Send this model to an MILP solver (we in this paper use Gurobi optimization[2]), then an R-round DL trail with the best correlation under the round configuration $(R_d, 1, R_l)$ can be found.

Assuming the returned solution yields an objective function value of C, all DL trails under the round configuration $(R_d, 1, R_l)$ exhibit correlations of at most $2^{-\mathrm{C}}$. To collect more high-quality DL trails, we constrain the objective function to C, and iteratively add the *cutting-off inequality* (see Equation (B13) in Supplementary Material), which is originally proposed by Sun et al. [17], to exclude previously generated difference-mask pairs. This process continues until the updated model becomes infeasible (as outlined in Algorithm 2), then a set of difference-mask pairs, all with the same optimal correlation of $2^{-\mathrm{C}}$, is obtained. Next, we move to the second stage.

Algorithm 2 Collect R-round high-quality DL trails

1: **Input:** The round configuration $(R_d, 1, R_l)$
2: **Output:** A set of difference–mask pairs and the common correlation
3: Initialize $\mathbb{P}$ as an empty set
4: Construct an MILP model $\mathcal{M}$ as described above and solve $\mathcal{M}$ using Gurobi
5: Retrieve the correlation weight C and the difference–mask pair (Δ, λ) from $\mathcal{M}$'s current solution
6: Add (Δ, λ) to $\mathbb{P}$
7: Append a cutting–off inequality corresponding to (Δ, λ) to $\mathcal{M}$ and constrain $\mathcal{M}$'s objective function to C
8: **while** $\mathcal{M}$ is feasible **do**
9: Retrieve the new difference–mask pair (Δ, λ) from $\mathcal{M}$'s current solution
10: Add (Δ, λ) to $\mathbb{P}$
11: Append a cutting–off inequality corresponding to (Δ, λ) to $\mathcal{M}$
12: **end while**
13: **return** $(\mathbb{P}, 2^{-\mathrm{C}})$

Stage two: compute the aggregated correlations. In this stage, we aim to compute the exact correlation for each prepared DL trail and select the one with the highest value. As specified by Equation (1), given a difference-mask pair (Δ_I, λ_O), we must enumerate all compatible (Δ_O, λ_I) pairs and compute their signed correlations.

[2] https://www.gurobi.com/

The exact correlation for $\Delta_I \xrightarrow{E} \lambda_O$ is then obtained by summing these signed values. Consequently, modeling the correlation sign of DL trails becomes essential. Note that the correlation sign of the trail $\Delta_I \xrightarrow{E_d} \Delta_O \xrightarrow{E_m} \lambda_I \xrightarrow{E_l} \lambda_O$ is identical to that of $\Delta_O \xrightarrow{E_m} \lambda_I$. This equivalence holds because both the differential trail $\Delta_I \xrightarrow{E_d} \Delta_O$ (contributing probability) and the linear trail $\lambda_I \xrightarrow{E_l} \lambda_O$ (contributing squared correlation) are strictly positive. Since the middle part E_m covers only one round in our DL trail, the correlation sign of $\Delta_O \xrightarrow{E_m} \lambda_I$ can be directly derived from the DLCTs of the Sboxes within E_m. Therefore, we only require signed DLCT modeling for E_m. To achieve this, we re-characterize the Sbox's DLCT representation by introducing an auxiliary variable s to encode the correlation sign as

$$s = \begin{cases} 0 & \text{if the DLCT's entry is positive,} \\ 1 & \text{if the DLCT's entry is negative.} \end{cases} \tag{3}$$

For example, Table A8 shows the DLCT of Gleeok's 3-bit Sbox. All non-zero entries have an absolute value of 4, confirming that the absolute value of correlation of valid DL transition is constantly 1. Therefore, we encode the signed DLCT using 7-dimensional points. For instance, the valid DL transitions 0x1 $\xrightarrow{\text{DLCT}}$ 0x1 and 0x0 $\xrightarrow{\text{DLCT}}$ 0x7 are with entries -4 and 4, encoded as $(0,0,1,0,0,1,1)$ and $(0,0,0,1,1,1,0)$, respectively. Using the algorithm in [18], we obtain a system of inequalities (see Equation (B4) in Supplementary Material B) characterizing the signed DLCT through the MILP-based inequality selection method. The sign (positive or negative) of a trail's correlation is determined by the parity (even or odd) of the number of Sboxes that have a sign of 1. An even count yields positive correlation, and an odd count yields negative correlation. Therefore, if we use a binary variable s to represent the sign of a trail's correlation as

$$\mathrm{s} = \begin{cases} 0 & \text{if the trail's correlation is positive,} \\ 1 & \text{otherwise,} \end{cases}$$

then s can be constrained by

$$\mathrm{s} + \sum_{i=1}^{m} s_i = 2d,$$

where m ($m = 32$ for Gleeok-128) is the number of Sboxes, $s_i \in \{0,1\}$ is the sign of the i^{th} Sbox's correlation and $d \in \mathbb{Z}$ is a dummy integer variable ranging from 0 to $m/2$.

Based on the signed correlation modeling method, we now proceed to compute the exact correlation for a given difference-mask pair. Our core idea is to enumerate all valid trails sharing this common pair under a fixed correlation weight and compute the aggregation effect by summing their signed correlations. We assume that the first stage search (Algorithm 2) has already provided a minimum correlation weight C and a set of difference-mask pairs $\mathbb{P}$. For a pair $(\Delta, \lambda) \in \mathbb{P}$, we aim to compute its aggregated correlation by reusing the MILP model $\mathcal{M}$ constructed in the first stage for finding the best DL trail. To adapt $\mathcal{M}$ for this purpose, we replace the unsigned DLCT constraints

in the middle part E_m with their signed counterparts, as described earlier. We then fix the input difference and output mask to Δ and λ, respectively, and constrain the model's objective function to a correlation weight $\mathrm{C}' \geq \mathrm{C}$. Finally, we configure the solver parameters[3] to enumerate all feasible trails under these constraints as below:

$$\mathcal{M}.\mathsf{Params.PoolSearchMode} = 2$$

and

$$\mathcal{M}.\mathsf{Params.PoolSolutions} = 200000000.$$

This parameter configuration enables Gurobi to generate a solution pool when solving the model $\mathcal{M}$. Each solution corresponds to a unique trail with difference-mask pair (Δ, λ) and signed correlation $(-1)^{\mathrm{s}} \cdot 2^{-\mathrm{C}'}$. Given N total solutions in the pool, we compute the aggregated signed correlation for weight C' as

$$2^{-\mathrm{C}'} \cdot \sum_{i=1}^{\mathrm{N}} (-1)^{\mathrm{s}_i},$$

where s_i denotes the correlation sign of the trail corresponding to the i^{th} solution. By repeating this process for correlation weights from C down to a threshold weight C_t (typically with $\mathrm{C}_t - \mathrm{C} \geq 10$), we approximate the exact correlation of the DL distinguisher $\Delta \xrightarrow{E} \lambda$ through the summation:

$$\sum_{\mathrm{C}'=\mathrm{C}}^{\mathrm{C}_t} \left(2^{-\mathrm{C}'} \cdot \sum_{i} (-1)^{\mathrm{s}_i} \right),$$

where the inner sum is taken over all trails with correlation weight C'. By computing and comparing the aggregated correlation for each pair in $\mathbb{P}$, we select the distinguisher with the maximum absolute correlation as the final DL distinguisher. The overall procedure of the second stage is outlined in Algorithm 3 and a schematic illustration is provided in Figure 3.

Application to the overall PRF. As previously stated, the above two-stage framework can be applied not only to branches (Branch1/Branch2/Branch3) but also to the overall PRF. For the PRF setting, in the first stage, we establish a common round configuration and construct models for each of the three inner branches individually. These models are then integrated into a unified model by constraining them to share identical input difference and output mask, according to the differential propagation rule through the branching operation as well as linear mask propagation rule through the XOR operation. Furthermore, the objective function of this model is to minimize the sum of correlation weights of the three branches. Solving this model with Gurobi provides a set of difference-mask pairs, the minimum correlation weight for PRF as well as the correlation weights for each of the three inner branches, as outlined in Algorithm 2. In the second stage, we compute the aggregated correlations for each branch

[3]For more details, please refer to Gurobi's reference manual at https://docs.gurobi.com/projects/optimizer/en/current/.

Algorithm 3 Select the best one from the prepared DL trails

1: **Input:** The minimal correlation weight C, a set of difference-mask pairs $\mathbb{P}$, and an MILP model $\mathcal{M}$ that describes $(R_d, 1, R_l)$-round DL trail.
2: **Output:** A difference-mask pair and its aggregated correlation.
3: $\mathrm{C}_{best} \leftarrow 0$, $\mathrm{C}_t \leftarrow \mathrm{C} + 10$, $(\Delta_I, \lambda_O) \leftarrow (0, 0)$
4: **for** each (Δ, λ) in $\mathbb{P}$ **do**
5: $\quad \mathrm{C}_{temp} \leftarrow 0$
6: $\quad$ Re-constrain $\mathcal{M}$'s difference-mask pair to (Δ, λ)
7: $\quad$ **for** C' from C to C_t **do**
8: $\quad\quad$ Re-constrain $\mathcal{M}$'s objective function to C'
9: $\quad\quad$ $\mathcal{M}$.Params.PoolSearchMode $\leftarrow 2$
10: $\quad\quad$ $\mathcal{M}$.Params.PoolSolutions $\leftarrow 200000000$
11: $\quad\quad$ Solve $\mathcal{M}$ by Gurobi
12: $\quad\quad$ $\mathrm{N} \leftarrow \mathcal{M}$.SolCount
13: $\quad\quad$ $\mathrm{s}_{temp} \leftarrow 0$
14: $\quad\quad$ **for** i from 0 to $\mathrm{N} - 1$ **do**
15: $\quad\quad\quad$ $\mathcal{M}$.Params.SolutionNumber $\leftarrow i$
16: $\quad\quad\quad$ Retrieve the correlation sign s_i from the i^{th} solution
17: $\quad\quad\quad$ $\mathrm{s}_{temp} \leftarrow \mathrm{s}_{temp} + (-1)^{\mathrm{s}_i}$
18: $\quad\quad$ **end for**
19: $\quad\quad$ $\mathrm{C}_{temp} \leftarrow \mathrm{C}_{temp} + \mathrm{s}_{temp} \cdot 2^{-\mathrm{C}'}$
20: $\quad$ **end for**
21: $\quad$ **if** $|\mathrm{C}_{temp}| > \mathrm{C}_{best}$ **then**
22: $\quad\quad$ $\mathrm{C}_{best} \leftarrow |\mathrm{C}_{temp}|$
23: $\quad\quad$ $(\Delta_I, \lambda_O) \leftarrow (\Delta, \lambda)$
24: $\quad$ **end if**
25: **end for**
26: **return** (Δ_I, λ_O) and C_{best}

individually (Lines 5-18, Algorithm 3), and subsequently derive the overall aggregated correlation for the PRF using the Pilling-up Lemma [19].

3.2 Searching for DL Distinguishers of Gleeok-128

We apply the two-stage framework to search for DL distinguishers of Gleeok-128 and its underlying keyed permutations Branch1/Branch2/Branch3. All the DDTs, LATs, DLCTs as well as the corresponding linear inequality descriptions used in the search can be found in Supplementary Material.

Setting round configuration. The crucial step before searching is to choose an appropriate round configuration. Inspired by the differential and linear behaviors of each branches (see Table 4), we give the following round configuration:

$$(R_d, 1, R_l) = \begin{cases} (\frac{R-1}{2}, 1, \frac{R-1}{2}) & \text{if } R \text{ is odd,} \\ (\frac{R}{2} - 1, 1, \frac{R}{2}) & \text{otherwise,} \end{cases}$$

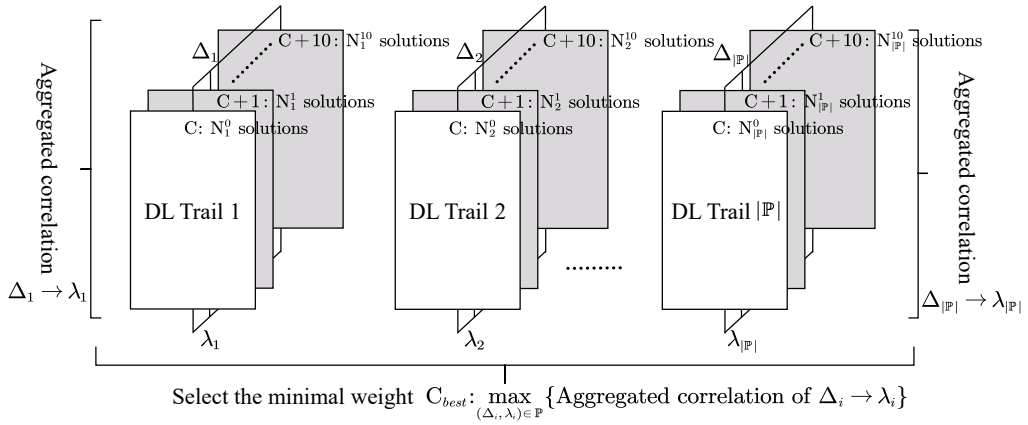

Fig. 3: Overview of Algorithm 3

for Branch1/Branch2/Branch3 since both the differential probability and linear approximation squared correlation decrease rapidly with the rounds growing, while the latter declines at a slightly slower rate compared to the former. For instance, to search for 5-round DL distinguisher of Branch1, we prefer the configuration $(2, 1, 2)$ rather than $(3, 1, 1)$ or $(1, 1, 3)$ since the former one can probably derive the DL trail with a correlation of 2^{-16} whereas that of the latter exhibits 2^{-26} or 2^{-24}.

Table 4: The optimal differential and linear distinguishers [5] up to five rounds for Branch1, Branch2, and Branch3. DC and C^2 denote the differential probability and squared correlation, respectively.

Branch	DC/C^2	#Round				
		1	2	3	4	5
Branch1	DC	2^{-2}	2^{-8}	2^{-24}	2^{-50}	$\leq 2^{-67}$
	C^2	2^{-2}	2^{-8}	2^{-22}	2^{-44}	$\leq 2^{-64}$
Branch2	DC	2^{-2}	2^{-8}	2^{-24}	2^{-50}	$\leq 2^{-66}$
	C^2	2^{-2}	2^{-8}	2^{-22}	2^{-44}	$\leq 2^{-64}$
Branch3	DC	2^{-2}	2^{-8}	2^{-20}	2^{-32}	2^{-36}
	C^2	2^{-2}	2^{-8}	2^{-20}	2^{-30}	2^{-36}

Finding DL distinguishers. Based on the aforementioned round configurations and utilizing our two-stage algorithms, we identify effective DL distinguishers for Gleeok-128 and its underlying branches. In particular, we explore DL distinguishers up to 7/7/8 rounds for its Branch1/Branch2/Branch3 with the squared

correlation[4] $2^{-88.12}/2^{-88.12}/2^{-38.73}$. Note that the best known distinguisher for Branch1/Branch2/Branch3, presented by designers [5], is the 7/7/8-round boomerang/-boomerang/differential distinguisher with probability $2^{-100}/2^{-100}/2^{-64}$. Compared to these results, our DL distinguishers do not cover a larger number of rounds, but the data complexities are significantly reduced. As for the whole PRF Gleeok-128, we can only find DL distinguishers up to 4 rounds limited to the computation resource. We list the result in Table 5.

Table 5: The squared correlation of DL distinguishers for Gleeok-128.

Cipher	#Round							
	1	2	3	4	5	6	7	8
Branch1	1	1	1	$2^{-5.36}$	$2^{-18.86}$	$2^{-44.83}$	$2^{-88.12}$	-
Branch2	1	1	1	$2^{-5.36}$	$2^{-18.86}$	$2^{-44.83}$	$2^{-88.12}$	-
Branch3	1	1	$2^{-0.83}$	$2^{-5.77}$	$2^{-12.42}$	$2^{-22.00}$	$2^{-28.90}$	$2^{-38.73}$
Gleeok-128	1	$2^{-0.39}$	$2^{-11.72}$	$2^{-49.04}$	$\leq 2^{-127.40}$	-	-	-

It is observed from our experimental results that there is only one active Sbox in the input difference as well as the output mask of the high-quality DL trails in general. To investigate the upper bound of 5-round distinguisher, we separately compute the aggregated correlation for three branches by constraining the number of Sbox of input/output to one. The results are 2^{-23}, 2^{-23} and $2^{-17.70}$, respectively. Note that we do not constrain that the input/output of the three models are identical to each other. Consequently, we regard the product of the three aggregated correlations as the upper bound, i.e., $2^{-63.70}$.

Verification on DL distinguishers. All the above DL distinguishers are explored and evaluated by leveraging MILP automatic tool based on our two-stage framework. To validate the correctness of these DL distinguishers and prove the effectiveness of our framework, we conduct some experimental verification on branches as well as the PRF under 100 random keys and record the average value. The results are listed in Table 6. It can be seen that the gap between the estimated values and experimental values is notably small, primarily attributable to statistical bias, which is almost negligible. These results demonstrate both the validity of the high-round DL distinguishers we identified through our search and the efficacy of our methodology.

4 Integral-Based Key-Recovery Attacks on Gleeok-128

In this section, we present the key-recovery attacks on Gleeok-128. We start by identifying new integral distinguishers for its three branches, based on an analysis of the upper

[4]It means that the square of the aggregated correlation of DL distinguishers, which can be intuitively compared to other distinguishing attacks in terms of the data complexity. For instance, the 7-round DL distinguisher of Branch1 with squared correlation $2^{-88.12}$ indicates that distinguishing Branch1 from a random permutation requires approximately $2^{88.12}$ pairs of chosen plaintexts.

Table 6: Verification on DL distinguishers of Gleeok-128 and its underlying branches. Est. and Exp. denote the estimated value and experimental value, respectively.

Cipher	Round	Difference-mask pair	Est.	Exp.
Branch1	5	(0x6000, 0x89df5aa33e89239079ebd0c7d964685d)	$2^{-18.86}$	$2^{-18.85}$
Branch2	5	(0xe0000000000000000000000000, 0x7e656d6aacfa669e41e7a7431f659180)	$2^{-18.86}$	$2^{-18.87}$
Branch3	6	(0x800000000000000, 0x400000400000000000000000)	$2^{-22.00}$	$2^{-21.52}$
Gleeok-128	3	(0x20000000000000000000, 0x2000000000000000000000000000)	$2^{-11.72}$	$2^{-11.70}$

bound on algebraic degree. Notably, the number of rounds covered by the integral distinguishers differs between Branch1/Branch2 and Branch3. Using this observation, we then design a general attack framework by prepending one round before the distinguisher to enable key recovery. Following this, we apply this framework in two settings: with the full codebook and without the full codebook.

4.1 New Integral Distinguishers for Gleeok-128

In [5], the authors employ the SAT-based automatic tool based on division property to search for integral distinguishers of Gleeok-128's underlying branches. Under the data complexity of 2^{127}, they obtain the distinguishers up to 6/6/5/5 rounds for Branch1/Branch2/Branch3/Gleeok-128. However, for more rounds, determining the existence of such distinguishers becomes computationally infeasible within a reasonable time. To address this limitation and enable efficient exploration of longer-round distinguishers, we instead adopt an approach based on evaluating the upper bound of the algebraic degree.

To demonstrate the effectiveness of degree evaluation in identifying integral distinguishers, let us first revisit the data complexity of the 6-/6-round integral distinguisher [5] of Branch1/Branch2. Note the nonlinear layer of Branch1/Branch2 adopts the 3- and 5-bit variants of the χ operations, resulting an algebraic degree of 2 for its round function. Given this, the trivial upper bound on the algebraic degree of 6-round output is $2^6 = 64$. Therefore, by activating 65 input bits, one can construct a structure of 2^{65} plaintexts that guarantees all output bits are balanced after 6 rounds. This leads to a valid 6-round integral distinguisher. Compared to the integral distinguisher proposed in the original design document [5], our newly explored one based on degree evaluation significantly reduced the data complexity (from 2^{127} to 2^{65}).

In order to construct longer integral distinguishers, it is essential to tighten the upper bound on the algebraic degree. To this end, we adopt the iterative degree estimation formula proposed by Carlet, as illustrated in the following theorem, to estimate the algebraic degree of Branch1/Branch2/Branch3 for rounds exceeding 6/6/4, respectively.

Theorem 1 (Carlet-bound [13]) *Let F be a permutation of $\mathbb{F}_2^n$ and let G be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. Then we have*

$$\deg(G \circ F) \leq n - \left\lceil \frac{n - \deg(G)}{\deg(F^{-1})} \right\rceil,$$

where F^{-1} represents the inverse of F.

We give the following example to clarify how to use the Carlet-bound to evaluate the algebraic degree.

Example 1 We have previously estimated the algebraic degree for 6-round Branch1/Branch2 according to the trivial bound. To evaluate the 7-round case, we treat the first 6 rounds as a function G and the 7^{th} round encryption as the function F, thus $\deg(G) = 64$ and $\deg(F^{-1}) = 3$ as the inverses of 3-bit Sbox and 5-bit Sbox have the algebraic degrees of 2 and 3, respectively. According to the theorem, the 7-round algebraic degree, $\deg(G \circ F)$, can be derived:

$$128 - \left\lceil \frac{128 - 64}{3} \right\rceil = 106.$$

By iteratively applying Carlet-bound, we obtain the upper bound on algebraic degree for Branch1/Branch2/Branch3. Moreover, the degree of the overall PRF Gleeok-128 can be determined by the maximum degree among the underlying branches. The results are summarized in Table 7. As shown, these bounds confirm the existence of longer-round integral distinguishers. To be more specific, the 9-/9-/7-/7-round integral distinguisher can be constructed for Branch1/Branch2/Branch3/Gleeok-128 with a data complexity $2^{126}/2^{126}/2^{127}/2^{127}$, such that all the output bits are balanced.

Table 7: The upper bound on algebraic degree of Branch1/Branch2/Branch3 as well as Gleeok-128 up to 10 rounds.

Cipher	#Round									
	1	2	3	4	5	6	7	8	9	10
Branch1	2	4	8	16	32	64	106	120	125	127
Branch2	2	4	8	16	32	64	106	120	125	127
Branch3	3	9	27	81	112	122	126	127	127	127
Gleeok-128	3	9	27	81	112	122	126	127	127	127

4.2 An Integral-Based Attack Framework

In the general key-recovery attacks based on integral distinguishers, it needs to prepend or/and append a certain number of rounds to make guesses about the related key bits that satisfy the distinguisher. However, for Gleeok-128, which consists of three

branches, the adversary need to guess two outputs of the permutations to perform the backward computation from the output. This structural feature significantly complicates the attachment of key-recovery steps at the end of the cipher. For instance, consider appending one additional round (excluding the linear layer) to a 4-round integral distinguisher of Branch1, which can be mounted with a data complexity of 2^{17}. To detect the integral property of one output bit of 4-round Branch1, the attacker must guess 4 related key bits and 4 bits of sum of 5-round outputs of the two other branches. Consequently, for each key guess, the adversary would need to evaluate $(2^4)^{2^{17}} = 2^{2^{19}}$ potential values—far exceeding the 2^{256} complexity of an exhaustive key search.

To avoid guessing the outputs of the other two branches, we can only prepend rounds before the distinguisher. It is observed from Table 7 that Branch1 and Branch2 exhibit a lower degree growth compared to Branch3, which implies that prepending one or two rounds to Branch1/Branch2 does not affect their output balance property under the same data complexity. Therefore, we only need to guess the key bits involved in the prepended rounds of Branch3. For example, the 4-round distinguisher of Gleeok-128 requires a data complexity of 2^{82}. After prepending one/two rounds before each branches, the output of 5-/6-round Branch1/Branch2 still remains balanced under data complexity 2^{82} as the degree of 5-/6-round Branch1/Branch2 is no more than 32/64. Thus we only need to guess the related-key bits in the prepended rounds of Branch3.

Based on this idea, we construct a full-cipher distinguisher by extending the integral distinguisher of Branch3, and use it as the basis for recovering the Branch3's whitening key bits involved in the Sboxes where there are both constant and active output bits. That is, we proceed an $(r+1)$-round attack based on an $(r+0.5)$-round distinguisher, where the 0.5 round includes all operations except for the Sbox layer. The overall framework is illustrated in Figure 4.

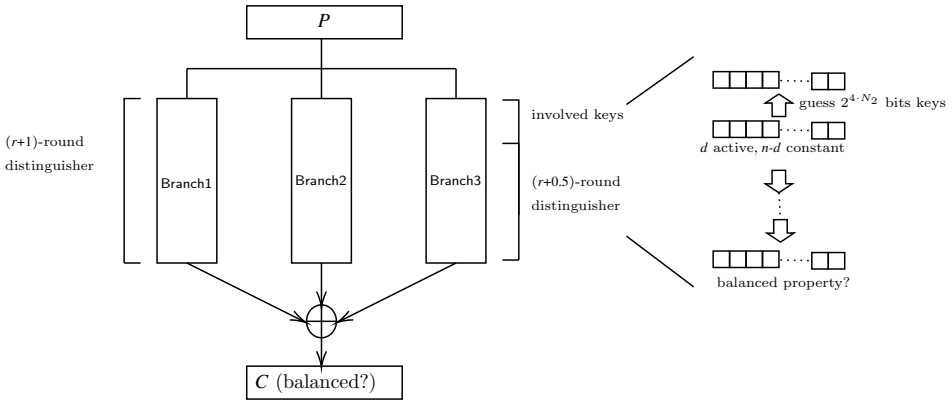

Fig. 4: Overview of the integral-based key-recovery framework on Gleeok-128

Assume the $(r + 0.5)$-round integral distinguisher on Branch3 involves d active input bits. We fix the remaining $n - d$ constant (non-active) bits and enumerate all

possible values of the d active bits, thereby forming a set of 2^d inputs of $(r+0.5)$-round encryption. Suppose these $n-d$ constant bits cover N_1 Sboxes where all the four output bits of the Sbox are constant and N_2 Sboxes where the output of Sbox contains both constant and active bits. We guess $2^{4\cdot N_2}$ bits whitening keys involved in these N_2 Sboxes and use them to partially decrypt the $(r+0.5)$-round inputs, calculating the corresponding 2^d plaintexts. For each guessed key, we can generate 2^d ciphertexts by 2^d queries under the corresponding plaintexts. If the ciphertexts exhibit the balanced property, the guessed key is considered a valid candidate.

4.3 Key-Recovery Attacks on Round-Reduced Gleeok-128

We begin by introducing the relevant notations. In Branch3, the plaintext, the whitening key and the plaintext after XORing with the whitening key are denoted by $X_3 = (x_0, x_1, \cdots, x_{127}) \in \mathbb{F}_2^{128}$, $WK = (wk_0, wk_1, \cdots, wk_{127}) \in \mathbb{F}_2^{128}$ and $Y = (y_0, y_1, \cdots, y_{127}) \in \mathbb{F}_2^{128}$, respectively. Also, Y is the input of Sbox layer in the first round encryption, especially $(y_{4i-4}, y_{4i-3}, y_{4i-2}, y_{4i-1})$ corresponds to the i^{th} $(1 \le i \le 32)$ Sbox.

A 7-round key-recovery attack. Recall that the 6-round distinguisher of Gleeok-128 has data complexity 2^{123}, which can be naturally extended to a 6.5-round integral distinguisher as the 0.5 round encryption excludes the nonlinear layer. By prepending the Sbox layer, we can mount a 7-round key-recovery attack. Particularly, in the Sbox layer of the first round encryption of Branch3, let the 4 out of 5 constant bits cover the output of the first Sbox and the remaining one cover the output of the second Sbox. Moreover, all the remaining 123 output bits of the S-box layer are active, as depicted in Figure 5. Therefore, a set of 2^{123} states at the input of Sbox layer can be generated where (y_0, y_1, y_2, y_3) is constant, (y_4, y_5, y_6, y_7) has 8 values and other 120 bits traverse all the 2^{120} values. Note the output of the second Sbox corresponding to 8 values of (y_4, y_5, y_6, y_7) has three active bits and one constant bit. We need to guess the four whitening key bits (wk_4, wk_5, wk_6, wk_7), for each of the 16 guessed values, we do:

1. Partially decrypt (y_4, y_5, y_6, y_7) to get (x_4, x_5, x_6, x_7) under the guessed key. Then, set (x_0, x_1, x_2, x_3) to a constant and traverse all the 2^{120} values[5] for $(x_8, x_9, \cdots, x_{127})$ to get 2^{123} plaintexts.
2. Query Gleeok-128 for 7-round ciphertexts under the plaintexts.
3. Check the integral property of the multiset of ciphertexts. If it is balanced, we regard the guessed value as a candidate.

In fact, the set of plaintexts derived from different key guesses differ only in the values of (x_4, x_5, x_6, x_7), resulting in 2^{124} different plaintexts where (x_0, x_1, x_2, x_3) are fixed and the remaining 124 bits span all possible combinations. Moreover, the reason why the correct key in the above process cannot be uniquely determined is that certain wrongly guessed keys cause the second Sbox output to preserve the 3-bit active property after partial encryption of the plaintext.

[5] These 120 active bits on plaintext ensure that the output of 7-round Branch1/Branch2 always has the balanced integral property as the 7-round degree is upper bounded by 106.

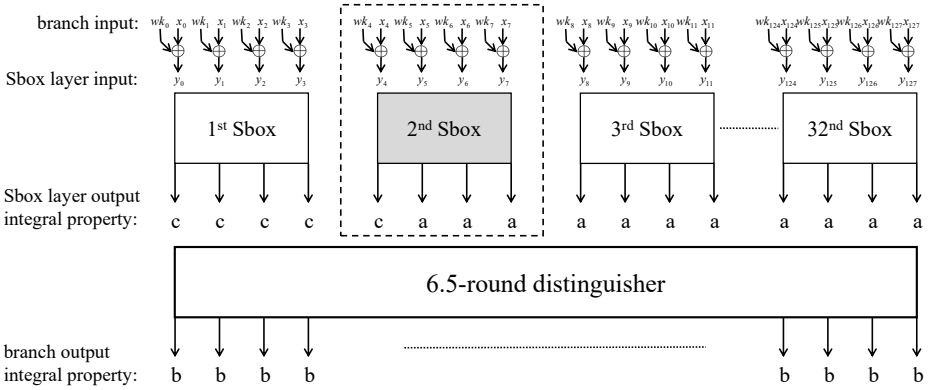

Fig. 5: Overview of Branch3 in the 7-round attack

To clarify how many candidates of the guessed key remain, we conduct experimental tests on the Sbox. Specifically, we construct four sets of 8 inputs for the Sbox, denoted by $\mathbb{S}_i$ $(0 \leq i \leq 3)$, where $\mathbb{S}_i$ can ensure that the i^{th} output bit of the Sbox is constant 0, others are active. When encryption is performed under an incorrect key, it is equivalent to XORing every value in set $\mathbb{S}_i$ with a non-zero constant, where this constant precisely equals the differential value between the correct and wrong keys. Consequently, we apply 16 constants (from 0x0 to 0xf) to each set $\mathbb{S}_i$ to investigate the active properties of the corresponding Sbox output sets, and record cases exhibiting 3-bit activity as shown in Table 8.

Table 8: The input set of the Sbox and the difference of the correct key and survived wrong key. IP is the integral property of output set of $\mathbb{S}_i$ through the Sbox S_4, where 'c' and 'a' mean constant and active properties, respectively.

Input set	$\mathbb{S}_0$	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_3$
IP	caaa	acaa	aaca	aaac
Difference	0x0, 0x5	0x0, 0x1, 0xe	0x0, 0x8, 0xf	0x0, 0xe, 0xf

The results demonstrate that the number of candidates depends on the selection of the 8 inputs to the Sbox, and there are 1-2 wrong keys surviving for different $\mathbb{S}_i$. It is worth noting that the differential intersection of $\mathbb{S}_0$ and $\mathbb{S}_1$ is 0x0, meaning that the intersection of their corresponding candidate key sets yields the correct key. Hence, we can exploit $\mathbb{S}_0$ and $\mathbb{S}_1$ to mount attacks and obtain their respective candidates. Then the correct one can be uniquely determined by taking the intersection of the two candidate sets. The complete attack process to recover (wk_4, wk_5, wk_6, wk_7) is revisited as follows:

1. Under $\mathbb{S}_0$, calculate 16 sets of 2^{123} plaintexts by partial decrytion for 16 guessed keys and query the 7-round Gleeok-128 to generate 16 sets of 2^{123} ciphertexts. Check the integral property for each set of ciphertexts. The guessed keys are retained as candidates if the corresponding sets of ciphertexts have balanced property, forming a set of key candidates $\mathbb{K}_c^0$.
2. Under $\mathbb{S}_1$, similarly to step 1, a set of candidates $\mathbb{K}_c^1$ can be derived.
3. The correct value of (wk_4, wk_5, wk_6, wk_7) is determined as $\mathbb{K}_c^0 \cap \mathbb{K}_c^1$.

As analyzed earlier, the data complexity is 2^{124} chosen plaintexts. Regardless of the time complexity of recovering (wk_4, wk_5, wk_6, wk_7), it is mainly contributed by the $2^{123} \times 2^4 \times 2 = 2^{128}$ 7-round encryptions. According to the key schedule, these four recovered whitening key bits uniquely correspond to four bits of the 256-bit master key. As a result, to recover the full 256-bit master key, the total time complexity is $2^{128} + 2^{252} \approx 2^{252}$.

An improved 7-round key-recovery attack. The aforementioned 7-round attack recovers only four key bits and remaining 252 key bits are recovered by the brute-force attack. To reduce the time complexity, we aim to extract more key bits information by further exploiting the integral property. Recall the above 7-round attack, we fix the one constant bit on the second Sbox so that we recover (wk_4, wk_5, wk_6, wk_7). In fact, we can relocate this constant bit to other Sboxes e.g, the third one while the four constant bits still stay in the first Sbox. In this case, we can recover the whitening key bits $(wk_8, wk_9, wk_{10}, wk_{12})$ with the same 2^{124} chosen plaintexts as used in the previous 7-round attack. That is, we can recover the 124 bits of whitening key (wk_i for $4 \leq i \leq 127$) by sliding the single constant bit from the 2^{nd} Sbox to the 32^{nd} Sbox. As a consequence, the time complexity is significantly decreased from 2^{252} to

$$2^{123} \times 2^4 \times 2 \times 31 + 2^{132} \approx 2^{133.6}$$

whereas the data complexity remains unchanged. Moreover, if we pre-compute the 2^{124} 7-round plaintext-ciphertext pairs and store them in a table, causing a memory complexity of $2^{124} \times 2^5 = 2^{129}$ bytes, then the time complexity can be further reduced to

$$2^{124} + 2^{132} \approx 2^{132}.$$

An 8-round key-recovery attack within the full codebook. If we use the 7.5-round distinguisher with the data complexity 2^{127} to mount the 8-round attack, the full codebook is required as there is only one constant bit in the output of the prepended Sbox layer. In this case, the constant bit can be positioned at any Sbox so that all the 128 whitening key bits can be recovered. As a result, the time complexity is

$$2^{127} \times 2^4 \times 2 \times 32 + 2^{128} = 2^{137}.$$

Consider storing the codebook in the pre-processing phase with a memory complexity of 2^{133} bytes, the time complexity is reduced to $2^{128} + 2^{128} = 2^{129}$, consisting of querying the full codebook and the exhaustive key search over the remaining 128 bits.

5 Exploring New Linear-layer Parameters for Branch3 against Linear Attacks

When searching for DL distinguishers of Gleeok-128's Branch3, we find that the linear distinguishers, as reported in the design document [5], are false for round numbers exceeding 2. Upon further investigation, we identify the reason: the authors incorrectly characterized the linear mask propagation through θ operation when utilizing SAT-based automatic tool. For Branch3 of Gleeok-128, the θ operation can be regarded as a linear transformation over $\mathbb{F}_2^{128}$. Assuming the corresponding transformation matrix is M, the input and output are $\boldsymbol{x} = (x_0, x_1, \cdots, x_{127})^T$ and $\boldsymbol{y} = (y_0, y_1, \cdots, y_{127})^T$, respectively, then $\boldsymbol{y} = M \cdot \boldsymbol{x}$. Denote the mask corresponding to $\boldsymbol{x}$ (resp. $\boldsymbol{y}$) by $\boldsymbol{a} = (a_0, a_1, \cdots, a_{127})^T$ (resp. $\boldsymbol{b} = (b_0, b_1, \cdots, b_{127})^T$). According to the mask propagation rules through XOR and branching operations, the relation of input and output masks can be accurately expressed by the following transformation over $\mathbb{F}_2^{128}$:

$$\boldsymbol{a} = M^T \cdot \boldsymbol{b}.$$

However, the design document may mistakenly describe this relation as $\boldsymbol{b} = M \cdot \boldsymbol{a}$. We re-characterize the mask propagation through θ using above exact relation and update the linear distinguishers as shown in Table 9. Also, the new distinguishers for lower rounds are experimentally verified, supporting the correctness of our revised analysis.

Table 9: The original [5] and our corrected squared correlations of Branch3's linear distinguisher of Gleeok-128.

Round	1	2	3	4	5	6	7	8
[5]	2^{-2}	2^{-8}	2^{-20}	2^{-30}	2^{-36}	2^{-48}	2^{-52}	2^{-64}
Ours	2^{-2}	2^{-8}	$\mathbf{2^{-12}}$	$\mathbf{2^{-16}}$	$\mathbf{2^{-20}}$	$\mathbf{2^{-24}}$	$\mathbf{2^{-28}}$	$\mathbf{2^{-32}}$

Notably, our evaluation reveal that the r-round (for $r > 1$) optimal linear distinguisher has squared correlation $2^{-4 \cdot r}$. This result indicates that the full-round Branch3 can be distinguished with the expected data complexity 2^{48}, exposing a significantly weaker resistance to linear attacks and undermining the overall security of the PRF. Therefore, it is of great significance to strengthen the security of Branch3. Given this vulnerability, it becomes imperative to reinforce the linear cryptanalytic strength of Branch3. A direct and effective countermeasure is to redesign its linear layer.

To enhance the linear resistance of Branch3 while preserving the original design rationale, we aim to redesign the θ and π layers by tuning their parameters. Our goal is to achieve improved linear security without compromising diffusion properties. Following the original approach, the θ operation is constructed by XORing three bit and π is a bit-permutation. Given $n = 128$ and $gcd(p, 128) = 1$, there are in total $64 \times \binom{128}{3}$ candidates. Among them, we retain only those ensuring full diffusion within 2.5 rounds, yielding 4352 viable configurations. As in the original design, we further

evaluate the minimum number of influenced bits at 0.5 rounds before full diffusion, resulting in 4096 candidates that maintain the same level of diffusion as the original design.

To evaluate their linear resistance, we apply SAT-based automatic tools to compute the maximal squared correlation over the first four rounds. Among the 4096 candidates, 3136 reproduce the same linear profile as the original (i.e., squared correlations: 2, 8, 12, 16). Notably, the remaining 960 parameter sets yield a stronger linear profile, indicating a significantly enhanced resistance against linear cryptanalysis in the early rounds. The newly identified parameter sets are listed in Supplementary Material Table C11. To illustrate the enhanced security, we select representative sets from each category and conduct a thorough evaluation of their linear resistance over extended rounds. The results are summarized in Table 10.

Table 10: Squared correlation of the optimal linear distinguishers up to eight rounds for different parameter categories of Branch3.

Category	Count	1	2	3	4	5	6	7	8
Original Design	3136	2^{-2}	2^{-8}	2^{-12}	2^{-16}	2^{-20}	2^{-24}	2^{-28}	2^{-32}
Improved Class A	64	2^{-2}	2^{-6}	2^{-14}	2^{-24}	$2^{-34}/2^{-36}$	$2^{-46}/2^{-48}$	$2^{-54}/2^{-58}$	2^{-66}
Improved Class B	64	2^{-2}	2^{-6}	2^{-14}	2^{-26}	2^{-42}	2^{-54}	2^{-66}	2^{-78}
Improved Class C	256	2^{-2}	2^{-8}	2^{-14}	2^{-28}	2^{-46}	2^{-62}	$2^{-74}/2^{-76}$	2^{-84}
Improved Class D	64	2^{-2}	2^{-8}	2^{-16}	2^{-28}	2^{-36}	2^{-44}	2^{-52}	2^{-60}
Improved Class E	512	2^{-2}	2^{-8}	2^{-16}	2^{-32}	2^{-36}	2^{-48}	2^{-52}	2^{-64}

Notes: We classify all 4096 candidates based on their squared correlation up to round 4. For higher-round evaluations, results are merged within each class.

From the results in Table 10, it is evident that parameter adjustment improves security while preserving the same diffusion property. We select the well-performing Improved Classes B and C as potential candidates. In addition, differential characteristics are also taken into consideration. Ultimately, we recommend Improved Class B as the optimal choice. This recommendation is justified by its markedly stronger resistance to linear attacks compared with the original design, along with its overall balanced performance among all improved classes, while still maintaining strong differential properties. The detailed differential results are presented in Table 11.

6 Conclusion

In this paper, we present the first third-party cryptanalysis of Gleeok-128, providing a comprehensive evaluation that encompasses DL and integral distinguishers, as well as key-recovery attacks targeting the overall PRF. We propose a two-stage MILP-based method for accurately identifying high-quality DL distinguishers, achieving reduced data complexity compared to previous results. By analyzing algebraic degree bounds, we construct the longest known integral distinguishers for Gleeok-128 and design an attack framework that addresses the inherent challenges posed by its parallel branching structure. Moreover, we reveal weaknesses in Branch3's linear resistance caused by

Table 11: The probability of optimal differential characteristics of Branch3 for the Improved Class B and C.

Category	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Original Design	2^{-2}	2^{-8}	2^{-20}	2^{-32}	2^{-36}	2^{-48}	2^{-52}	2^{-64}
Improved Class B	2^{-2}	2^{-8}	2^{-15}	$2^{-27}/2^{-25}$	$2^{-44}/2^{-45}$	$\leq 2^{-62}$	–	–
Improved Class C	2^{-2}	2^{-8}	2^{-12}	2^{-16}	2^{-20}	2^{-24}	2^{-28}	2^{-32}

Notes: (1) The notation "$\leq$" denotes that the reported value is a upper bound, since the optimal solution could not be obtained within the given time limit. (2) A dash (−) indicates that the result could not be obtained within the given time limit.

incorrect mask propagation modeling in the original specification, and propose optimized linear-layer parameters that significantly enhance its security. Furthermore, the proposed frameworks in this work can naturally extend to Gleeok-256. Our results do not threaten the security of Gleeok-128, but we believe these contributions not only advance the cryptanalysis of Gleeok-128, but also provide a deeper understanding for designing the multi-branch ciphers.

Acknowledgements. We would like to thank all the anonymous reviewers for their professional and insightful comments, which help us to significantly improve the quality of this paper. This work was supported by the National Natural Science Foundation of China (Grant No. 62272147, 12471492, 62072161, and 12401687), the Innovation Group Project of the Natural Science Foundation of Hubei Province of China (Grant No. 2023AFA021), the National Key Research and Development Project (Grant No. 2018YFA0704705), Shandong Provincial Natural Science Foundation (Grant No. ZR2024QA205) and the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-035).

Appendix A Truth table, DDT, LAT, and DLCT of S_3, S_4, and S_5

In this section, we first give the truth tables of S_3, S_4, and S_5, and then present the DDT, LAT, and DLCT, respectively. Based on the entries in these tables, feasible points can be generated to represent valid differential, linear, or DL transitions, which will be further used to construct MILP-based models.

Table A1: Truth tables of Sboxes used in Branch1, Branch2, and Branch3. All Sboxes are given in hexadecimal.

(a) Sbox S_3

x	0	1	2	3	4	5	6	7
$S_3(x)$	0	5	3	2	6	1	4	7

(b) Sbox S_4

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S_4(x)$	1	0	2	4	3	8	6	d	9	a	b	e	f	c	7	5

(c) Sbox S_5

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S_5(x)$	0	5	a	b	14	11	16	17	9	c	3	2	d	8	f	e
x	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
$S_5(x)$	12	15	18	1b	6	1	4	7	1a	1d	10	13	1e	19	1c	1f

Table A2: DDT for S_3

Δ_{in} \ Δ_{out}	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	0	2	0	2	0	2
2	0	0	2	2	0	0	2	2
3	0	2	2	0	0	2	2	0
4	0	0	0	0	2	2	2	2
5	0	2	0	2	2	0	2	0
6	0	0	2	2	2	2	0	0
7	0	2	2	0	2	0	0	2

Table A3: DDT for S_4

Δ_{in} \ Δ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	4	0	2	2	0	0	0	0	4	0	0	0	0
2	0	0	2	2	4	4	0	0	2	2	0	0	0	0	0	0
3	0	2	2	0	0	2	0	2	0	0	2	2	0	0	4	0
4	0	0	2	0	2	0	4	0	2	2	0	2	2	0	0	0
5	0	0	2	2	0	4	0	0	0	4	0	0	0	0	2	2
6	0	2	2	0	2	0	0	2	0	0	0	0	2	2	2	2
7	0	2	0	0	0	0	2	4	0	0	2	0	4	2	0	0
8	0	2	0	0	2	0	0	0	4	2	4	0	2	0	0	0
9	0	0	0	2	0	0	0	2	0	2	2	2	2	0	0	4
a	0	2	0	0	2	0	0	0	0	2	2	2	0	2	4	0
b	0	0	2	0	0	0	2	0	2	0	2	2	0	2	0	4
c	0	2	2	2	0	2	0	0	0	0	2	0	2	2	2	0
d	0	2	0	2	0	0	2	2	2	2	0	0	0	2	0	2
e	0	0	0	0	0	2	4	2	4	0	0	0	0	2	0	2
f	0	0	0	2	4	0	0	2	0	0	0	2	2	2	2	0

Table A4: DDT for S_5

Δ_{in} \ Δ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	8	0	8	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0
5	0	4	0	4	0	0	0	0	0	0	0	0	0	0	4	0	4	0	4	0	0	0	0	0	0	0	0	0	0	4	0	4
6	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0
7	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
8	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4
a	0	0	4	4	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4
b	0	4	4	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4
c	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0
d	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0
e	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
f	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	4	4	4	4	0	0	0	0	4	4
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
14	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4
15	0	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4	0
16	0	0	4	0	4	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4	0
17	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0
18	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
19	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
1a	0	0	0	0	0	0	0	0	4	4	0	0	0	0	4	4	4	4	0	0	0	0	4	4	0	0	0	0	0	0	0	0
1b	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0
1c	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
1d	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0
1e	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0
1f	0	2	2	0	2	0	0	2	2	0	0	2	0	2	2	0	2	0	0	2	0	2	2	0	0	2	2	0	2	0	0	2

Table A5: LAT for S_3

λ_{in} \ λ_{out}	0	1	2	3	4	5	6	7
0	4	0	0	0	0	0	0	0
1	0	2	0	2	0	-2	0	2
2	0	0	2	-2	0	0	2	2
3	0	2	-2	0	0	2	2	0
4	0	0	0	0	2	2	-2	2
5	0	-2	0	2	2	0	2	0
6	0	0	2	2	-2	2	0	0
7	0	2	2	0	2	0	0	-2

Table A6: LAT for S_4

λ_{in} \ λ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-4	-4	0	2	-2	2	-2	2	2	-2	-2	0	0	0	0
2	0	0	2	2	4	0	2	-2	-2	-2	0	0	2	-2	0	4
3	0	-4	2	-2	-2	-2	0	0	0	0	2	2	2	-2	-4	0
4	0	2	0	-2	4	-2	0	-2	0	2	4	2	0	2	0	-2
5	0	-2	4	-2	2	0	-2	0	-2	0	-2	-4	0	2	0	-2
6	0	-2	2	-4	0	2	2	0	2	0	0	2	-2	0	4	2
7	0	2	2	0	-2	-4	4	-2	0	-2	-2	0	-2	0	0	-2
8	0	2	2	0	2	0	0	2	4	2	-2	0	2	-4	0	-2
9	0	2	-2	-4	0	-2	-2	0	2	-4	0	-2	2	0	0	2
a	0	-2	0	2	2	0	2	4	2	-4	2	0	0	2	0	-2
b	0	-2	0	2	0	-2	-4	-2	0	-2	0	2	0	-2	4	-2
c	0	0	2	2	-2	2	0	-4	4	0	2	-2	2	2	0	0
d	0	0	-2	-2	0	4	2	-2	-2	-2	0	0	2	-2	0	-4
e	0	0	0	0	-2	-2	2	2	-2	2	2	-2	4	0	4	0
f	0	0	0	0	0	0	0	0	0	0	4	-4	-4	-4	0	0

Table A7: LAT for S_5

λ_{in} \ λ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	8	0	0	0	0	0	0	0	4	0	4	0	-4	0	4	0	-8	0	0	0	0	0	0	0	4	0	4	0	-4	0	4
2	0	0	8	-8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4	0	0	-4	-4	0	0	4	4
3	0	0	-8	0	0	0	0	0	0	4	0	4	0	4	0	-4	0	0	4	-4	0	0	4	4	0	4	-4	0	0	4	4	0
4	0	0	0	0	8	4	-8	4	0	0	0	0	0	4	0	4	0	0	0	0	0	-4	0	-4	0	0	0	0	0	4	0	4
5	0	0	0	0	8	4	8	-4	0	-4	0	4	0	0	0	0	0	0	0	0	0	-4	0	4	0	-4	0	4	0	0	0	0
6	0	0	0	0	-8	4	0	-4	0	0	0	0	0	4	0	4	0	0	4	4	0	-4	4	0	0	0	4	4	0	4	-4	0
7	0	0	0	0	8	-4	0	-4	0	4	0	-4	0	0	0	0	0	0	4	4	0	4	4	0	0	4	4	0	0	0	-4	-4
8	0	0	0	0	0	0	0	0	8	0	4	-4	-8	0	4	-4	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4
9	0	8	0	0	0	0	0	0	0	4	-4	0	0	-4	-4	0	0	8	0	0	0	0	0	0	0	0	-4	4	0	0	4	4	0
a	0	0	0	0	0	0	0	0	8	0	4	-4	8	0	-4	4	0	0	-4	-4	0	0	4	4	0	0	0	0	0	0	0	0
b	0	0	0	8	0	0	0	0	0	4	-4	0	0	4	4	0	0	0	-4	4	0	0	4	4	0	-4	0	-4	0	-4	0	4
c	0	0	0	0	0	4	0	4	-8	0	4	-4	0	4	-4	0	0	0	0	0	0	4	0	4	0	0	4	4	0	-4	4	0
d	0	0	0	0	0	4	0	-4	0	-4	-4	0	8	0	4	-4	0	0	0	0	0	4	0	-4	0	4	4	0	0	0	4	4
e	0	0	0	0	0	4	0	4	8	0	-4	4	0	4	-4	0	0	0	4	4	0	4	-4	0	0	0	0	0	0	-4	0	-4
f	0	0	0	0	0	-4	0	4	0	4	4	0	8	0	4	-4	0	0	4	4	0	-4	-4	0	0	-4	0	4	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	4	4	-4	4	-8	0	0	0	4	4	-4	4
11	0	-8	0	0	0	0	0	0	0	4	0	4	0	-4	0	4	0	0	0	0	4	4	4	-4	0	-4	0	4	4	0	4	0
12	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	-4	-4	4	0	0	0	0	-4	4	-4	4	0	0
13	0	0	8	0	0	0	0	0	0	4	0	4	0	4	0	-4	0	0	-4	-4	4	-4	0	0	0	4	4	0	4	0	0	-4
14	0	0	0	0	0	-4	0	-4	0	0	0	0	0	4	0	4	8	0	0	0	4	0	-4	0	8	0	0	0	-4	0	4	0
15	0	0	0	0	0	-4	0	4	0	-4	0	4	0	0	0	0	0	8	0	0	4	0	4	0	0	4	0	4	-4	-4	-4	4
16	0	0	0	0	0	-4	8	4	0	0	0	0	0	4	0	4	0	0	4	-4	-4	0	0	-4	0	0	4	-4	4	0	0	4
17	0	0	0	0	0	4	8	4	0	4	0	-4	0	0	0	0	0	0	-4	4	4	0	0	-4	0	4	-4	0	-4	4	0	0
18	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4	-8	0	0	0	4	4	-4	4	0	0	4	-4	-4	4	0	0
19	0	8	0	0	0	0	0	0	-4	4	0	0	4	4	0	0	0	0	0	0	4	4	4	-4	0	-4	-4	0	4	0	0	-4
1a	0	0	0	0	0	0	0	0	0	0	4	4	0	0	-4	-4	0	0	-4	4	-4	4	0	0	8	0	0	0	4	4	-4	4
1b	0	0	0	8	0	0	0	0	0	-4	4	0	0	-4	-4	0	0	0	4	4	4	-4	0	0	0	4	0	-4	4	0	4	0
1c	0	0	0	0	0	4	0	4	0	0	4	4	0	-4	4	0	8	0	0	0	-4	0	4	0	0	0	4	-4	-4	0	0	-4
1d	0	0	0	0	0	4	0	-4	0	4	4	0	0	0	4	4	0	8	0	0	-4	0	-4	0	0	4	-4	0	4	-4	0	0
1e	0	0	0	0	0	4	0	4	0	0	-4	-4	0	-4	4	0	0	0	4	-4	4	0	0	4	8	0	0	0	4	0	-4	0
1f	0	0	0	0	0	-4	0	4	0	-4	-4	0	0	0	4	4	0	0	-4	4	-4	0	0	4	0	4	0	4	4	4	4	-4

Table A8: DLCT for S_3

Δ_{in} \ λ_{out}	0	1	2	3	4	5	6	7
0	4	4	4	4	4	4	4	4
1	4	-4	0	0	0	0	0	0
2	4	0	-4	0	0	0	0	0
3	4	0	0	-4	0	0	0	0
4	4	0	0	0	-4	0	0	0
5	4	0	0	0	0	-4	0	0
6	4	0	0	0	0	0	-4	0
7	4	0	0	0	0	0	0	-4

Table A9: DLCT for S_4

Δ_{in} \ λ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	-4	-4	0	4	-4	-4	4	4	0	0	-4	0	0	0	0
2	8	0	4	0	0	0	-4	0	4	0	0	0	-4	0	-8	0
3	8	0	-4	-4	0	0	0	0	0	-4	4	0	0	4	0	-4
4	8	4	0	0	0	-4	0	0	0	4	-4	-4	-4	0	0	0
5	8	-4	0	-4	0	0	0	0	0	0	0	0	0	4	-8	4
6	8	0	0	0	-4	0	0	-4	0	0	0	0	4	0	0	-4
7	8	0	0	0	-4	0	0	-4	0	-4	-4	0	0	0	8	0
8	8	4	4	0	4	0	0	-4	-4	-4	0	0	-4	-4	0	0
9	8	-4	-4	4	0	0	0	-4	-4	0	0	0	0	0	0	4
a	8	0	0	-4	0	-4	0	0	-4	0	4	4	0	0	0	-4
b	8	0	-4	0	0	4	0	0	-4	4	0	-4	0	-4	0	0
c	8	0	0	-4	0	0	-4	0	0	-4	0	0	4	0	0	0
d	8	-4	0	0	0	0	4	0	0	0	-4	0	0	-4	0	0
e	8	0	0	0	-4	4	4	4	0	0	-4	-4	-4	-4	0	0
f	8	0	0	4	-4	-4	-4	0	0	0	0	4	0	0	0	-4

Table A10: DLCT for S_5

Δ_{in} \ λ_{out}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	-16	0	0	0	0	0	0	16	-16	0	0	0	0	0	0	16	-16	0	0	0	0	0	16	-16	0	0	0	0	0	0	0
2	16	16	-16	-16	0	0	0	0	0	0	0	0	0	0	0	0	16	16	-16	-16	0	0	0	0	0	0	0	0	0	0	0	0
3	16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	16	16	16	16	-16	-16	-16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	16	-16	0	0	0	0	0	0	0	0	0	0	16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	16	16	-16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	16	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	16	0	16	0	16	0	16	0	-16	0	-16	0	-16	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	16	0	0	16	0	0	0	0	-16	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	16	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	-16	0	0	0	0	0	0
b	16	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	-16	0	0	0	0	0
c	16	0	16	0	-16	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	16	0	0	16	0	0	0	0	0	0	0	0	0	-16	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	16	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	16	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	16	0	0	0	16	0	0	0	16	0	0	0	16	0	0	0	-16	0	0	0	-16	0	0	0	-16	0	0	0	-16	0	0	0
11	16	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0
12	16	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0
13	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	16	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	-16	0	0	0	0	0	0	0	0	0	0
15	16	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	-16	0	0
16	16	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	-16	0	0	0	0	0	0	0	0
17	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	16	0	0	0	16	0	0	0	-16	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	16	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1a	16	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	-16	0
1b	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0
1c	16	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1d	16	0	0	0	0	0	0	0	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1e	16	0	0	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1f	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-16

Appendix B Inequalities Used in MILP Models

The inequalities presented in this Supplementary Material are used to model the DDT, LAT and DLCT of Sboxes within the MILP model. These inequalities were obtained by applying the algorithm in [18], following the MILP-based modelling method of Sasaki and Todo [20], which also enables the selection of a minimal subset of inequalities.

- Encoding the DDT of S_3 with $(x_0, x_1, x_2, y_0, y_1, y_2, p)$, where $(x_0, x_1, x_2) \xrightarrow{S_3} (y_0, y_1, y_2)$ denotes the valid differential transition with probability 2^{-2p}. The DDT is characterized as:

$$\begin{cases} -3x_2 - 3x_1 + x_0 - 3y_2 - 3y_1 + y_0 + 10p \geq 0 \\ -3x_2 + x_1 - 3x_0 - 3y_2 + y_1 - 3y_0 + 10p \geq 0 \\ x_2 - 3x_1 - 3x_0 + y_2 - 3y_1 - 3y_0 + 10p \geq 0 \\ 4x_2 + 7x_1 + 2x_0 + 4y_2 + y_1 + 6y_0 - 8p \geq 0 \\ 4x_2 + 7x_1 + 6x_0 + 4y_2 + y_1 + 2y_0 - 8p \geq 0 \\ 6x_2 + x_1 + 4x_0 + 2y_2 + 7y_1 + 4y_0 - 8p \geq 0 \\ 2x_2 + x_1 + 4x_0 + 6y_2 + 7y_1 + 4y_0 - 8p \geq 0 \end{cases} \tag{B1}$$

- Encoding the LAT of S_3 with $(x_0, x_1, x_2, y_0, y_1, y_2, c)$, where $(x_0, x_1, x_2) \xrightarrow{S_3} (y_0, y_1, y_2)$ denotes the valid linear approximation with correlation 2^{-c}. The DDT is characterized as:

$$\begin{cases} -x_2 + 3x_1 - x_0 + 6y_2 + 3y_1 + 6y_0 - 4c \geq 0 \\ 6x_2 + 3x_1 - x_0 - y_2 + 3y_1 + 6y_0 - 4c \geq 0 \\ 4x_2 + x_1 + 8x_0 + 4y_2 + 7y_1 - 2y_0 - 6c \geq 0 \\ 3x_2 + 6x_1 + 6x_0 + 3y_2 - y_1 - y_0 - 4c \geq 0 \\ -3x_2 - 3x_1 + x_0 - 3y_2 - 3y_1 + y_0 + 10c \geq 0 \\ x_2 - 3x_1 - 3x_0 + y_2 - 3y_1 - 3y_0 + 10c \geq 0 \\ -3x_2 + x_1 - 3x_0 - 3y_2 + y_1 - 3y_0 + 10c \geq 0 \end{cases} \tag{B2}$$

- Encoding the unsigned DLCT of S_3 with $(x_0, x_1, x_2, y_0, y_1, y_2)$, where $(x_0, x_1, x_2) \xrightarrow{S_3} (y_0, y_1, y_2)$ denotes the valid DL approximation with constant correlation 1. The unsigned DLCT is characterized as:

$$\begin{cases} -2x_2 + 4x_1 - 6x_0 - y_2 - 7y_1 + 4y_0 \geq -8 \\ 4x_2 - 6x_1 - 2x_0 - 7y_2 + 4y_1 - y_0 \geq -8 \\ -6x_2 - 2x_1 + 4x_0 + 4y_2 - y_1 - 7y_0 \geq -8 \end{cases} \tag{B3}$$

- Encoding the signed DLCT of S_3 with $(x_0, x_1, x_2, y_0, y_1, y_2, s)$, where $s = 0$ indicates a positive correlation and $s = 1$ indicates a negative correlation. The signed DLCT is characterized as:

$$\begin{cases} -8x_2 - 3x_1 - 3x_0 + 5y_2 - 14y_1 + 2y_0 + 7s \geq -14 \\ -2x_2 - 8x_1 - 2x_0 - y_2 + 6y_1 - 11y_0 + 6s \geq -12 \\ -4x_2 - 4x_1 - 4x_0 - 11y_2 + 2y_1 - y_0 + 10s \geq -12 \\ -5x_2 + 9x_1 - 3x_0 + 4y_2 - 9y_1 + 2y_0 - 7s \geq -9 \\ 5x_2 - 8x_1 + 3x_0 - 4y_2 + 9y_1 - 2y_0 - 9s \geq -8 \end{cases} \tag{B4}$$

- Encoding the DDT of S_4 with $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, p_0, p_1)$, where $(x_0, x_1, x_2, x_3) \xrightarrow{S_4} (y_0, y_1, y_2, y_3)$ denotes the valid differential transition with probability $2^{-(2p_0+p_1)}$. This DDT is characterized as:

$$
\begin{cases}
-17x_3 - 29x_2 + 7x_1 - 19x_0 - 8y_3 + 11y_2 - 24y_1 + 5y_0 + 52p_0 + 85p_1 \geq 0 \\
-3x_3 - 4x_2 - 3x_1 + 6x_0 - y_3 + 4y_2 - 12y_0 - 30p_0 - 22p_1 \geq -41 \\
-11x_3 - 25x_2 - x_1 - 3x_0 - 14y_3 - 6y_2 - 7y_1 - 9y_0 + 51p_0 + 75p_1 \geq 0 \\
-8x_3 + 24x_2 + 14x_1 + 38x_0 + 15y_3 - 17y_2 - 39y_1 + y_0 + 5p_0 - 5p_1 \geq -30 \\
10x_3 + 21x_2 - 2x_1 - x_0 - 20y_3 - 22y_2 - 10y_1 + 2y_0 + 16p_0 - 6p_1 \geq -39 \\
12x_3 - 9x_2 - 12x_1 - 4x_0 + 3y_3 + 8y_2 - 12y_1 - 9y_0 + 2p_0 - 2p_1 \geq -36 \\
18x_3 + 15x_2 + 18x_1 - 3x_0 - 17y_3 + 3y_2 - y_1 - y_0 + 9p_0 - 9p_1 \geq -13 \\
6x_3 + 3x_2 + 17x_1 + 14x_0 + 3y_3 - 14y_2 - 11y_1 - 10y_0 + 2p_0 - 5p_1 \geq -20 \\
4x_3 + 5x_2 + 6x_0 + 2y_3 - 5y_2 + y_1 - 2y_0 - 20p_0 - 26p_1 \geq -27 \\
11x_3 - 2x_2 + 4x_1 - 23x_0 + 11y_3 - y_2 + 22y_1 + 21y_0 + 4p_0 - 4p_1 \geq -7 \\
26x_3 + x_2 - 6x_1 + 22x_0 - 12y_3 + 13y_2 - 8y_1 + 4y_0 - 85p_0 - 61p_1 \geq -79 \\
-37x_3 + 17x_2 - 3x_1 - 4x_0 + 42y_3 - 2y_2 - 4y_1 - 21y_0 - 28p_0 + 11p_1 \geq -57 \\
-6x_3 - 2x_2 - 6x_1 + 8x_0 + 4y_3 - 4y_2 - 8y_1 + 10y_0 + p_0 - 9p_1 \geq -25 \\
3x_3 - 11x_2 - 2x_1 - 8x_0 + 11y_3 - 11y_2 + 8y_1 - 3y_0 + 2p_0 - 7p_1 \geq -31 \\
13x_3 + 12x_2 + 6x_1 - x_0 - 16y_3 - 9y_2 + 12y_1 - 3y_0 - 57p_0 - 52p_1 \geq -65 \\
-7x_3 - 12x_2 - 16x_1 - 4x_0 - 9y_3 + 16y_2 + 16y_1 - 4y_0 + 2p_0 - 6p_1 \geq -42 \\
8x_3 - 27x_2 + 8x_1 + 19x_0 + 49y_3 + 51y_2 + 14y_1 + 5y_0 - 42p_0 + 10p_1 \geq -4 \\
-8x_3 + 3x_2 + 2x_1 + 2x_0 + 12y_3 + 16y_2 + 13y_1 + 16y_0 - 4p_0 - 8p_1 \geq 0 \\
-20x_3 - 20x_2 - 8x_1 + 6x_0 + 14y_3 - 2y_2 + 10y_1 + 20y_0 + 3p_0 - 11p_1 \geq -41 \\
-27x_3 + 54x_2 - 2x_1 - 27x_0 + 23y_3 - 2y_2 + 24y_1 + 7y_0 - 15p_0 + 15p_1 \geq -19
\end{cases}
\tag{B5}
$$

- Encoding the LAT of S_4 with $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, c_0, c_1)$, where $(x_0, x_1, x_2, x_3) \xrightarrow{S_4} (y_0, y_1, y_2, y_3)$ denotes the valid linear approximation with correlation $2^{-(c_0+c_1)}$. This LAT is characterized as:

$$
\begin{cases}
14x_3 - 3x_2 + 14x_1 - 7x_0 + 2y_3 - 11y_2 + y_1 - 7y_0 + 14c_0 + 12c_1 \geq 0 \\
-3x_3 + x_1 + x_0 + 8y_3 + 9y_2 + 9y_1 + 9y_0 - 3c_0 - 3c_1 \geq 0 \\
-9x_3 + x_2 - 10x_1 - 6x_0 - 3y_3 - 3y_2 + y_1 + 10y_0 + 15c_0 - 8c_1 \geq -15 \\
20x_3 - x_2 + 20x_1 + 15x_0 + 7y_3 + 5y_2 - 3y_1 + 20y_0 - 3c_0 - 13c_1 \geq 0 \\
-4x_3 - 3x_2 - 4x_0 - 9y_3 - 5y_2 - 9y_1 + 6y_0 + 32c_0 - 7c_1 \geq 0 \\
-47x_3 + x_2 + 4x_1 - 47x_0 + 7y_3 + 5y_2 + 8y_1 + 11y_0 + 77c_0 + 10c_1 \geq 0 \\
2x_3 + 2x_2 - 6x_1 - 6x_0 + y_3 - 6y_2 - 7y_1 + 4y_0 + 12c_0 - 5c_1 \geq -12 \\
-8x_3 + 13x_2 - 3x_1 + 15x_0 - 10y_3 + 20y_2 - 28y_1 - 4y_0 + 25c_0 + 24c_1 \geq 0 \\
-x_3 + 8x_2 + 19x_1 + 6x_0 - 4y_3 - 16y_2 - 12y_1 + 13y_0 + 8c_0 + 15c_1 \geq 0 \\
4x_3 - x_2 - 3x_1 - 6x_0 + 4y_3 - 5y_2 - 11y_1 - 11y_0 + 13c_0 + 11c_1 \geq -13 \\
-6x_3 - 2x_2 - 5x_1 - x_0 + 10y_3 + 8y_2 + 3y_0 + 4c_0 + 5c_1 \geq 0 \\
17x_3 - x_2 + 9x_1 - 3x_0 - 23y_3 - 7y_2 + 16y_1 + 5y_0 + 8c_0 + 21c_1 \geq 0 \\
-2x_3 + 11x_2 - 3x_1 - 7x_0 - 2y_3 - 5y_2 + 11y_1 + 3y_0 + 8c_0 + 9c_1 \geq 0 \\
13x_3 + 13x_2 - x_1 + 8x_0 + 13y_3 + 5y_2 + 13y_1 - 3y_0 - 3c_0 - 6c_1 \geq 0 \\
-x_3 + 4x_2 - x_1 + 4x_0 + 4y_3 - 4y_2 + 3y_1 - 4y_0 + 3c_0 - c_1 \geq -4 \\
6x_3 - 8x_1 + 2x_0 - 4y_3 + 8y_2 - y_1 + 4y_0 + 5c_0 + 7c_1 \geq 0 \\
-11x_3 - 12x_2 - 9x_1 - 12x_0 + 6y_3 - y_2 + 3y_1 - y_0 + 20c_0 - 6c_1 \geq -20 \\
2x_3 - 5x_2 - 5x_1 + 2x_0 - 5y_3 - 3y_2 + 5y_1 - 6y_0 + 10c_0 - c_1 \geq -10 \\
-x_3 + 4x_2 + 4x_1 - 2x_0 - 4y_3 - 3y_2 + 4y_1 - 4y_0 + 6c_0 - 2c_1 \geq -6 \\
-x_3 + 3x_2 + 3x_1 - x_0 + 3y_3 + 2y_2 - 3y_1 - 3y_0 + 4c_0 - 3c_1 \geq -4 \\
2x_3 + 2x_2 - 18x_1 - 22x_0 - 20y_3 + 5y_2 + 5y_1 + 3y_0 + 51c_0 + 4c_1 \geq 0 \\
2x_3 - 9x_2 - 9x_1 + 5x_0 + 7y_3 - y_2 + 4y_1 + 4y_0 + 9c_0 + 7c_1 \geq -1 \\
-x_3 + 4x_2 - x_1 + 4x_0 - 4y_3 + 4y_2 - 3y_1 - 4y_0 + 5c_0 - c_1 \geq -5 \\
x_3 - 3x_2 + x_1 - 3x_0 - 3y_3 + 3y_2 - 2y_1 - 3y_0 + 7c_0 - 3c_1 \geq -7 \\
-7x_3 - 7x_2 + 2x_1 + 3x_0 - 5y_3 - 8y_2 - 7y_1 - y_0 + 16c_0 - 4c_1 \geq -16 \\
26x_3 + 26x_2 + 23x_1 + 17x_0 - 9y_3 - 9y_2 - 3y_1 - y_0 - c_0 - 3c_1 \geq 0 \\
-3x_3 - 16x_2 + 6x_1 + 10x_0 - 2y_3 + 10y_2 + 11y_1 - 22y_0 + 21c_0 + 19c_1 \geq 0 \\
x_3 - 8x_2 - 8x_1 + x_0 + 3y_3 + 2y_2 - 8y_1 - 8y_0 + 37c_0 - 8c_1 \geq 0
\end{cases}
\tag{B6}
$$

- Encoding the unsigned DLCT of S_4 with $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, c)$, where $(x_0, x_1, x_2, x_3) \xrightarrow{S_4} (y_0, y_1, y_2, y_3)$ denotes the valid DL approximation with correlation 2^{-c}. This unsigned DLCT is characterized as:

$$
\begin{cases}
31x_3 - 24x_2 + x_1 - 4x_0 - 4y_3 - 6y_2 - 10y_1 - 6y_0 - 10c \geq -32 \\
17x_3 + 12x_2 - 9x_1 - 10x_0 - 11y_3 - 3y_2 - y_1 + 2y_0 - 28c \geq -32 \\
-50x_3 + 12x_2 + 6x_1 + 10x_0 + 6y_3 + y_2 + 3y_1 - 7y_0 - c \geq -45 \\
-50x_3 - 6x_2 - x_1 - 3x_0 - 2y_3 + 48y_2 + 43y_1 + 47y_0 + 49c \geq -12 \\
23x_3 + 17x_2 - 50x_1 + 20x_0 + 25y_3 + 4y_2 - 5y_1 - 44y_0 + 6c \geq -50 \\
19x_3 + 10x_2 + 50x_1 - 7x_0 - 28y_3 - 21y_2 - 8y_1 - 6y_0 + 3c \geq -35 \\
13x_3 + 17x_2 - 3x_1 - 30x_0 + 4y_3 - 14y_2 - 5y_1 + 20y_0 - 15c \geq -34 \\
47x_3 - 42x_2 - 6x_1 - 17x_0 + 12y_3 - 4y_2 + 14y_1 - 50y_0 - 11c \geq -65 \\
42x_3 - 32x_2 + 3x_1 + 12x_0 - 20y_3 - 43y_2 + 7y_1 - 4y_0 - 5c \geq -52 \\
-9x_3 - 4x_2 + 11x_1 + 8x_0 + 11y_3 + 2y_2 - 12y_1 + y_0 + 5c \geq -12 \\
-50x_3 + 8x_2 - 11x_1 - 9x_0 - 10y_3 + 3y_2 + 3y_1 + 4y_0 + 2c \geq -67 \\
31x_3 - 10x_2 + 4x_1 + 4x_0 - 26y_3 + 18y_2 - 16y_1 - 20y_0 + c \geq -36 \\
42x_3 - 7x_2 - 16x_1 - 4x_0 - 32y_3 + 6y_2 + 13y_1 - 12y_0 - 47c \geq -59 \\
37x_3 - 13x_2 - 28x_1 - 5x_0 + 8y_3 - 23y_2 - 19y_1 + 21y_0 - 4c \geq -46 \\
31x_3 - 33x_2 - 9x_1 + 7x_0 - 2y_3 - 4y_2 - 28y_1 - 8y_0 + 9c \geq -44 \\
19x_3 - x_2 + 6x_1 - 24x_0 + 3y_3 - 5y_2 + 6y_1 + 3y_0 - 20c \geq -25 \\
17x_3 - 7x_2 - 3x_1 - 15x_0 + 18y_3 - 5y_2 + 8y_1 - 20y_0 + c \geq -25 \\
43x_3 - 40x_2 + 10x_1 - 4x_0 - 6y_3 + 7y_2 - 50y_1 + 5y_0 + 2c \geq -50 \\
22x_3 + 2x_2 - 24x_1 - 5x_0 + 6y_3 + 2y_2 - 6y_1 + 7y_0 - 23c \geq -29 \\
-50x_3 - 46x_2 - 48x_1 + x_0 + 48y_3 + 49y_2 + 47y_1 + 2y_0 + 49c \geq -94 \\
-49x_3 + 49x_2 + 4x_1 + 2x_0 - y_3 - 48y_2 + 47y_1 - 50y_0 + 46c \geq -98
\end{cases}
\tag{B7}
$$

- Encoding the signed DLCT of S_4 with $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, c, s)$, where $(x_0, x_1, x_2, x_3) \xrightarrow{S_4} (y_0, y_1, y_2, y_3)$ denotes the valid DL approximation with signed correlation $(-1)^s \cdot 2^{-c}$. This signed DLCT is characterized as:

$$
\begin{cases}
-12x_3 - 7x_2 + 8x_1 + 3x_0 - 5y_3 - y_2 + 3y_1 - 13y_0 + 6c + 12s \geq -19 \\
-4x_3 + x_2 - 24x_1 - 2x_0 + 20y_3 + 23y_2 + 22y_1 - 8y_0 - 8c - 8s \geq -30 \\
-4x_3 - 18x_2 + 2x_1 + x_0 + 10y_3 + 24y_2 + 15y_1 + 29y_0 - 33c + 7s \geq -22 \\
4x_3 + 4x_2 + 20x_1 - 19x_0 - 8y_3 + 6y_2 - 20y_1 + 10y_0 - c + 12s \geq -28 \\
4x_3 - 2x_2 - 2x_1 - 9x_0 - 15y_3 - 19y_2 - 9y_1 + 17y_0 - 18c - s \geq -56 \\
8x_3 + 6x_2 + 4x_1 + 18x_0 - 9y_2 - 3y_1 - 7y_0 - 18c + 10s \geq -19 \\
-5x_3 - 5x_2 + 27x_1 - 7x_0 + 15y_3 - 9y_2 + 20y_1 + 18y_0 - c - 17s \geq -17 \\
-7x_3 + 2x_2 + 5x_1 - 3x_0 + 5y_3 - 7y_2 + 7y_1 - y_0 + 4c - 3s \geq -10 \\
16x_3 + 3x_2 + 4x_1 - 3x_0 - 5y_3 + 10y_2 + 11y_1 + 12y_0 + 3c - 16s \geq -5 \\
-10x_3 + 20x_2 + 16x_1 + 12x_0 - 18y_3 + 6y_2 - 30y_1 + 8y_0 - c - 19s \geq -48 \\
5x_3 - 2x_2 - 2x_1 - x_0 - 3y_3 - 6y_2 + 5y_1 - 3y_0 + 3c - 4s \geq -12 \\
-18x_3 - 6x_2 + 2x_1 - 8x_0 + 2y_3 - 32y_2 + 10y_1 + 27c + 3s \geq -32 \\
-4x_3 - 5x_2 - 4x_1 + x_0 + 21y_3 + 21y_2 - 13y_1 + 2y_0 + 45c - 38s \geq -13 \\
-4x_3 + 13x_2 - 16x_1 - 7x_0 + 6y_3 - 8y_2 - 20y_1 - 2y_0 + 5c + 10s \geq -32 \\
2x_3 + x_2 + x_1 + 9x_0 + 7y_3 + 13y_2 + 6y_1 - 13y_0 + 31c - 40s \geq -13 \\
-7x_3 - 3x_2 + 8x_1 - 4x_0 - 6y_3 + 5y_2 - 6y_1 - 10y_0 + 6c - 2s \geq -22 \\
12x_3 - 2x_2 - 12x_1 + 9x_0 - 2y_3 + 12y_2 - 3y_1 - 4y_0 + c - 4s \geq -14 \\
3x_3 + x_2 + 2x_1 + y_3 - 2y_2 - 3y_1 - 3y_0 + 2c - 5s \geq -8 \\
-44x_3 - 26x_2 + x_1 + 4x_0 + 4y_3 - 10y_2 - 6y_1 - 54y_0 + 97c - 36s \geq -70 \\
15x_3 - 17x_2 - 4x_1 + 5x_0 + 9y_3 - 4y_2 - 17y_1 + 17y_0 - c - 2s \geq -28 \\
-2x_3 + 14x_2 - 25x_1 - 10x_0 + 2y_3 - 21y_2 + 3y_1 + 27y_0 - 6c + 13s \geq -37 \\
-4x_3 - 27x_2 + 19x_1 + 14x_0 + y_3 + 14y_2 - 29y_1 - 6y_0 - 2c + 35s \geq -35 \\
-6x_3 + 22x_2 - 23x_1 + 24x_0 - 8y_3 - 20y_2 + 10y_1 - 4y_0 - c + 22s \geq -32 \\
-28x_3 - 13x_2 - 4x_1 - 4x_0 - 49y_3 + 14y_2 + 7y_1 + 5y_0 + 37c + s \geq -49 \\
7x_3 + x_2 + 2x_1 - 39x_0 + 3y_3 + 8y_2 + 5y_1 - 39y_0 + 27c + 12s \geq -39
\end{cases}
\tag{B8}
$$

- Encoding the DDT of S_5 with $(x_0, x_1, x_2, x_3, x_4, y_0, y_1, y_2, y_3, y_4, p_0, p_1, p_2)$, where $(x_0, x_1, x_2, x_3, x_4) \xrightarrow{S_5} (y_0, y_1, y_2, y_3, y_4)$ denotes the valid differential transition with probability $2^{-(2p_0+3p_1+4p_2)}$. The DDT is characterized as:

$$
\begin{cases}
-39x_4 - 50x_3 - 21x_2 + 5x_1 - 23x_0 - y_4 - 9y_3 - 9y_2 - 2y_1 + 2y_0 - 156p_1 - 112p_2 \geq -260 \\
67x_4 - 42x_3 + 42x_2 - 38x_1 + 20x_0 + 68y_4 - 68y_3 + y_2 - 10y_1 + 10y_0 + 2p_0 - 24p_1 + 2p_2 \geq -114 \\
-14x_4 - 7x_2 + x_0 - 2y_4 + y_3 - 7y_2 + 2y_0 - 33p_0 - 21p_1 - 19p_2 \geq -49 \\
x_4 - 11x_3 - 15x_2 - 12x_1 - 7x_0 + y_4 + y_3 + 2y_2 - 2y_1 - 2y_0 + 4p_0 + 26p_1 + 37p_2 \geq -10 \\
-x_4 - 2x_3 - x_2 - 4x_1 - 3x_0 - 3y_4 - 3y_3 + 3y_2 - 3y_1 - 3y_0 + 2p_2 \geq -18 \\
-24x_4 - 36x_3 - 39x_2 - 36x_1 - 43x_0 - 13y_4 + y_3 + 4y_2 - 11y_1 + 6y_0 - 213p_0 - 123p_1 - 60p_2 \geq -261 \\
-33x_4 - 28x_3 - 35x_2 - 45x_1 - 52x_0 - 16y_4 - 5y_3 + 7y_2 + y_1 + 3y_0 - 225p_0 - 129p_1 - 61p_2 \geq -274 \\
-40x_4 - 41x_3 - 17x_2 + x_1 - 17x_0 + 2y_4 - 3y_3 - 9y_2 + 8y_1 + 2y_0 - 138p_1 - 98p_2 \geq -222 \\
94x_4 - 8x_3 - 3x_2 - 35x_1 - 10x_0 - y_4 + 2y_3 - 3y_2 + 95y_1 + 95y_0 + 7p_0 + 25p_1 - 35p_2 \geq 0 \\
-23x_4 - 19x_3 - 38x_2 - 42x_1 - 34x_0 + 7y_4 - y_3 + 2y_2 + 2y_1 - 7y_0 - 175p_1 - 114p_2 \geq -277 \\
-x_4 - 4x_3 - 3x_2 - x_1 - 2x_0 + 3y_4 - 3y_3 - 3y_2 - 3y_1 - 3y_0 + 2p_2 \geq -18 \\
-143x_4 - 119x_3 - 120x_2 - 116x_1 - 146x_0 + 8y_4 - 21y_3 + 17y_2 - 3y_1 + 9y_0 + 55p_0 + 323p_1 + 580p_2 \geq -85 \\
92x_4 + 231x_3 - 2x_2 + 30x_1 - 183x_0 + 99y_4 - 40y_3 + 41y_2 + 18y_1 + 199y_0 + p_0 - 6p_1 + 4p_2 \geq 0 \\
-25x_4 + x_3 - 19x_2 - 25x_1 + 4x_0 + 6y_4 + 5y_3 - 25y_2 - 25y_1 + 4y_0 - 23p_0 - 4p_1 + 13p_2 \geq -98 \\
-3x_4 + 20x_3 - 3x_2 - 5x_1 + 20x_0 - 4y_4 - 2y_3 + y_2 - y_1 - 40p_0 - 50p_2 \geq -48 \\
-2x_4 + 17x_3 - 64x_2 + 17x_1 + 26x_0 + 3y_4 + 60y_2 + 6y_1 - 5y_0 - 202p_0 - 161p_1 - 175p_2 \geq -206 \\
-90x_4 - 115x_3 - 103x_2 + x_1 - 58x_0 - y_4 + 24y_3 + 21y_2 + 5y_1 - 5y_0 - 581p_0 - 422p_1 - 308p_2 \geq -678 \\
-40x_4 + 46x_3 + 97x_2 - 54x_1 + 60x_0 - 14y_4 + 12y_3 + 98y_2 - 98y_1 + y_0 + 8p_0 - 36p_1 + 2p_2 \geq -144 \\
68x_4 + 112x_3 - 47x_2 + 63x_1 - 44x_0 - 3y_4 + 112y_3 - 112y_2 - y_1 - 5y_0 + 24p_0 - 38p_1 + 11p_2 \geq -138 \\
28x_4 - x_3 + 24x_2 - 35x_1 - 31x_0 - 58y_4 + 57y_3 - 2y_1 - 3y_0 - 51p_0 - 17p_1 + 17p_2 \geq -88 \\
-58x_4 + 4x_3 - 69x_2 - 126x_1 - 139x_0 - 17y_4 + 8y_3 - 7y_2 - 5y_1 - 11y_0 - 102p_1 + 33p_2 \geq -390 \\
-16x_4 - 16x_3 + 5x_2 - 8x_1 + x_0 - 16y_4 - 16y_3 - 8y_2 - y_1 - 22p_0 - 5p_1 + 5p_2 \geq -70 \\
-24x_4 - 54x_3 - 60x_2 - 29x_1 + 5x_0 - y_4 - 3y_3 - 7y_2 - 8y_1 - 98p_0 - 44p_1 + 12p_2 \geq -169 \\
-4x_4 - 6x_3 - x_2 + x_1 + 2y_4 + 4y_3 - 15p_0 - 10p_1 \geq -17 \\
-4x_3 - 6x_2 + x_0 + 2y_3 + 4y_2 - 15p_0 - 10p_1 \geq -17 \\
-20x_4 - 24x_3 - 12x_2 + x_1 - 15x_0 + 4y_3 + 2y_2 - 48p_0 - 19p_1 + 2p_2 \geq -68 \\
-8x_4 - 8x_3 + 26x_2 + 13x_1 - x_0 + 13y_4 + 26y_3 - y_2 + 2p_0 - 3p_1 - 8p_2 \geq 0 \\
134x_4 - 2x_3 + 67x_2 + 67x_1 - 4x_0 - 4y_4 - 6y_3 - 11y_1 + 7y_0 + 12p_0 + 20p_1 - 107p_2 \geq 0 \\
44x_4 + 44x_3 - x_2 + 66x_1 - 4x_0 - y_4 - 4y_3 + 3y_2 - 5y_1 + 22y_0 + 3p_0 - 73p_2 \geq 0 \\
13x_4 + 18x_3 + 24x_2 + 9x_1 + 26x_0 + 4y_4 - y_2 + y_1 - 14y_0 - 3p_0 - 37p_2 \geq 0 \\
-219x_4 + 32x_3 + 62x_2 + 2x_1 + 21x_0 + 225y_4 + 34y_3 + 3y_2 + 4y_1 + 2y_0 - 117p_0 + 43p_1 + 21p_2 \geq -111 \\
59x_4 + 4x_3 + 17x_2 - 209x_1 + 31x_0 + y_4 + 4y_3 + 2y_2 + 217y_1 + 34y_0 - 114p_0 + 43p_1 + 21p_2 \geq -106 \\
3x_4 + x_3 - 209x_2 - 130x_1 + 28x_0 + 2y_4 + y_3 + 130y_2 + 51y_1 + 5y_0 - 199p_0 + 10p_1 + 28p_2 \geq -278 \\
-47x_4 - 43x_3 - 43x_1 - 35x_0 + 2y_4 + y_3 - 10y_2 + 4y_1 - 4y_0 - 111p_0 + 21p_2 \geq -160 \\
-34x_4 + x_3 - 34x_2 - 41x_1 - 41x_0 - 8y_4 - 4y_3 - y_2 + y_1 - 5y_0 - 100p_0 - 20p_1 + 19p_2 \geq -147 \\
x_4 - 8x_3 - 15x_2 - 16x_1 - 11x_0 - y_1 - 2y_0 + 4p_0 + 24p_1 + 39p_2 \geq -13 \\
2x_4 - 28x_3 + 9x_2 + 3x_1 + 9x_0 - 36y_2 + 5y_1 - 23p_0 + 5p_1 + 2p_2 \geq -51 \\
-35x_4 - 64x_3 + x_2 + 2x_1 + 13y_4 + 37y_3 + 2y_2 - 5y_1 - 58p_0 + 2p_1 + 16p_2 \geq -85 \\
-70x_4 + x_3 - 126x_2 - 112x_1 + x_0 - 42y_4 + 28y_2 - 28y_1 + 3y_0 - 232p_0 + 3p_1 + 5p_2 \geq -372 \\
-31x_3 - 29x_2 - 14x_1 - y_4 + 4y_3 + 2y_2 - 14y_1 - 53p_0 + 8p_2 \geq -81 \\
-53x_4 + 3x_3 - 9x_2 + x_1 - 54x_0 - 5y_4 - 59y_3 + 59y_2 + 9y_1 - 8y_0 - 84p_0 - 29p_1 + 32p_2 \geq -153 \\
x_4 - 28x_3 - 15x_2 + 15x_1 - 26x_0 - 2y_3 + 4y_2 + 30y_1 - 30y_0 - 18p_0 - 7p_1 + 22p_2 \geq -78 \\
-66x_4 - 29x_3 + 3x_2 - 30x_1 - 61x_0 - 5y_4 - 7y_3 + 3y_2 - y_0 - 107p_0 - 46p_1 + 17p_2 \geq -179 \\
12x_4 + 2x_3 - 74x_2 - 75x_1 - 60x_0 - 2y_4 - y_3 + 4y_2 + 8y_1 - 8y_0 - 109p_0 - 38p_1 + 38p_2 \geq -182 \\
-37x_4 + 10x_3 - 45x_2 - 84x_1 - 91x_0 - 10y_4 - 2y_3 + 2y_2 + y_1 - 6y_0 - 149p_0 - 63p_1 + 23p_2 \geq -246 \\
13x_4 + 16x_3 + 18x_2 + 16x_1 + 18x_0 + 3y_4 - 6y_2 + 2y_1 - 4y_0 - 8p_0 - 24p_1 - 24p_2 \geq 0 \\
-3x_4 + 20x_3 + 77x_2 + 47x_1 + 68x_0 + 23y_4 + 3y_3 - 62y_2 + 14y_1 - 7y_0 + p_0 - 12p_1 - 19p_2 \geq 0
\end{cases}
\tag{B9}
$$

- Encoding the LAT of S_5 with $(x_0, x_1, x_2, x_3, x_4, y_0, y_1, y_2, y_3, y_4, c_0, c_1)$, where $(x_0, x_1, x_2, x_3, x_4) \xrightarrow{S_5} (y_0, y_1, y_2, y_3, y_4)$ denotes the valid linear approximation with correlation $2^{-(c_0+c_1)}$. The LAT is characterized as:

$$\begin{cases} -x_4 - 3x_3 - 3x_2 - x_1 + 3x_0 + 2y_4 - 3y_3 - 3y_2 - 4y_1 - 3y_0 + 6c_0 + 6c_1 \geq -6 \\ x_4 - 3x_3 + x_2 + 5x_1 - 3x_0 + 2y_2 - 4y_1 - 3y_0 + 5c_0 + c_1 \geq -4 \\ -x_4 + 3x_3 - x_2 - 3x_1 - 3x_0 - 3y_4 - 3y_3 + 2y_2 - 4y_1 - 3y_0 + 6c_0 + 6c_1 \geq -6 \\ 5x_4 + 2x_3 - 46x_2 - 5x_1 - 2x_0 + 23y_4 + 30y_3 + 41y_2 - 12y_1 + y_0 + 19c_0 + 23c_1 \geq 0 \\ 10x_4 + x_3 + 2x_1 - 5y_4 + y_3 + 8y_1 + 10y_0 - 5c_1 \geq 0 \\ -x_4 - 5x_3 - x_1 + 9y_4 + 10y_3 - y_2 - y_1 + 10y_0 + 4c_0 - 5c_1 \geq 0 \\ -3x_4 + x_3 - 3x_2 + x_1 + 5x_0 - 3y_4 + 2y_1 - 4y_0 + 5c_0 + c_1 \geq -4 \\ -3x_4 + x_3 + 3x_2 + x_1 - 3x_0 - 3y_4 - 3y_3 - 3y_2 + 2y_1 - 3y_0 + 5c_0 + 5c_1 \geq -5 \\ x_4 + 2x_2 + 10x_0 + y_4 + 8y_2 + 10y_1 - 5y_0 - 5c_1 \geq 0 \\ -4x_3 + 2x_2 - 4x_0 + 4y_4 - y_3 + 2y_2 - 4y_1 - 4y_0 + 8c_0 - 3c_1 \geq -8 \\ 3x_4 + x_3 - 3x_2 - 3x_1 + x_0 - 3y_4 + 2y_3 - 3y_2 - 3y_1 - 3y_0 + 5c_0 + 5c_1 \geq -5 \\ -9x_4 - x_3 - 9x_2 + 9x_1 - x_0 - 9y_4 + 8y_3 - 3y_2 - 3y_1 - 9y_0 + 19c_0 - 3c_1 \geq -19 \\ 2x_3 + 10x_1 + x_0 + 8y_3 + 10y_2 - 5y_1 + y_0 - 5c_1 \geq 0 \\ 10x_3 + x_2 + 2x_0 + 10y_4 - 5y_3 + y_2 + 8y_0 - 5c_1 \geq 0 \\ -2x_4 + 7x_3 + 82x_2 - 4x_1 + 24x_0 + 82y_4 + 58y_3 - 17y_2 + 17y_1 - 6y_0 - 4c_0 - 49c_1 \geq 0 \\ -20x_4 - 2x_3 + 2x_1 + x_0 + 18y_4 - 5y_3 + 11y_1 + 14y_0 + 7c_0 + 9c_1 \geq 0 \\ -x_4 - x_2 - 5x_1 - y_4 + 10y_3 + 9y_2 + 10y_1 - y_0 + 4c_0 - 5c_1 \geq 0 \\ 2x_4 - 4x_2 - 4x_0 + 2y_4 - 4y_3 - 4y_2 + 4y_1 - 3y_0 + 8c_0 - c_1 \geq -8 \\ 4x_4 + x_3 + 34x_2 + 2x_1 - 9x_0 + 15y_4 + 24y_3 - 92y_2 + 2y_1 - 58y_0 + 67c_0 + 73c_1 \geq 0 \\ x_4 + 25x_3 - 9x_2 + 2x_1 - 12x_0 + 3y_4 - 76y_3 - 30y_2 - 46y_1 - 58y_0 + 102c_0 + 125c_1 \geq 0 \\ -13x_4 - 2x_3 + 22x_2 - 13x_1 - x_0 - 47y_4 + 3y_3 - 54y_2 - 16y_1 - 25y_0 + 75c_0 + 92c_1 \geq 0 \\ -8x_4 - 8x_3 + x_2 + 3x_1 + x_0 - 8y_4 - 8y_3 - 8y_2 - 8y_1 + 2y_0 + 53c_0 - 8c_1 \geq 0 \\ -18x_4 + 8x_3 + x_2 - 18x_1 + x_0 - 5y_4 - 5y_3 - 18y_2 - 18y_1 + 7y_0 + 92c_0 - 18c_1 \geq 0 \\ 16x_4 - 29x_3 - 4x_2 - 29x_1 - 6x_0 - 29y_4 - 29y_3 - 11y_2 - 9y_1 + 12y_0 + 139c_0 - 9c_1 \geq 0 \\ -x_3 - x_1 - 5x_0 - y_4 - y_3 + 10y_2 + 9y_1 + 10y_0 + 4c_0 - 5c_1 \geq 0 \\ 3x_4 + x_3 + 3x_2 + 2x_1 + 2x_0 - 23y_4 + y_3 - 24y_2 + y_1 - 24y_0 + 57c_0 + 11c_1 \geq 0 \\ x_4 - 18x_3 - 6x_2 + 4x_1 + 10x_0 - 22y_4 - 19y_3 - 23y_2 - 31y_1 - 25y_0 + 60c_0 + 83c_1 \geq 0 \\ 3x_4 + 6x_3 + 4x_2 - 14x_1 - 6x_0 - 59y_4 - 41y_3 - 54y_2 - 38y_1 - 40y_0 + 105c_0 + 144c_1 \geq 0 \\ x_4 - 18x_3 + x_2 - 18x_1 + 8x_0 - 18y_4 - 18y_3 + 7y_2 - 5y_1 - 5y_0 + 92c_0 - 18c_1 \geq 0 \end{cases} \tag{B10}$$

- Encoding the unsigned DLCT of S_5 with $(x_0, x_1, x_2, x_3, x_4, y_0, y_1, y_2, y_3, y_4)$, where $(x_0, x_1, x_2, x_3, x_4) \xrightarrow{S_5} (y_0, y_1, y_2, y_3, y_4)$ denotes the valid DL approximation with constant correlation 1. The unsigned DLCT is characterized as:

$$\begin{cases} 7x_4 + x_3 + 11x_2 - 17x_1 - 25x_0 + y_4 + 11y_3 - 39y_2 + 7y_1 + y_0 \geq -42 \\ 7x_4 - 35x_3 - 28x_2 + 15x_1 + x_0 - 63y_4 + 17y_3 - 3y_2 + y_1 + 22y_0 \geq -66 \\ -9x_4 - 2x_3 + 5x_1 - 2y_4 + y_3 + 5y_2 - 9y_1 \geq -11 \\ 3x_3 + x_1 - 8x_0 - 2y_4 + y_3 - 8y_1 + 3y_0 \geq -10 \\ x_4 - 26x_2 - 19x_1 + 10x_0 + 17y_4 - 43y_3 + 12y_2 - 2y_1 + 3y_0 \geq -47 \\ -8x_4 + 3x_2 + x_0 + 3y_4 - 2y_3 + y_2 - 8y_0 \geq -10 \\ 21x_4 - 2x_3 - 7x_2 - 34x_1 - 6y_4 - 2y_3 - 32y_2 + 21y_1 - 3y_0 \geq -44 \\ -x_4 - 59x_3 - 7x_2 - 7x_1 + 37x_0 - 3y_4 - 8y_3 - 10y_2 + 37y_1 - 57y_0 \geq -78 \\ 2x_4 + 30x_3 - 22x_2 - 22x_1 - 6x_0 - 3y_4 - 41y_3 + 18y_2 - 8y_1 \geq -52 \end{cases} \tag{B11}$$

- Encoding the signed DLCT of S_5 with $(x_0, x_1, x_2, x_3, x_4, y_0, y_1, y_2, y_3, y_4, s)$, where $s = 0$ indicates a positive correlation and $s = 1$ indicates a negative correlation. The signed DLCT is characterized as:

$$\begin{cases} 21x_4 + 42x_3 + 34x_2 + x_1 + 5x_0 + 21y_4 - 8y_3 - 32y_2 + y_1 + 5y_0 - 42s \geq -40 \\ -10x_4 + x_3 - 42x_1 - 17x_0 + 13y_4 + 11y_3 - 52y_2 + 37y_1 - 8y_0 - 2s \geq -69 \\ -14x_3 + x_2 - 5x_1 - 12y_4 + 14y_3 - 4y_2 + 5y_1 - 2s \geq -19 \\ -2x_4 - 20x_2 + 18x_0 + y_4 - 14y_3 - 7y_2 - 4y_0 + 7s \geq -25 \\ 3x_3 - 3x_0 - 2y_1 - y_0 + s \geq -3 \\ -3x_4 + 3x_2 - y_4 - 2y_0 + s \geq -3 \\ 2x_4 + 4x_3 - 84x_2 + x_1 + 51x_0 + 2y_4 - 92y_3 + 51y_2 + 9y_1 - 14y_0 - 18s \geq -106 \\ -x_4 - 6x_3 + 5x_1 - 4y_4 - 2y_3 - y_1 + 2s \geq -7 \\ 21x_4 - 36x_2 + x_1 + 2x_0 - 40y_4 + 4y_2 + 7y_1 + 21y_0 - 8s \geq -40 \\ x_4 - 3x_3 + 27x_2 + 5x_1 - 111x_0 + 3y_4 + 27y_3 - 47y_2 - 69y_1 + 71y_0 - 12s \geq -119 \\ x_4 + 23x_3 - 39x_1 + 4x_0 + 23y_4 - 41y_3 + 6y_1 + 4y_0 - 8s \geq -41 \\ -33x_4 - x_3 + 2x_2 + 19x_1 + 3y_4 + 7y_3 + 19y_2 - 36y_1 - 7s \geq -37 \\ -36x_3 + x_2 + 2x_1 + 21x_0 + 4y_3 + 7y_2 + 21y_1 - 40y_0 - 8s \geq -40 \\ 3x_4 - 3x_1 - 2y_2 - y_1 + s \geq -3 \end{cases} \tag{B12}$$

To exclude a specific feasible solution $(x_0, x_1, \ldots, x_{m-1}) = (\delta_0, \delta_1, \ldots, \delta_{m-1})$ from the solution space, we add the following *cutting-off inequality* to the MILP model:

$$\sum_{i=0}^{m-1} (-1)^{\delta_i} x_i + \sum_{i=0}^{m-1} \delta_i \geq 1. \tag{B13}$$

This inequality ensures that the current solution is removed from the feasible region while keeping all other valid solutions, thus allowing the MILP solver to discover new solutions in subsequent iterations.

Appendix C Newly Identified Linear-layer Parameter Sets

Table C11: New linear-layer parameters grouped by category.

Category	**Parameters** (p, t_0, t_1, t_2)
Improved Class A	(35,0,42,85); (35,1,44,86); (35,2,45,88); (35,4,46,89); (35,5,48,90); (35,6,49,92); (35,8,50,93); (35,9,52,94); (35,10,53,96); (35,12,54,97); (35,13,56,98); (35,14,57,100); (35,16,58,101); (35,17,60,102); (35,18,61,104); (35,20,62,105)
	(35,21,64,106); (35,22,65,108); (35,24,66,109); (35,25,68,110); (35,26,69,112); (35,28,70,113); (35,29,72,114); (35,30,73,116); (35,32,74,117); (35,33,76,118); (35,34,77,120); (35,36,78,121); (35,37,80,122); (35,38,81,124); (35,40,82,125); (35,41,84,126)
	(93,1,43,86); (93,2,45,87); (93,3,46,89); (93,5,47,90); (93,6,49,91); (93,7,50,93); (93,9,51,94); (93,10,53,95); (93,11,54,97); (93,13,55,98); (93,14,57,99); (93,15,58,101); (93,17,59,102); (93,18,61,103); (93,19,62,105); (93,21,63,106)
	(93,22,65,107); (93,23,66,109); (93,25,67,110); (93,26,69,111); (93,27,70,113); (93,29,71,114); (93,30,73,115); (93,31,74,117); (93,33,75,118); (93,34,77,119); (93,35,78,121); (93,37,79,122); (93,38,81,123); (93,39,82,125); (93,41,83,126); (93,42,85,127)

Continued on next page

Category	Parameters (p, t_0, t_1, t_2)
Improved Class B	(29,1,43,86); (29,2,45,87); (29,3,46,89); (29,5,47,90); (29,6,49,91); (29,7,50,93); (29,9,51,94); (29,10,53,95); (29,11,54,97); (29,13,55,98); (29,14,57,99); (29,15,58,101); (29,17,59,102); (29,18,61,103); (29,19,62,105); (29,21,63,106)
	(29,22,65,107); (29,23,66,109); (29,25,67,110); (29,26,69,111); (29,27,70,113); (29,29,71,114); (29,30,73,115); (29,31,74,117); (29,33,75,118); (29,34,77,119); (29,35,78,121); (29,37,79,122); (29,38,81,123); (29,39,82,125); (29,41,83,126); (29,42,85,127)
	(99,0,42,85); (99,1,44,86); (99,2,45,88); (99,4,46,89); (99,5,48,90); (99,6,49,92); (99,8,50,93); (99,9,52,94); (99,10,53,96); (99,12,54,97); (99,13,56,98); (99,14,57,100); (99,16,58,101); (99,17,60,102); (99,18,61,104); (99,20,62,105)
	(99,21,64,106); (99,22,65,108); (99,24,66,109); (99,25,68,110); (99,26,69,112); (99,28,70,113); (99,29,72,114); (99,30,73,116); (99,32,74,117); (99,33,76,118); (99,34,77,120); (99,36,78,121); (99,37,80,122); (99,38,81,124); (99,40,82,125); (99,41,84,126)
Improved Class C	(3,3,31,59); (3,3,31,103); (3,3,75,103); (3,7,35,63); (3,7,35,107); (3,7,79,107); (3,11,39,67); (3,11,39,111); (3,11,83,111); (3,15,43,71); (3,15,43,115); (3,15,87,115); (3,19,47,75); (3,19,47,119); (3,19,91,119); (3,23,51,79)
	(3,23,51,123); (3,23,95,123); (3,27,55,83); (3,27,55,127); (3,27,99,127); (3,31,59,87); (3,35,63,91); (3,39,67,95); (3,43,71,99); (3,47,75,103); (3,51,79,107); (3,55,83,111); (3,59,87,115); (3,63,91,119); (3,67,95,123); (3,71,99,127)
	(29,0,8,68); (29,0,60,68); (29,0,60,120); (29,4,12,72); (29,4,64,72); (29,4,64,124); (29,8,16,76); (29,8,68,76); (29,12,20,80); (29,12,72,80); (29,16,24,84); (29,16,76,84); (29,20,28,88); (29,20,80,88); (29,24,32,92); (29,24,84,92)
	(29,28,36,96); (29,28,88,96); (29,32,40,100); (29,32,92,100); (29,36,44,104); (29,36,96,104); (29,40,48,108); (29,40,100,108); (29,44,52,112); (29,44,104,112); (29,48,56,116); (29,48,108,116); (29,52,60,120); (29,52,112,120); (29,56,64,124); (29,56,116,124)
	(35,3,7,11); (35,3,7,127); (35,3,123,127); (35,7,11,15); (35,11,15,19); (35,15,19,23); (35,19,23,27); (35,23,27,31); (35,27,31,35); (35,31,35,39); (35,35,39,43); (35,39,43,47); (35,43,47,51); (35,47,51,55); (35,51,55,59); (35,55,59,63)
	(35,59,63,67); (35,63,67,71); (35,67,71,75); (35,71,75,79); (35,75,79,83); (35,79,83,87); (35,83,87,91); (35,87,91,95); (35,91,95,99); (35,95,99,103); (35,99,103,107); (35,103,107,111); (35,107,111,115); (35,111,115,119); (35,115,119,123); (35,119,123,127)
	(61,0,36,72); (61,0,36,92); (61,0,56,92); (61,4,40,76); (61,4,40,96); (61,4,60,96); (61,8,44,80); (61,8,44,100); (61,8,64,100); (61,12,48,84); (61,12,48,104); (61,12,68,104); (61,16,52,88); (61,16,52,108); (61,16,72,108); (61,20,56,92)
	(61,20,56,112); (61,20,76,112); (61,24,60,96); (61,24,60,116); (61,24,80,116); (61,28,64,100); (61,28,64,120); (61,28,84,120); (61,32,68,104); (61,32,68,124); (61,32,88,124); (61,36,72,108); (61,40,76,112); (61,44,80,116); (61,48,84,120); (61,52,88,124)
	(67,3,39,75); (67,3,39,95); (67,3,59,95); (67,7,43,79); (67,7,43,99); (67,7,63,99); (67,11,47,83); (67,11,47,103); (67,11,67,103); (67,15,51,87); (67,15,51,107); (67,15,71,107); (67,19,55,91); (67,19,55,111); (67,19,75,111); (67,23,59,95)
	(67,23,59,115); (67,23,79,115); (67,27,63,99); (67,27,63,119); (67,27,83,119); (67,31,67,103); (67,31,67,123); (67,31,87,123); (67,35,71,107); (67,35,71,127); (67,35,91,127); (67,39,75,111); (67,43,79,115); (67,47,83,119); (67,51,87,123); (67,55,91,127)
	(93,0,4,8); (93,0,4,124); (93,0,120,124); (93,4,8,12); (93,8,12,16); (93,12,16,20); (93,16,20,24); (93,20,24,28); (93,24,28,32); (93,28,32,36); (93,32,36,40); (93,36,40,44); (93,40,44,48); (93,44,48,52); (93,48,52,56); (93,52,56,60)
	(93,56,60,64); (93,60,64,68); (93,64,68,72); (93,68,72,76); (93,72,76,80); (93,76,80,84); (93,80,84,88); (93,84,88,92); (93,88,92,96); (93,92,96,100); (93,96,100,104); (93,100,104,108); (93,104,108,112); (93,108,112,116); (93,112,116,120); (93,116,120,124)
	(99,3,11,71); (99,3,63,71); (99,3,63,123); (99,7,15,75); (99,7,67,75); (99,7,67,127); (99,11,19,79); (99,11,71,79); (99,15,23,83); (99,15,75,83); (99,19,27,87); (99,19,79,87); (99,23,31,91); (99,23,83,91); (99,27,35,95); (99,27,87,95)
	(99,31,39,99); (99,31,91,99); (99,35,43,103); (99,35,95,103); (99,39,47,107); (99,39,99,107); (99,43,51,111); (99,43,103,111); (99,47,55,115); (99,47,107,115); (99,51,59,119); (99,51,111,119); (99,55,63,123); (99,55,115,123); (99,59,67,127); (99,59,119,127)
	(125,0,28,56); (125,0,28,100); (125,0,72,100); (125,4,32,60); (125,4,32,104); (125,4,76,104); (125,8,36,64); (125,8,36,108); (125,8,80,108); (125,12,40,68); (125,12,40,112); (125,12,84,112); (125,16,44,72); (125,16,44,116); (125,16,88,116); (125,20,48,76)
	(125,20,48,120); (125,20,92,120); (125,24,52,80); (125,24,52,124); (125,24,96,124); (125,28,56,84); (125,32,60,88); (125,36,64,92); (125,40,68,96); (125,44,72,100); (125,48,76,104); (125,52,80,108); (125,56,84,112); (125,60,88,116); (125,64,92,120); (125,68,96,124)
Improved Class D	(61,2,10,18); (61,2,10,122); (61,2,114,122); (61,6,14,22); (61,6,14,126); (61,6,118,126); (61,10,18,26); (61,14,22,30); (61,18,26,34); (61,22,30,38); (61,26,34,42); (61,30,38,46); (61,34,42,50); (61,38,46,54); (61,42,50,58); (61,46,54,62)

Continued on next page

Category	Parameters (p, t_0, t_1, t_2)
	(61,50,58,66); (61,54,62,70); (61,58,66,74); (61,62,70,78); (61,66,74,82); (61,70,78,86); (61,74,82,90); (61,78,86,94); (61,82,90,98); (61,86,94,102); (61,90,98,106); (61,94,102,110); (61,98,106,114); (61,102,110,118); (61,106,114,122); (61,110,118,126)
	(67,1,9,17); (67,1,9,121); (67,1,113,121); (67,5,13,21); (67,5,13,125); (67,5,117,125); (67,9,17,25); (67,13,21,29); (67,17,25,33); (67,21,29,37); (67,25,33,41); (67,29,37,45); (67,33,41,49); (67,37,45,53); (67,41,49,57); (67,45,53,61)
	(67,49,57,65); (67,53,61,69); (67,57,65,73); (67,61,69,77); (67,65,73,81); (67,69,77,85); (67,73,81,89); (67,77,85,93); (67,81,89,97); (67,85,93,101); (67,89,97,105); (67,93,101,109); (67,97,105,113); (67,101,109,117); (67,105,113,121); (67,109,117,125)
Improved Class E	(3,3,19,75); (3,3,59,75); (3,3,59,115); (3,7,23,79); (3,7,63,79); (3,7,63,119); (3,11,27,83); (3,11,67,83); (3,11,67,123); (3,15,31,87); (3,15,71,87); (3,15,71,127); (3,19,35,91); (3,19,75,91); (3,23,39,95); (3,23,79,95); (3,27,43,99)
	(3,27,83,99); (3,31,47,103); (3,31,87,103); (3,35,51,107); (3,35,91,107); (3,39,55,111); (3,39,95,111); (3,43,59,115); (3,43,99,115); (3,47,63,119); (3,47,103,119); (3,51,67,123); (3,51,107,123); (3,55,71,127); (3,55,111,127); (13,0,40,80)
	(13,0,40,88); (13,0,48,88); (13,4,44,84); (13,4,44,92); (13,4,52,92); (13,8,48,88); (13,8,48,96); (13,8,56,96); (13,12,52,92); (13,12,52,100); (13,12,60,100); (13,16,56,96); (13,16,56,104); (13,16,64,104); (13,20,60,100); (13,20,60,108)
	(13,20,68,108); (13,24,64,104); (13,24,64,112); (13,24,72,112); (13,28,68,108); (13,28,68,116); (13,28,76,116); (13,32,72,112); (13,32,72,120); (13,32,80,120); (13,36,76,116); (13,36,76,124); (13,36,84,124); (13,40,80,120); (13,44,84,124); (19,3,27,51)
	(19,3,27,107); (19,3,83,107); (19,7,31,55); (19,7,31,111); (19,7,87,111); (19,11,35,59); (19,11,35,115); (19,11,91,115); (19,15,39,63); (19,15,39,119); (19,15,95,119); (19,19,43,67); (19,19,43,123); (19,19,99,123); (19,23,47,71); (19,23,47,127)
	(19,23,103,127); (19,27,51,75); (19,31,55,79); (19,35,59,83); (19,39,63,87); (19,43,67,91); (19,47,71,95); (19,51,75,99); (19,55,79,103); (19,59,83,107); (19,63,87,111); (19,67,91,115); (19,71,95,119); (19,75,99,123); (19,79,103,127); (29,0,8,16)
	(29,0,8,120); (29,0,112,120); (29,4,12,20); (29,4,12,124); (29,4,116,124); (29,8,16,24); (29,12,20,28); (29,16,24,32); (29,20,28,36); (29,24,32,40); (29,28,36,44); (29,32,40,48); (29,36,44,52); (29,40,48,56); (29,44,52,60); (29,48,56,64)
	(29,52,60,68); (29,56,64,72); (29,60,68,76); (29,64,72,80); (29,68,76,84); (29,72,80,88); (29,76,84,92); (29,80,88,96); (29,84,92,100); (29,88,96,104); (29,92,100,108); (29,96,104,112); (29,100,108,116); (29,104,112,120); (29,108,116,124); (35,3,11,19)
	(35,3,11,123); (35,3,115,123); (35,7,15,23); (35,7,15,127); (35,7,119,127); (35,11,19,27); (35,15,23,31); (35,19,27,35); (35,23,31,39); (35,27,35,43); (35,31,39,47); (35,35,43,51); (35,39,47,55); (35,43,51,59); (35,47,55,63); (35,51,59,67)
	(35,55,63,71); (35,59,67,75); (35,63,71,79); (35,67,75,83); (35,71,79,87); (35,75,83,91); (35,79,87,95); (35,83,91,99); (35,87,95,103); (35,91,99,107); (35,95,103,111); (35,99,107,115); (35,103,111,119); (35,107,115,123); (35,111,119,127); (45,0,24,48)
	(45,0,24,104); (45,0,80,104); (45,4,28,52); (45,4,28,108); (45,4,84,108); (45,8,32,56); (45,8,32,112); (45,8,88,112); (45,12,36,60); (45,12,36,116); (45,12,92,116); (45,16,40,64); (45,16,40,120); (45,16,96,120); (45,20,44,68); (45,20,44,124)
	(45,20,100,124); (45,24,48,72); (45,28,52,76); (45,32,56,80); (45,36,60,84); (45,40,64,88); (45,44,68,92); (45,48,72,96); (45,52,76,100); (45,56,80,104); (45,60,84,108); (45,64,88,112); (45,68,92,116); (45,72,96,120); (45,76,100,124); (51,3,43,83)
	(51,3,43,91); (51,3,51,91); (51,7,47,87); (51,7,47,95); (51,7,55,95); (51,11,51,91); (51,11,51,99); (51,11,59,99); (51,15,55,95); (51,15,55,103); (51,15,63,103); (51,19,59,99); (51,19,59,107); (51,19,67,107); (51,23,63,103); (51,23,63,111)
	(51,23,71,111); (51,27,67,107); (51,27,67,115); (51,27,75,115); (51,31,71,111); (51,31,71,119); (51,31,79,119); (51,35,75,115); (51,35,75,123); (51,35,83,123); (51,39,79,119); (51,39,79,127); (51,39,87,127); (51,43,83,123); (51,47,87,127); (61,0,16,72)
	(61,0,56,72); (61,0,56,112); (61,4,20,76); (61,4,60,76); (61,4,60,116); (61,8,24,80); (61,8,64,80); (61,8,64,120); (61,12,28,84); (61,12,68,84); (61,12,68,124); (61,16,32,88); (61,16,72,88); (61,20,36,92); (61,20,76,92); (61,24,40,96)
	(61,24,80,96); (61,28,44,100); (61,28,84,100); (61,32,48,104); (61,32,88,104); (61,36,52,108); (61,36,92,108); (61,40,56,112); (61,40,96,112); (61,44,60,116); (61,44,100,116); (61,48,64,120); (61,48,104,120); (61,52,68,124); (61,52,108,124); (67,3,19,75)
	(67,3,59,75); (67,3,59,115); (67,7,23,79); (67,7,63,79); (67,7,63,119); (67,11,27,83); (67,11,67,83); (67,11,67,123); (67,15,31,87); (67,15,71,87); (67,15,71,127); (67,19,35,91); (67,19,75,91); (67,23,39,95); (67,23,79,95); (67,27,43,99)
	(67,27,83,99); (67,31,47,103); (67,31,87,103); (67,35,51,107); (67,35,91,107); (67,39,55,111); (67,39,95,111); (67,43,59,115); (67,43,99,115); (67,47,63,119); (67,47,103,119); (67,51,67,123); (67,51,107,123); (67,55,71,127); (67,55,111,127); (77,0,40,80)

Continued on next page

Category	Parameters (p, t_0, t_1, t_2)
	(77,0,40,88); (77,0,48,88); (77,4,44,84); (77,4,44,92); (77,4,52,92); (77,8,48,88); (77,8,48,96); (77,8,56,96); (77,12,52,92); (77,12,52,100); (77,12,60,100); (77,16,56,96); (77,16,56,104); (77,16,64,104); (77,20,60,100); (77,20,60,108)
	(77,20,68,108); (77,24,64,104); (77,24,64,112); (77,24,72,112); (77,28,68,108); (77,28,68,116); (77,28,76,116); (77,32,72,112); (77,32,72,120); (77,32,80,120); (77,36,76,116); (77,36,76,124); (77,36,84,124); (77,40,80,120); (77,44,84,124); (83,3,27,51)
	(83,3,27,107); (83,3,83,107); (83,7,31,55); (83,7,31,111); (83,7,87,111); (83,11,35,59); (83,11,35,115); (83,11,91,115); (83,15,39,63); (83,15,39,119); (83,15,95,119); (83,19,43,67); (83,19,43,123); (83,19,99,123); (83,23,47,71); (83,23,47,127)
	(83,23,103,127); (83,27,51,75); (83,31,55,79); (83,35,59,83); (83,39,63,87); (83,43,67,91); (83,47,71,95); (83,51,75,99); (83,55,79,103); (83,59,83,107); (83,63,87,111); (83,67,91,115); (83,71,95,119); (83,75,99,123); (83,79,103,127); (93,0,8,16)
	(93,0,8,120); (93,0,112,120); (93,4,12,20); (93,4,12,124); (93,4,116,124); (93,8,16,24); (93,12,20,28); (93,16,24,32); (93,20,28,36); (93,24,32,40); (93,28,36,44); (93,32,40,48); (93,36,44,52); (93,40,48,56); (93,44,52,60); (93,48,56,64)
	(93,52,60,68); (93,56,64,72); (93,60,68,76); (93,64,72,80); (93,68,76,84); (93,72,80,88); (93,76,84,92); (93,80,88,96); (93,84,92,100); (93,88,96,104); (93,92,100,108); (93,96,104,112); (93,100,108,116); (93,104,112,120); (93,108,116,124); (99,3,11,19)
	(99,3,11,123); (99,3,115,123); (99,7,15,23); (99,7,15,127); (99,7,119,127); (99,11,19,27); (99,15,23,31); (99,19,27,35); (99,23,31,39); (99,27,35,43); (99,31,39,47); (99,35,43,51); (99,39,47,55); (99,43,51,59); (99,47,55,63); (99,51,59,67)
	(99,55,63,71); (99,59,67,75); (99,63,71,79); (99,67,75,83); (99,71,79,87); (99,75,83,91); (99,79,87,95); (99,83,91,99); (99,87,95,103); (99,91,99,107); (99,95,103,111); (99,99,107,115); (99,103,111,119); (99,107,115,123); (99,111,119,127); (109,0,24,48)
	(109,0,24,104); (109,0,80,104); (109,4,28,52); (109,4,28,108); (109,4,84,108); (109,8,32,56); (109,8,32,112); (109,8,88,112); (109,12,36,60); (109,12,36,116); (109,12,92,116); (109,16,40,64); (109,16,40,120); (109,16,96,120); (109,20,44,68); (109,20,44,124)
	(109,20,100,124); (109,24,48,72); (109,28,52,76); (109,32,56,80); (109,36,60,84); (109,40,64,88); (109,44,68,92); (109,48,72,96); (109,52,76,100); (109,56,80,104); (109,60,84,108); (109,64,88,112); (109,68,92,116); (109,72,96,120); (109,76,100,124); (115,3,43,83)
	(115,3,43,91); (115,3,51,91); (115,7,47,87); (115,7,47,95); (115,7,55,95); (115,11,51,91); (115,11,51,99); (115,11,59,99); (115,15,55,95); (115,15,55,103); (115,15,63,103); (115,19,59,99); (115,19,59,107); (115,19,67,107); (115,23,63,103); (115,23,63,111)
	(115,23,71,111); (115,27,67,107); (115,27,67,115); (115,27,75,115); (115,31,71,111); (115,31,71,119); (115,31,79,119); (115,35,75,115); (115,35,75,123); (115,35,83,123); (115,39,79,119); (115,39,79,127); (115,39,87,127); (115,43,83,123); (115,47,87,127); (125,0,16,72)
	(125,0,56,72); (125,0,56,112); (125,4,20,76); (125,4,60,76); (125,4,60,116); (125,8,24,80); (125,8,64,80); (125,8,64,120); (125,12,28,84); (125,12,68,84); (125,12,68,124); (125,16,32,88); (125,16,72,88); (125,20,36,92); (125,20,76,92); (125,24,40,96)
	(125,24,80,96); (125,28,44,100); (125,28,84,100); (125,32,48,104); (125,32,88,104); (125,36,52,108); (125,36,92,108); (125,40,56,112); (125,40,96,112); (125,44,60,116); (125,44,100,116); (125,48,64,120); (125,48,104,120); (125,52,68,124); (125,52,108,124)

References

[1] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) Advances in Cryptology - ASIACRYPT 2012, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer, ??? (2012)

[2] Avanzi, R.: The QARMA block cipher family - almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. IACR Trans. Symmetric Cryptol. **2017**(1), 4–44 (2017)

[3] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y.,

Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. 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 II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer, ??? (2016). https://doi.org/10.1007/978-3-662-53008-5_5 . https://doi.org/10.1007/978-3-662-53008-5_5

[4] 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

[5] 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

[6] 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 . https://doi.org/10.1007/3-540-48658-5_3

[7] Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: A new tool for differential-linear cryptanalysis. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11476, pp. 313–342. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-17653-2_11 . https://doi.org/10.1007/978-3-030-17653-2_11

[8] Blondeau, C., Leander, G., Nyberg, K.: Differential-linear cryptanalysis revisited. J. Cryptol. **30**(3), 859–888 (2017)

[9] Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher square. In: Biham, E. (ed.) Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1267, pp. 149–165. Springer, ??? (1997). https://doi.org/10.1007/BFB0052343 . https://doi.org/10.1007/BFb0052343

[10] Knudsen, L.R., Wagner, D.A.: Integral cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2365, pp. 112–127. Springer, ??? (2002). https://doi.org/10.1007/3-540-45661-9_9 . https://doi.org/10.1007/3-540-45661-9_9

[11] Boura, C., Canteaut, A., Cannière, C.D.: Higher-order differential properties of keccak and *Luffa*. In: Joux, A. (ed.) Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6733, pp. 252–269. Springer, ??? (2011). https://doi.org/10.1007/978-3-642-21702-9_15 . https://doi.org/10.1007/978-3-642-21702-9_15

[12] Boura, C., Canteaut, A.: On the influence of the algebraic degree of f^{-1} on the algebraic degree of G ∘ F. IEEE Trans. Inf. Theory **59**(1), 691–702 (2013) https://doi.org/10.1109/TIT.2012.2214203

[13] Carlet, C.: Graph indicators of vectorial functions and bounds on the algebraic degree of composite functions. IEEE Trans. Inf. Theory **66**(12), 7702–7716 (2020) https://doi.org/10.1109/TIT.2020.3017494

[14] Todo, Y.: Structural evaluation by generalized integral property. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 287–314. Springer, ??? (2015). https://doi.org/10.1007/978-3-662-46800-5_12 . https://doi.org/10.1007/978-3-662-46800-5_12

[15] Liu, M.: Degree evaluation of nfsr-based cryptosystems. 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. 227–249. Springer, ??? (2017). https://doi.org/10.1007/978-3-319-63697-9_8 . https://doi.org/10.1007/978-3-319-63697-9_8

[16] Chen, S., Xiang, Z., Zeng, X., Zhang, S.: On the relationships between different methods for degree evaluation. IACR Trans. Symmetric Cryptol. **2021**(1), 411–442 (2021) https://doi.org/10.46586/TOSC.V2021.I1.411-442

[17] Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer, ??? (2014). https://doi.org/10.1007/978-3-662-45611-8_9 . https://doi.org/10.1007/978-3-662-45611-8_9

[18] Feng, X., Tian, Y., Wang, Y., Xu, S., Zhang, A.: Full linear integer inequality characterization of sets over Z_2^n. ChinaXiv **202210.00055V2** (2022)

[19] Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T.

(ed.) Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer, ??? (1993). https://doi.org/10.1007/3-540-48285-7_33 . https://doi.org/10.1007/3-540-48285-7_33

[20] Sasaki, Y., Todo, Y.: New algorithm for modeling s-box in MILP based differential and division trail search. In: Farshim, P., Simion, E. (eds.) Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10543, pp. 150–165. Springer, ??? (2017). https://doi.org/10.1007/978-3-319-69284-5_11 . https://doi.org/10.1007/978-3-319-69284-5_11